Linköping studies in science and technology. Licentiate Thesis.
No. 1642

Model Predictive Control in Flight Control Design

Stability and Reference Tracking

Daniel Simon

Division of Automatic Control
Department of Electrical Engineering
Linköping University, SE-581 83 Linköping, Sweden
http://www.control.isy.liu.se
dansi@isy.liu.se

Linköping 2014

This is a Swedish Licentiate's Thesis.

Swedish postgraduate education leads to a Doctor's degree and/or a Licentiate's degree.
A Doctor's Degree comprises 240 ECTS credits (4 years of full-time studies).
A Licentiate's degree comprises 120 ECTS credits,
of which at least 60 ECTS credits constitute a Licentiate's thesis.

Linköping studies in science and technology. Licentiate Thesis.
No. 1642

Model Predictive Control in Flight Control Design – Stability and Reference Tracking

Daniel Simon

dansi@isy.liu.se
www.control.isy.liu.se
Department of Electrical Engineering
Linköping University
SE-581 83 Linköping
Sweden

ISBN 978-91-7519-422-6 ISSN 0280-7971 LIU-TEK-LIC-2013:76

Printed by LiU-Tryck, Linköping, Sweden 2014

To whom it may concern...

Abstract

Aircraft are dynamic systems that naturally contain a variety of constraints and nonlinearities such as, e.g., maximum permissible load factor, angle of attack and control surface deflections. Taking these limitations into account in the design of control systems are becoming increasingly important as the performance and complexity of the controlled systems is constantly increasing. It is especially important in the design of control systems for fighter aircraft. These require maximum control performance in order to have the upper hand in a dogfight or when they have to outmaneuver an enemy missile. Therefore pilots often maneuver the aircraft very close to the limit of what it is capable of, and an automatic system (called flight envelope protection system) against violating the restrictions is a necessity.

In other application areas, nonlinear optimal control methods have been successfully used to solve this but in the aeronautical industry, these methods have not yet been established. One of the more popular methods that are well suited to handle constraints is Model Predictive Control (MPC) and it is used extensively in areas such as the process industry and the refinery industry. Model predictive control means in practice that the control system iteratively solves an advanced optimization problem based on a prediction of the aircraft's future movements in order to calculate the optimal control signal. The aircraft's operating limitations will then be constraints in the optimization problem.

In this thesis, we explore model predictive control and derive two fast, low complexity algorithms, one for guaranteed stability and feasibility of nonlinear systems and one for reference tracking for linear systems. In reference tracking model predictive control for linear systems we build on the dual mode formulation of MPC and our goal is to make minimal changes to this framework, in order to develop a reference tracking algorithm with guaranteed stability and low complexity suitable for implementation in real time safety critical systems.

To reduce the computational burden of nonlinear model predictive control several methods to approximate the nonlinear constraints have been proposed in the literature, many working in an ad hoc fashion, resulting in conservatism, or worse, inability to guarantee recursive feasibility. Also several methods work in an iterative manner which can be quit time consuming making them inappropriate for fast real time applications. In this thesis we propose a method to handle the nonlinear constraints, using a set of dynamically generated local inner polytopic approximations. The main benefits of the proposed method is that while computationally cheap it still can guarantee recursive feasibility and convergence.

Populärvetenskaplig sammanfattning

Flygplan är dynamiska system som naturligt innehåller en mängd begränsningar och olinjäriteter så som t.ex. max tillåten lastfaktor, anfallsvinkel och roderlägen. Att ta hänsyn till dessa begränsningar i designen av styrsystem blir allt viktigare då prestanda och komplexiteten hos de styrda systemen hela tiden ökar. Speciellt viktigt är det i designen av styrsystem för jaktflygplan. Dessa kräver maximal manöverprestanda för att kunna ha övertaget i en luftstrid eller då de måste utmanövrera en fientlig missil. Detta gör att piloterna manövrerar flygplanen väldigt nära gränsen för vad farkosten klarar av och ett automatiskt skydd (s.k. manöverskydd) mot att bryta mot begränsningarna är en nödvändighet.

Inom andra tillämpningsområden har olinjära optimalstyrningsmetoder framgångsrikt använts för att lösa detta men inom flygindustrin har dessa metoder ännu inte etablerats. En av de mer populära metoder som är väl lämpade för att hantera begränsningar är modellbaserad prediktionsreglering (MPC) som tillämpats flitigt inom områden så som processindustrin och raffineringsindustrin. Modellbaserad prediktionsreglering innebär i praktiken att styrsystemet hela tiden iterativt löser ett avancerat optimeringsproblem baserat på en prediktion av flygplanets framtida rörelser för att beräkna den bästa möjliga styrsignalen. Flygplanets manöverbegränsningar blir då bivillkor till optimeringsproblemet.

I detta arbete tittar vi närmare på prediktionsreglering och några egenskaper som är centrala för flygindustrin. Vi kommer behandla aspekter som stabilitet hos regleralgortimerna och följning av pilotens kommandon. Två centrala egenskaper för flygindustrin som man hela tiden måste ta i beaktning är komplexiteten hos regleralgoritmerna och möjligheten att certifiera dessa. Låg komplexitet hos algoritmerna eftersträvas för att den flygsäkerhetsgodkända hårdvara som måste användas sällan innehåller state-of-art processorer med stor beräkningskraft. Den andra viktiga aspekten är möjligheten att certifiera styrlagarna. Det ställs väldigt strikta krav på den mjukvara som finns i ett styrsystem i ett flygplan och i dagsläget är det tveksamt att den relativt konservativa branschen kan acceptera att online optimeringsalgoritmer beräknar flygplanets styrkommandon.

Acknowledgments

If someone would have told me four years ago that I would be half way to a PhD degree by now, I would never have believed them. I was convinced that I lacked both the knowledge and intellect to undertake such a task, but in some mysterious way, here it is, the half-way-mark, my Licentiate thesis. This is an accomplishment which I would never have been able to finish if it wasn't for all the help I have gotten along the way from people with both more knowledge and higher intellect than me; some of which deserves a special mentioning.

First and foremost I would like to thank my supervisors, Assoc. Prof. Johan Löfberg and Prof. Torkel Glad, who have guided me with the greatest patience through the tangled forest that is academic research. I would also like to thank my industry supervisor, Dr. Ola Härkegård for encouraging me to start on this journey and for always asking me those fundamental questions which I have overlooked.

A person who perhaps hasn't been a formal supervisor but still the unlucky subject for all my, sometimes almost slow-witted, questions is Dr. Daniel Petersson. He deserves my deepest gratitude for always having time for me, whether it concerned algebra, optimization or computer issues. I would also like to thank him and Dr. Christian Lyzell for introducing me to Ti*kZ*, which greatly increased the quality of the figures in this thesis.

Speaking of the thesis; Lic. Patrik Axelsson, Lic. Roger Larsson, Lic. Ylva Jung, Lic. Sina Khoshfetrat Pakazad and M.Sc. Isak Nielsen all deserves a special acknowledgment for proofreading various chapter of this thesis. Here I would also like to acknowledge the two former members of the group, Dr. Gustaf Hendeby and Dr. Henrik Tidefeldt for the excellent LaTeX-template which prevented me from spending an eternity on various layout issues.

To my brother in arms, Lic. Roger Larsson, with whom I have the pleasure of sharing office and travel to distant countries with, to you I only say: *Don't go left, left, left and four stairs up...*

I would also like to thank the rest my colleagues at the Automatic Control Group and Prof. Svante Gunnarsson as Head of the group for creating a fun and inspiring working environment.

The financial support from VINNOVA and the NFFP5-program which made this journey possible is gratefully acknowledged.

Almost last but far from least. I would like to thank my family and friends for always supporting me, with love and patience, through all the discouragements of my life. Especially *Anna* whose smile brightens even the darkest winter days.

Finally, to You, who I have forgot to mention, Thank You!

Linköping, January 2014
Daniel Simon

Contents

Notation **xiii**

1 Introduction **1**
1.1 Research motivation . . . 1
1.2 Publications and contributions . . . 6
1.3 Thesis outline . . . 7

2 Mathematical Preliminaries **9**
2.1 Optimization . . . 9
2.1.1 Convex optimization . . . 10
2.1.2 Duality . . . 14
2.1.3 Nonconvex optimization . . . 16
2.2 Convex polytopic geometry . . . 18

3 Introduction to Model Predictive Control **23**
3.1 Introduction . . . 23
3.2 Linear MPC . . . 24
3.2.1 Stability . . . 26
3.2.2 Reference tracking . . . 30
3.2.3 Integral control . . . 33
3.2.4 Slack variables . . . 37
3.2.5 The explicit solution . . . 39
3.3 Nonlinear MPC . . . 43

4 A Low Complexity Reference Tracking MPC Algorithm **47**
4.1 Introduction . . . 48
4.2 The proposed controller . . . 50
4.2.1 Vertex enumeration reformulation . . . 52
4.2.2 Dual formulation of terminal set constraints . . . 53
4.2.3 The QP formulation . . . 55
4.2.4 Stability and feasibility of the proposed algorithm . . . 56
4.3 Examples from the aeronautical industry . . . 61
4.3.1 Maneuver limitations on a fighter aircraft . . . 61

4.3.2 Helicopter flight envelope protection 66

5 Method for Guaranteed Stability and Recursive Feasibility in Nonlinear MPC 71

5.1 Introduction 72

5.1.1 Feedback linearization 73

5.2 The proposed algorithm 74

5.2.1 Nonlinear constraint approximations 75

5.2.2 MPC receding horizon setup 77

5.3 Examples 79

5.3.1 Illustrative scalar example 79

5.3.2 Nonlinear aircraft example 81

6 Conclusions and Future Work 87

Bibliography 89

Notation

Mathematics

Notation	Meaning
x^T, A^T	Vector or matrix transpose
A^{-1}	Matrix inverse
$A \succeq 0$	Positive semidefinite matrix
$x \geq 0$	Elementwise inequality
I	Identity matrix
$\mathbb{R}^n$	Space of real vectors of dimension n
$\lVert x \rVert$	General (arbitrary) norm of a vector
$\lVert x \rVert_2$	Euclidean norm of a vector
$\lVert x \rVert_\infty$	Infinity norm of a vector
$\lVert x \rVert_D$	Dual norm
$\lVert x \rVert_Q^2$	Weighted quadratic function, $x^T Q x$
$\mathcal{P}_1 \oplus \mathcal{P}_2$	Minkovsky sum of two sets
$\mathcal{P}_1 \ominus \mathcal{P}_2$	Pontryagin difference of two sets
$\mathcal{P}_1 \times \mathcal{P}_2$	Carthesian product of two sets
$conv(\cdot)$	Convex hull
$\text{int}(\mathcal{P})$	Interior of the set $\mathcal{P}$
$\text{int}_\epsilon(\mathcal{P})$	The epsilon-interior of the set $\mathcal{P}$
$\text{dom}(f)$	Domain of a function $f(x)$
$f_0(x)$	Optimization problem objective function
f_0^*	Optimal value of objective function
x^*	Optimal value on optimization variable
$f_i(x)$	Inequality constraint functions
$g_i(x)$	Equality constraint functions
$\inf f(x)$	Infimum of $f(x)$
$\nabla f(x)$	Gradient of a function
$\nabla^2 f(x)$	Hessian of a function (second derivative)

MPC

Notation	Meaning
$\mathcal{X}$	State constraint set
$\mathcal{U}$	Input constraint set
$\mathcal{T}$	Terminal state constraint set
Π	Nonconvex input constraint set
$\mathcal{G}$	Global polytopic inner approximation of Π
$\mathcal{I}_i^k$	Local polytopic inner approximation of Π, i time steps into the future at time k
$\mathcal{C}_k$	Outer polytopic approximation of Π
$\mathcal{J}_k$	Objective function value at time k
$\ell(\cdot)$	Stage cost
$\Psi(\cdot)$	Terminal state cost
$\phi(\cdot)$	Pseudo reference variable penalty function
$\kappa(x)$	Optimal control law
r	Reference input
$\bar{r}_k$	Pseudo reference variable
$\bar{x}_k$	Pseudo steady state
$\bar{u}_k$	Pseudo steady state control
ε	Slack variable
$\{x_i\}_{i=0}^N$	A sequence of variables x_i from $i = 0, \ldots, N$. I.e., $\{x_i\}_{i=0}^N = \{x_0, x_1, \ldots, x_N\}$
λ_k	Terminal state constraint set scaling variable
N	Prediction horizon
N_l	Prediction horizon for local polytope approximations, $\mathcal{I}_i^k$
$N_{\mathcal{X}}$	Number of state space partitions for explicit MPC

Aeronautical variables

Notation	Meaning
α	Angle of attack
β	Sideslip angle
q	Angular pitch rate
p	Roll rate
r	Yaw rate
θ	Pitch angle, i.e., angle between x-axis and horizontal plane
δ_e	Aircraft control surface angular deflection
δ_s	Helicopter swash plate pitch angle
v	Velocity vector x-component
Q	Dynamic pressure, $Q = \frac{1}{2}\rho v^2$
S	Wing surface area
$\bar{c}$	Mean cord
a	Helicopter rotor disc pitch angle
c	Helicopter stabilizer disc pitch angle

Abbreviations

Abbreviation	Meaning
KKT	Karush-Kuhn-Tucker
LQ	Linear Quadratic
MPC	Model Predictive Control
NMPC	Nonlinear Model Predictive Control
QP	Quadratic Program
SDP	SemiDefinite Programming
SQP	Sequential Quadratic Programming
s.t.	Subject to

1 Introduction

In this introductory chapter we give a background to the research topic that has been studied in this thesis. Section 1.1 explains the main problem that has motivated the research and in Section 1.2 we list the publications and explain the main contributions. Finally in Section 1.3 we give the outline of the thesis.

1.1 Research motivation

The motivation behind this thesis comes from the aeronautical industry. Aircraft and helicopters are very complex dynamic systems that put high requirements on performance and robustness of the controllers that are used. The dynamics is in general divided into two parts, the *pitch dynamics* which is the nose up/down movement and the *lateral dynamics* which is the wingtip up/down movement (roll motion) and nose left/right movement (yawing motion). The dynamics are normally considered to be rigid body motions and can be derived from Newton's laws of motion.

$$F = m\dot{v} + m(\omega \times V) \tag{1.1}$$

$$M = \mathbb{I}\dot{\omega} + \omega \times \mathbb{I}\omega \tag{1.2}$$

where F and M are the forces and moments acting on the aircraft, m is the mass, $\mathbb{I}$ is the inertia matrix, V is the velocity and ω is the angular velocity, all expressed in the aircraft body-fixed coordinate frame, see Figure 1.1. The forces and moments comes both from the engine's thrust, the gravity and from the aerodynamical properties of the aircraft, although for inner loop control law design the engine's contributions are normally neglected. The aerodynamic forces and moments are highly complex and nonlinear functions which depend on a variety of parameters such as, e.g., the aircraft's geometric properties, dynamic pressure, the aircraft's attitude against the airflow, control surface deflections etc.

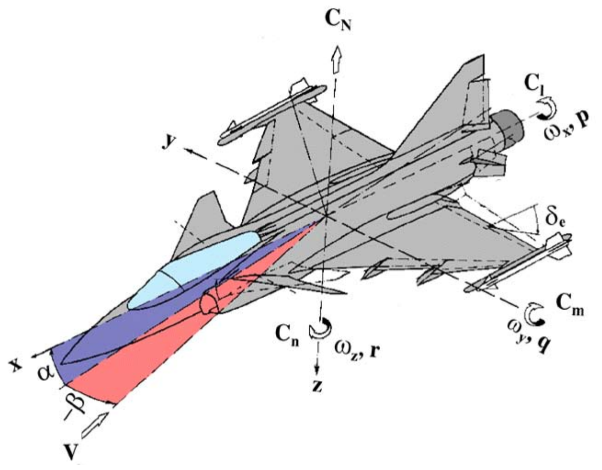

Figure 1.1: *Definitions of aerodynamic angles and moments of an aircraft in the body fixed coordinate frame.*

The standard way of modeling the aerodynamic forces and moments is to use dimensionless functions called *aerodynamic coefficient* and then scale these functions with the dynamic pressure, Q, wing sufrace, S and either wingspan, b, or the mean cord, $\bar{c}$ (depending on which moment and force that are modeled). For example the pitching moment and normal force are modeled as

$$F_z = -QSC_N \tag{1.3a}$$

$$M_y = QS\bar{c}C_m \tag{1.3b}$$

where C_N and C_m are the aerodynamic coefficients. The aerodynamic coefficients are normally modeled through wind tunnel testing or CFD (Computational Fluid Dynamics) computations. The aerodynamic coefficients depends on how the air flows around the aircraft and hence on the orientation of the aircraft in the airflow. This orientation is measured using the angles α and β, see Figure 1.1. If the angle α, known as the *angle of attack*, becomes too large the upper side of the wing is put out of the wind and the aircraft *stalls*, i.e., it looses its ability to a produce lifting force. On the other hand, when maneuvering an aircraft the pilot commands a change in the angle of attack and combined with a roll angle this will cause the aircraft to turn. The larger angle of attack that can be attained, the faster the aircraft turns so a trade off must be made between maneuverability and safety.

For flight control design one usually rearrange the equations (1.1) and (1.2) and then use *small disturbance theory* to derive linear differential equations; details of this can be found in Stevens and Lewis [2003]. The pitch dynamics, which is the dynamics we will focus on in this thesis, can then be modeled as a two state linear system where the angle, α, is one state and the angular pitch rate, q, is the other. The input to the system is the elevator control surface deflection, δ_e. We will refer to this motion as the *short period dynamics*

$$\begin{bmatrix}\dot{\alpha}\\ \dot{q}\end{bmatrix} = \begin{bmatrix}\frac{Z_a}{v} & 1\\ \left(M_\alpha + M_{\dot{\alpha}}\frac{Z_\alpha}{v}\right) & \left(M_q + M_{\dot{\alpha}}\right)\end{bmatrix}\begin{bmatrix}\alpha\\ q\end{bmatrix} + \begin{bmatrix}Z_{\delta_e}\\ M_{\delta_e} + \frac{M_{\dot{\alpha}}}{v}Z_{\delta_e}\end{bmatrix}\delta_e \tag{1.4}$$

Here $M_{(\cdot)}$ is the derivative of the moment equation (1.3b) with respect to the different variables and $Z_{(\cdot)}$ is the derivative of the force equation (1.3a) with respect to the different variables. The derivatives have the form

$$\begin{aligned}
Z_\alpha &= \frac{-C_{N_\alpha}QS}{m} & Z_{\delta_e} &= \frac{-C_{N_{\delta_e}}QS}{m}\\
M_\alpha &= \frac{C_{m_a}QS\bar{c}}{I_y} & M_q &= \frac{C_{m_q}QS\bar{c}^2}{2vI_y}\\
M_{\delta_e} &= \frac{C_{m_{\delta_e}}QS\bar{c}}{I_y} & M_{\dot{\alpha}} &= \frac{C_{m_{\dot{\alpha}}}QS\bar{c}^2}{2vI_y}
\end{aligned}$$

From the above we can see that the dynamics is changing with speed and also with altitude, since the dynamic pressure depends both on the air density, ρ, which depends on the altitude and the airspeed. The corresponding differential equations for the lateral dynamics is a three state dynamical system with the states, roll rate, p, sideslip, β and yaw rate, r.

One major objective of a flight control system, which is studied in this thesis, is to limit the aircraft response to the pilot inputs such that it does not exceed any structural or aerodynamical limitations of the vehicle. The flight control system shall be able to limit the response of the aircraft such that the states remain within a region where the aircraft is flyable, the so called *flight envelope*. This ability is called *flight envelope protection* or *maneuver load limitation*.

In civil aviation several severe incidents have occurred where a maneuver load limiting system would have resolved the problem. For example, after an engine flameout during cruise on China Airlines flight 006 the aircraft went into a steep dive and when the pilot performed agressive maneuvering to recover from the incident it caused massive damages to the horizontal stabilizers [NTSB, 1986], see Figure 1.2. Another example is that of American Airlines flight 587, that shortly after take off entered a wake vortex from a Boeing 747. To control the aircraft in the strong turbulence the pilot commanded excessive rudder commands which caused the tail to brake off and the aircraft to crash [NTSB, 2004].

In helicopter control one wants to limit the maximum descent speed in order to avoid the *vortex ring state*. This is a condition in which the helicopter descends into its own downwash generated by the rotor. This causes the rotor to generate a vortex around the helicopter without any lifting force.

Modern military aircraft offer a challenging control task since they operate over a wide range of conditions and are required to perform at their best in all conditions. For an agile fighter superior maneuverability is vital for its success in any

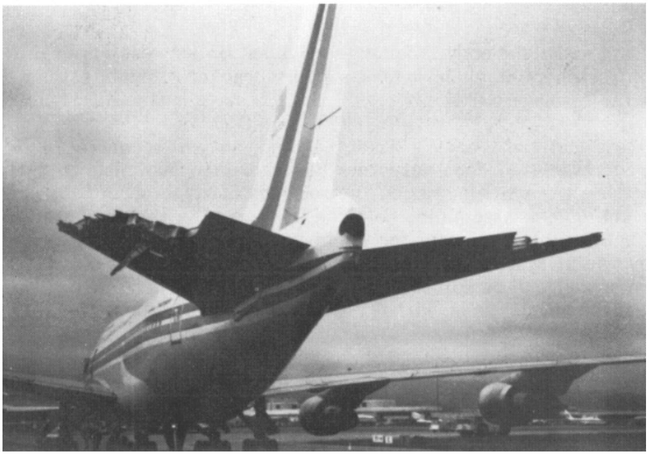

Figure 1.2: *Damages to the horizontal stabilizers of China Airlines flight 006, a Boeing 747, due to structural overload. Image from the report NTSB [1986] acquired via Wikimedia Commons.*

mission, e.g., avoiding enemy air defense missiles or outmaneuvering hostile aircraft. Therefore one wants to be able to control the aircraft to the limit of what is possible, but not in such way that the aircraft loses lift force and stalls. The modern concept of *carefree maneuvering* means that the pilot shall be able to entirely focus on the mission tasks while the flight control system automatically limits the attainable performance so that the aircraft remains controllable no matter what the pilot's inputs.

A way to systematically incorporate flight envelope protection and enable carefree maneuvering into the control system design is to add limitations on the states and control (or the pilot commands) but this results in a nonlinear control problem that in general is far more complicated to solve than the linear techniques that are standard in the industry today. One of the modern control system design techniques that by far has gained the most popularity in the aircraft industry is *Linear Quadratic Control* (LQ), e.g., the Swedish fighter aircraft *JAS 39 Gripen* uses a gain scheduled LQ controller for its primary stabilizing controller. The LQ design technique is based on minimizing an objective which is quadratic in the states, $x(t)$, and the controls, $u(t)$.

$$\underset{u(t)}{\text{minimize}} \quad \int_0^\infty x(t)^T Q x(t) + u(t)^T R u(t)\, dt \tag{1.5}$$

with (as the name indicate) linear system dynamics, i.e., the state evolution is described by a linear differential equation.

$$\dot{x}(t) = Ax(t) + Bu(t) \tag{1.6}$$

Kalman [1960] showed that this problem can be solved explicitly and the optimal control law, $\kappa(x)$, is a linear state feedback.

$$u(t) = \kappa(x) = -Kx(t)$$

However, adding constraints on states and controls to the LQ problem formulation (1.5) and (1.6) results in a nonlinear optimal control problem

$$\underset{u(t)}{\text{minimize}} \int_0^\infty x(t)^T Q x(t) + u(t)^T R u(t) \, dt \tag{1.7a}$$

$$\text{s.t.}$$

$$\dot{x}(t) = Ax + Bu \tag{1.7b}$$

$$x(t) \in \mathcal{X} \tag{1.7c}$$

$$u(t) \in \mathcal{U} \tag{1.7d}$$

which, in contrast to the LQ problem, can be extremely difficult to solve explicitly. An open loop solution can be obtained using the *Pontryagin Maximum Principle* (PMP) but to obtain an explicit closed loop feedback solution, i.e., $u(t) = \kappa(x)$, one has to solve the partial differential equation known as the *Hamilton-Jacobi-Bellman Equation*. Unfortunately this is for almost all practical cases impossible to solve analytically [Maciejowski, 2002].

To overcome this, other application areas, such as the process industry, most frequently use real-time optimization-based methods to approximate the nonlinear optimal control problem. One of the more popular optimization-based methods is *Model Predictive Control* (MPC), but in the aeronautical industry this methodology has not yet been established. Instead anti-windup-similar techniques like *Override control* [Turner and Postlethwaite, 2002, 2004, Glattfelder and Schaufelberger, 2003] and ad-hoc engineering solutions have been applied. The main drawback with these ad-hoc methods is that they usually lack any theoretical foundation and instead they rely on engineering experience and insight, and also that they require a tremendous amount of time to tune.

In the last decade there have been an increasing interest in investigating MPC for aeronautical applications [Keviczky and Balas, 2006, Gros et al., 2012, Dunbar et al., 2002]. The biggest interest have been to design reconfigurable flight control laws that can adapt to actuator failures and battle damages [Almeida and Leissling, 2009, Maciejowski and Jones, 2003, Kale and Chipperfield, 2005]. For example in Maciejowski and Jones [2003] the authors claim that the fatal crash of the El Al Flight 1862 [NLR, 1992] could have been avoided if a fault tolerant MPC controller had been implemented.

The aim of this thesis is to investigate Model Predictive Control from the perspective of flight envelope protection and to develop MPC algorithms that can be deemed appropriate to control the aircraft.

1.2 Publications and contributions

The main theoretical contributions of this thesis are on reference tracking in linear MPC and on guaranteed stability of MPC for nonlinear systems.

In the conference paper

> D. Simon, J. Löfberg, and T. Glad. Reference tracking MPC using terminal set scaling. In *51st IEEE Conference on Decision and Control (CDC)*, pages 4543–4548, Dec. 2012.

we extended the standard *dual mode* MPC formulation to a simple reference tracking algorithm and studied the stability properties of the proposed algorithm. Due to the iterative nature of MPC one must take special measures to ensure that the optimization problem remains feasible and that the controller stabilizes the system. However, these measures can in the severe cases limit the reference tracking ability or result in a complex algorithm. The main theoretical contributions of this paper was to improve the possibility of reference tracking in linear MPC by making simple adjustments to the existing stabilizing constraints of the dual mode formulation so that they are more suitable for tracking.

The proposed MPC algorithm that was derived in the above conference paper suffered some major drawbacks. It required the complete enumeration of all vertices in a, possibly complicated, polytopic set. Since the computational burden of this task, in the worst case, grows exponentially with the state dimension it was desired to reformulate the algorithm such that the vertex enumeration was avoided. In the journal paper

> D. Simon, J. Löfberg, and T. Glad. Reference Tracking MPC using Dynamic Terminal Set Transformation. *IEEE Transactions on Automatic Control*, 2014. Provisionally accepted for publication.

we derived a dual formulation of the constraints involving vertex enumerations. This reformulation greatly reduced the worst case complexity of the controller making it suitable for implementation. An example from the aircraft industry showed that the proposed controller has the potential to be far less complex than existing state of art algorithms without loosing any performance.

In the conference paper

> D. Simon, J. Löfberg, and T. Glad. Nonlinear Model Predictive Control using Feedback Linearization and Local Inner Convex Constraint Approximations. In *Proceedings of the 2013 European Control Conference*, number 3, pages 2056–2061, 2013.

we considered the task of controlling a constrained nonlinear system with a combination of feedback linearization and linear MPC. This approach in general lead to an optimization problem with nonlinear and state dependent constraints on the control signal. The main contribution in this paper is that we replace the nonlinear control signal constraints with a set of convex approximations. The proposed algorithm result in an easy solvable convex optimization problem for which we can guarantee recursive feasibility and convergence.. An example from the aircraft industry show that the performance loss compared to using a global nonlinear branch and bound algorithm can be very small.

1.3 Thesis outline

The outline of this thesis is as follows.

In Chapter 2 we outline the necessary mathematical background for the remaining parts of the thesis. We discuss nonlinear optimization and distinguish between *convex* problems and *nonconvex* problems and their properties. In this chapter we also give a brief introduction to *Convex Polytope Geometry* which will play a central roll in the upcoming chapters of the thesis.

Chapter 3 gives the reader an introduction to Model Predictive Control. For linear systems we present the main stability results, reference tracking concepts, practical aspects of robustifications such as integral control and soft constraints and derive the explicit MPC formulation. For nonlinear systems we only briefly discuss the complicating factors such as guaranteed stability, recursive feasibility and nonconvexity of the optimization problem.

The main results of the thesis are presented in Chapter 4 and Chapter 5. In Chapter 4 we derive a reference tracking MPC algorithm which has a low complexity and a simple structure. We calculate the explicit MPC formulation and compare the complexity to one other state of the art method. In this chapter we also introduce integral control to handle model errors.

In Chapter 5 we combine a feedback linearization controller with a linear MPC structure to stabilize the nonlinear dynamics of a fighter aircraft. We prove stability and recursive feasibility of the proposed algorithm and also compare the degree of sub-optimality of the proposed algorithm to a globally optimal branch and bound method.

The thesis finishes with some concluding remarks and thoughts on future work in Chapter 6.

2

Mathematical Preliminaries

In this chapter we will provide the necessary mathematical tools which we will use frequently in the remaining chapters of the thesis. Section 2.1 will give a very brief introduction to mathematical optimization, distinguish between *convex* and *nonconvex* optimization problems and discuss the concept of duality. For clarity of the presentation we have skipped many important details and concepts and we refer the reader to Boyd and Vandenberghe [2004] for a very comprehensive presentation of the material.

Section 2.2 outlines basic properties and fundamental calculations with convex polytopic sets.

2.1 Optimization

An optimization problem is the problem of finding a value for the variable, x, which minimizes (or maximizes) a certain objective, possibly, while satisfying a set of constraints. A general formulation of an optimization problem can be written as

$$\underset{x}{\text{minimize}}\ f_0(x) \tag{2.1a}$$

$$\text{s.t.}$$

$$f_i(x) \leq 0 \quad i = 1, \ldots, m \tag{2.1b}$$

$$g_i(x) = 0 \quad i = 1, \ldots, p \tag{2.1c}$$

where $f_0 : \mathbb{R}^n \to \mathbb{R}$ is the objective function (or cost function) which we want to minimize and the functions $f_i : \mathbb{R}^n \to \mathbb{R}$ and $g_i : \mathbb{R}^n \to \mathbb{R}$ are the inequality and equality constraint functions. We will adopt the convention of using a straight inequality sign, $\leq$, for scalars and for elementwise vector valued inequalities, while we will use the curly inequality sign, $\succeq$, to denote the positive semidefiniteness property of matrices.

The optimal objective value is denoted as f_0^* and the optimal (minimizing) variable value is denoted as x^*; i.e.,

$$f_0^* = \inf_x \ \{f_0(x) \mid f_i(x) \leq 0 \ i = 1, \ldots, m, \quad g_i(x) = 0 \ i = 1, \ldots, p\}$$
$$x^* = \{x \mid f_0(x) = f_0^*\}$$

A value $\hat{x}$ is said to be *feasible* if and only if $f_i(\hat{x}) \leq 0, \ \forall i = 1, \ldots, m$ and $g_i(\hat{x}) = 0, \ \forall i = 1, \ldots, p$ and it is *strictly feasible* if the inequalities hold strictly.

The problem (2.1) is often referred to as the *Primal problem* and a value $\hat{x}$, satisfying (2.1b) and (2.1c) as *Primal feasible*.

2.1.1 Convex optimization

The optimization problem (2.1) is said to be *convex* (or having a *convex representation*) if the objective function, f_0, is a convex function (if it is a minimization problem and concave if it is a maximization problem) of the variable x, the inequality constraint functions, f_i, are convex functions of x and the equality constraint functions are *affine*, i.e., $g_i(x) = a_i^T x + b_i$.

2.1 Definition. A function, $f(x)$, is said to be *convex* if and only if

$$f(\gamma x_1 + (1-\gamma)x_2) \leq \gamma f(x_1) + (1-\gamma) f(x_2)$$

for any two points x_1 and x_2 and any scalar $0 \leq \gamma \leq 1$.

In other words, a function is convex if the function curve between any two points lies below the line connecting those two points, see Figure 2.1. The function is *concave* if the opposite holds. From the definition we can derive the first and second order conditions for convexity.

2.2 Definition. A differentiable function is convex if and only if it for every two points x_1 and x_2

$$f(x_2) \geq f(x_1) + \nabla f(x_1)(x_2 - x_1)$$

or for a twice differential function

$$\nabla^2 f \succeq 0$$

and the function is strictly convex if the inequalities hold strictly.

We illustrate these definitions with an example.

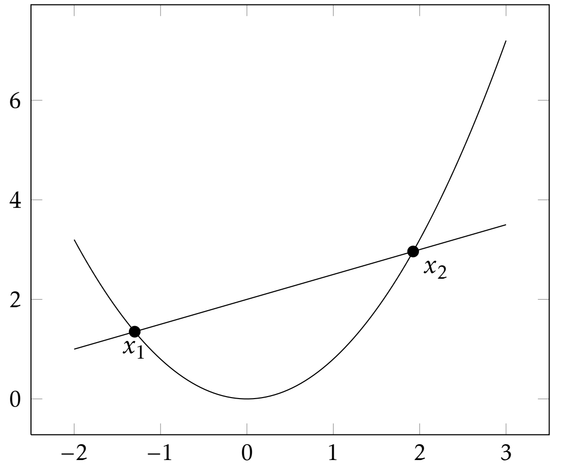

Figure 2.1: *An example of a convex function and its relation to the straight line that passes through two arbitrary points. The curve segment of the convex function between the two points always lies below the line.*

2.3 Example: Convex functions

Using the definition of convexity we can see that the norm function, $f(x) = \|x\|$, is convex. This because

$$\begin{aligned} f(\gamma x_1 + (1-\gamma)x_2) &= \left\|\gamma x_1 + (1-\gamma)x_2\right\| \\ &\leq \left\|\gamma x_1\right\| + \left\|(1-\gamma)x_2\right\| \\ &= \gamma \left\|x_1\right\| + (1-\gamma)\left\|x_2\right\| \\ &= \gamma f(x_1) + (1-\gamma) f(x_2) \end{aligned}$$

As another example, consider a quadratic function $f(x) = x^T Qx + q^T x + c$. Differentiating this function twice we have

$$\nabla^2 f = Q \succeq 0$$

This shows that a quadratic function is convex if and only if Q is positive semidefinite.

Affine functions, $f(x) = a^T x + b$, the negative logarithm, $f(x) = -\log x$, and the max-function, $f(x) = \max\{x_1, x_2, \ldots, x_n\}$ are some other important examples of convex functions.

The constraints in the optimization problem form a set of feasible values of x, a set that must be convex in order for the whole optimization problem to be convex.

2.4 Definition. A set, $\mathcal{X}$, is said to be convex if and only if, for any two points x_1 and x_2 in $\mathcal{X}$, all points on the line between x_1 and x_2 also belong to the set $\mathcal{X}$, i.e., if

$$x_1, x_2 \in \mathcal{X} \Rightarrow \gamma x_1 + (1-\gamma)x_2 \in \mathcal{X} \;\forall\; 0 \leq \gamma \leq 1$$

then the set $\mathcal{X}$ is convex.

From this definition of convex sets we can draw the important conclusion that the intersection between any two (or more) convex sets is also a convex set.

2.5 Example: Convex sets

A closed halfspace in $\mathbb{R}^n$ is defined as

$$\left\{x \in \mathbb{R}^n \mid a_i^T x \leq b_i\right\}$$

Then the intersection of a number of such halfspaces constitutes a convex set, see Figure 2.2. We write this as

$$\mathcal{P} = \left\{x \in \mathbb{R}^n \mid \bigcap_i a_i^T x \leq b_i\right\}$$

or equivalently as

$$\mathcal{P} = \{x \in \mathbb{R}^n \mid Ax \leq b\}$$

where A has a_i^T as its i:th row and b has b_i as its i:th element. Such an intersection is called a *Polyhedron* if it is unbounded and a *Polytope* if it is bounded.

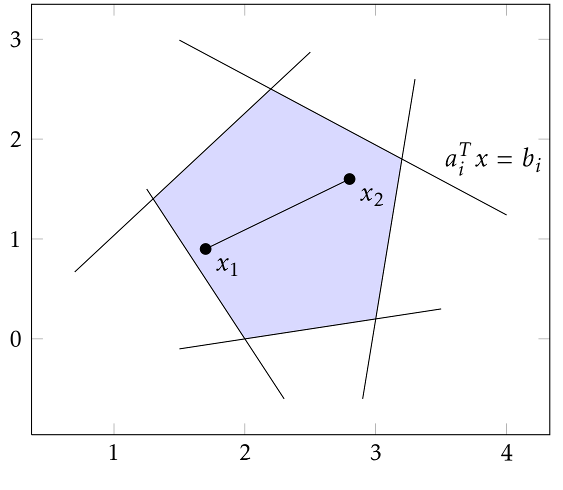

Figure 2.2: *The figure shows a polytope, the shaded area, which is the intersection of five halfspaces (indicated with the lines $a_i^T x = b_i$) where each halfspace constitutes one edge of the polytope and each vertex is the intersection of two halfspaces. Additionally the figure shows two arbitrary points in the polytope and the line connecting them. It is evident that the entire line belongs to the set for any two points, hence the polytope is a convex set.*

Convex optimization problems also called *convex programs* are generally divided into several different *standard forms* such as, e.g., *Linear Programs* (LP's), *Semidefinite Programs* (SDP's), *Geometric programs* (GP's) and *Quadratic programs* (QP's). In this thesis we will mostly consider QP 's since, as we will see in Chapter 3, the control problems that we consider can very often be formulated as QP's.

A QP has a convex quadratic objective function, f_0, and affine constraint functions, f_i and g_i

$$\underset{x}{\text{minimize}}\ x^T Qx + q^T x + c \tag{2.2a}$$
$$\text{s.t.}$$
$$Fx \leq b \tag{2.2b}$$
$$Gx = h \tag{2.2c}$$

The constraint functions form a feasible set defined by the intersection of the polytope, $Fx \leq b$, and the hyperplane $Gx = h$.

2.6 Example: Discrete time LQ controller

The discrete time LQ problem is given by

$$\underset{u_i, x_i}{\text{minimize}} \sum_{i=0}^{\infty} x_i^T Q x_i + u_i^T R u_i$$

where the states x_i have to satisfy the state equation

$$x_{i+1} = Ax_i + Bu_i, \quad x_0 = x(0)$$

We can see this as an (infinite dimensional) equality constrained QP in the variables x_i and u_i.

One great advantage and fundamental property of convex optimization problems which makes them very useful is given in the following theorem.

2.7 Theorem. *If x^* is a local minimum of a convex optimization problem (2.1), then it is also a global minimum of the problem. Furthermore if f_0 is strictly convex, then the minimum is unique.*

Proof: Let x^* be a local minimum of f_0, i.e.,

$$f_0(x^*) = \inf_x \{f_0(x) \mid x \in \mathcal{X}, \|x - x^*\|_2 \leq R\}$$

Now assume that there exist a feasible $\hat{x}$, with $\|\hat{x} - x^*\|_2 > R$, such that $f_0(\hat{x}) < f_0(x^*)$. Since both $\hat{x}$ and x^* are feasible there exist a feasible point as the convex combination of $\hat{x}$ and x^*

$$\tilde{x} = \theta\hat{x} + (1-\theta)x^*$$

With $\theta = \frac{R}{2\|\hat{x}-x^*\|_2} < \frac{1}{2}$ we have $\|\tilde{x} - x^*\|_2 = \theta\,\|\hat{x} - x^*\|_2 = R/2 < R$ and by convexity of f_0 we have

$$f_0(\tilde{x}) \leq \theta f_0(x^*) + (1-\theta) f_0(\hat{x}) < f_0(x^*)$$

but this contradicts the assumption that $f_0(x^*)$ was the minimum within the neighborhood of radius R, hence $\hat{x}$ cannot exist and x^* is the global minimum.

To show uniqueness of the solution assume instead that $f_0(x^*) = f_0(\hat{x})$ and that f_0 is strictly convex. Then it directly follows that

$$f_0(\tilde{x}) < \theta f_0(x^*) + (1-\theta) f_0(\hat{x}) = f_0(x^*)$$

which also contradicts the assumption that x^* (and $\hat{x}$) are minimum of f_0. □

A convex optimization problem is said to be *unbounded below* if the optimal objective value is $f_0^* = -\infty$ and *infeasible* if it is $f_0^* = \infty$.

In order to derive necessary and sufficient conditions for a point x^* to be the global minimum of a convex representation of the problem (2.1) we need to consider something called *Duality*.

2.1.2 Duality

Let us start by defining the *Lagrangian function* for the optimization problem (2.1) as

$$\mathcal{L}(x, \lambda, \nu) = f_0(x) + \sum_{i=1}^{m} \lambda_i f_i(x) + \sum_{i=1}^{p} \nu_i g_i(x)$$

The variables λ_i and ν_i are the so called *dual variables* or *Lagrange multipliers*. We also define the *Dual function* as

$$\mathcal{D}(\lambda, \nu) = \inf_{x \in \mathcal{S}} \mathcal{L}$$

where $\mathcal{S} = \left(\bigcap_{i=0}^{m} \operatorname{dom}(f_i)\right) \bigcap \left(\bigcap_{i=1}^{p} \operatorname{dom}(g_i)\right)$ is the domain of the problem. It can easily be shown that for $\lambda \geq 0$ the dual function fulfills $\mathcal{D}(\lambda, \nu) \leq f_0^*$ for all $x \in \mathcal{S}$. So the dual function defines a global lower bound for the optimal value of the primal problem (2.1).

From this we define the *Dual problem* as

$$\begin{aligned} &\underset{\lambda, \nu}{\text{maximize}}\ \mathcal{D}(\lambda, \nu) \\ &\text{s.t.} \\ &\lambda \geq 0 \end{aligned}$$

with the additional implicit constraint that the dual function must be bounded from below, i.e., $\mathcal{D}(\lambda, \nu) > -\infty$. The optimal value, $\mathcal{D}^*$, to the dual problem is the best lower bound for the primal problem. In the case when this best bound is tight, i.e., $\mathcal{D}^* = f_0^*$, we say that *Strong duality* holds. In the practical cases we will consider in this thesis (e.g., for most convex problems) we can assume that strong duality holds. This dual problem is one of the key details in the derivation of the controller in chapter 4.

2.8 Example: Dual problem to a LP problem

Consider the following LP maximization problem

$$\underset{x}{\text{maximize}}\ a^T x + b \quad \text{s.t.} \quad F^T x \leq g$$

This is equivalent to the minimization problem

$$\underset{x}{\text{minimize}}\ -(a^T x + b) \quad \text{s.t.} \quad F^T x \leq g$$

The Lagrangian is then given by

$$\mathcal{L}(x, \lambda) = -(a^T x + b) + \lambda^T (F^T x - g)$$

and the dual function

$$\begin{aligned}\mathcal{D}(\lambda) &= \inf_x \mathcal{L} \\ &= \inf_x\ -\lambda^T g - b + (F\lambda - a)^T x \\ &= \begin{cases} -\lambda^T g - b & \text{if} \quad F\lambda - a = 0 \\ -\infty & \text{otherwise} \end{cases}\end{aligned}$$

The dual problem is then given by

$$\begin{aligned} \underset{\lambda}{\text{maximize}}\ & -(g^T \lambda + b) \\ \text{s.t.} & \\ \lambda &\geq 0 \\ F\lambda - a &= 0 \end{aligned}$$

We now have the tools to characterize the optimal solution, x^*, to a convex optimization problem. As stated above, for most convex problems, strong duality hold, i.e., $\mathcal{D}(\lambda^*, \nu^*) = f_0(x^*)$. The definition of $\mathcal{D}$ then gives

$$f_0(x^*) = \mathcal{D}(\lambda^*, \nu^*) = f_0(x^*) + \sum_{i=1}^{m} \lambda_i^* f_i(x^*) + \sum_{i=1}^{p} \nu_i^* g_i(x^*)$$

and since $g_i(x^*) = 0$ it must hold that

$$\sum_{i=1}^{m} \lambda_i^* f_i(x^*) = 0 \tag{2.3}$$

This means that if the constraint is not active then the corresponding dual variable must be equal to zero, i.e., if $f_i(x^*) < 0$ then $\lambda_i = 0$ and if the constraint is active, i.e., $f_i(x^*) = 0$ then can the dual variable be nonzero. Note also that both λ_i and f_i can be zero at the same time (then f_i is called a *weakly active* constraint).

The condition (2.3) is called *complementary slackness.*

Furthermore since x^* minimizes the Lagrangian it must also hold that the gradient is zero,

$$\nabla \mathcal{L} = \nabla f_0(x^*) + \sum_{i=1}^{m} \lambda_i^* \nabla f_i(x^*) + \sum_{i=1}^{p} \nu_i^* \nabla g_i(x^*) = 0 \tag{2.4}$$

So to summarize, for a point x^* to be optimal for a convex instance of the optimization problem (2.1) it is necessary and sufficient that the following conditions hold

$$\nabla f_0(x^*) + \sum_{i=1}^{m} \lambda_i^* \nabla f_i(x^*) + \sum_{i=1}^{p} \nu_i^* \nabla g_i(x^*) = 0 \tag{2.5}$$

$$f_i(x^*) \leq 0 \tag{2.6}$$

$$g_i(x^*) = 0 \tag{2.7}$$

$$\lambda^* \geq 0 \tag{2.8}$$

$$\sum_{i=1}^{m} \lambda_i^* f_i(x^*) = 0 \tag{2.9}$$

These conditions are called the *Karush-Kuhn-Tucker conditions,* (KKT). The conditions (2.6) and (2.7) require x^* to be primal feasible and the condition (2.8) requires λ to be feasible for the dual problem.

Note that for the KKT conditions to be sufficient conditions for optimality strong duality must hold. We have not discussed under what conditions strong duality hold for a convex program; we merely state that for our applications strong duality does hold and refer the details to Boyd and Vandenberghe [2004].

We will use the KKT conditions in Section 3.2.5 to derive an explicit formulation of the model predictive controller.

2.1.3 Nonconvex optimization

In the previous section we did not discuss how to actually solve a convex optimization problem. There are different methods to solve such problems, e.g., by using the so called gradient methods which generally searches for the optimal point in the direction of the negative gradient. Due to the special properties of convex problems described in Theorem 2.7, this search will eventually lead us to the global optimal solution. However if the problem is nonconvex, then one can not use local information to find the global optimal solution, this makes the nonconvex problems harder to solve and requires us to use more complex solution algorithms.

2.9 Example: Controller for a nonlinear system

Let us consider the same objective function as in Example 2.6, but now with nonlinear system dynamics

$$\begin{aligned} \underset{u_i, x_i}{\text{minimize}} \quad & \sum_{i=0}^{\infty} x_i^T Q x_i + u_i^T R u_i \\ & x_{i+1} = f(x_i, u_i), \quad x_0 = x(0) \end{aligned}$$

Since the equality constraint now consists of a nonlinear function, the optimization problem is no longer convex (that would require the equality constraint to be affine).

In Chapter 5 we will look more into some methods to overcome this difficulty and present a new method that approximates a general nonlinear program with an easily solvable QP.

The techniques to solve nonconvex problems can be divided into two distinct categories, local techniques and global techniques.

The local techniques, such as, e.g., *Broyden–Fletcher–Goldfarb–Shanno* algorithm (BFGS) or *Sequential Quadratic Programming* (SQP), use a convex approximation of the problem around an initial guess of the optimum to calculate the search direction. This will lead the algorithm to converge to a local optimum, which unfortunately can be far off from the true global solution. The local solutions are obviously very dependent on the initial guess of the optimal point. The benefit of local methods in comparison to the global techniques is that they are relatively faster.

In contrast to local methods, techniques that find the true global optimal solution are extremely computationally expensive and even for modest scale problems they can take several hours to converge to the global solution. The global solution methods are either *heuristic* or *nonheuristic* in their nature. One nonheuristic method that has received a lot of attention is the *Branch and bound* method. The basic idea of the branch and bound method is to partition the feasible set into convex partitions and for each of these partitions calculate an upper and lower bound on the optimal value. The upper bound could be found by using either one of the local optimization techniques described or by simply selecting any feasible point. The lower bound can be found by solving the (always convex) dual problem or by some convex relaxations. The upper and lower bounds are compared for each of the partitions and if the lower bound of one partition has a higher value than the upper bound in some other partition, this partition can surely not contain the global optimum, and hence it can be discarded. The algorithm then continues by splitting the best partitions into smaller and smaller partitions repeating the computations and comparisons of the bounds until all partitions are discarded but one. The name, branch and bound, comes from the fact that the partitioning branches out like a search tree. One big advantage with the branch and bound method is that it can quantify the level of suboptimality.

This in contrast to the local methods that are unable to provide us with such quantities.

Another nonheuristic method that has gained significant popularity in the last decade is the so called *Semidefinite relaxation*. The relaxation of a general optimization problem

$$\underset{x}{\text{minimize}} \ \{f_0(x) \mid x \in \mathcal{X}\}$$

is another optimization problem

$$\underset{x}{\text{minimize}} \ \left\{\tilde{f}_0(x) \mid x \in \tilde{\mathcal{X}}\right\}$$

such that $\tilde{f}_0(x) \leq f_0(x), \forall x \in \mathcal{X}$ and $\mathcal{X} \subseteq \tilde{\mathcal{X}}$. In semidefinite relaxation the resulting relaxed optimization problem is an SDP. Lasserre [2001] propose a method to find the global solution to a nonconvex problem by solving a sequence of semidefinite relaxations of the original problem.

There also exist several heuristic methods for global optimization such as, e.g., *Simulated Annealing*, *Direct Monte-Carlo Sampling* and *Particle Swarm Optimization*. These methods essentially performs a guided random search converging to the globally optimal solution if given sufficient time.

2.2 Convex polytopic geometry

In this section, we will consider a branch of mathematics which is concerned with geometric problem solving and computation with convex sets of different dimension such as, e.g., points, lines, hyperplanes and polytopes. We will detail fundamental properties and some basic algebraic calculations with polytopic sets which are particularly interesting from the perspective of our application. For a more in depth description of these topics we refer to Grünbaum [2003] and for more general computational geometry problems we refer the reader to the work of de Berg et al. [2008].

Let us first recall the definition of a polytope from the previous section. We define a polytope as the bounded intersection of a finite number of halfspaces

$$\mathcal{P} = \{x \mid Ax \leq b\}$$

We will refer to this as the *Halfspace representation*.

Using the halfspace representation we can show that the intersection of two (or more) polytopes is a new polytope (unless it is the empty set). Consider two polytopes

$$\mathcal{P}_1 = \{x \mid A_1 x \leq b_1\}, \quad \mathcal{P}_2 = \{x \mid A_2 x \leq b_2\}$$

then the intersection can be written as the polytope

$$\mathcal{P}_3 = \mathcal{P}_1 \bigcap \mathcal{P}_2 = \left\{x \mid \begin{bmatrix} A_1 \\ A_2 \end{bmatrix} x \leq \begin{bmatrix} b_1 \\ b_2 \end{bmatrix}\right\}$$

In this description of the polytope there can exist a number of redundant hyperplanes which can be removed through solving a set of LPs in order to have a minimal representation of the polytope. Algorithms for doing this can be found in Baotic [2005].

For a set of N points, $X = \{x_1, x_2, \ldots, x_N\}$, the *Convex hull*, $conv(X)$, is defined as the smallest polytope that contains all N points. This can be represented as the convex combination of all points

$$conv(X) = \sum_{i=1}^{N} \beta_i x_i, \quad \forall \;\; 0 \le \beta_i \le 1, \; \sum_{i=1}^{N} \beta_i = 1$$

Similar to the halfspace representation there exist a *vertex representation* of a polytope which is the minimal representation of its convex hull, i.e.,

$$\mathcal{P} = \sum_{i=1}^{\nu_p} \beta_i v_i$$

where v_i are those x_i that constitute the vertices of the polytope and ν_p is the number of vertices in the polytope, see Figure 2.3. Algorithms for extracting the convex hull from a set of points can be found in de Berg et al. [2008].

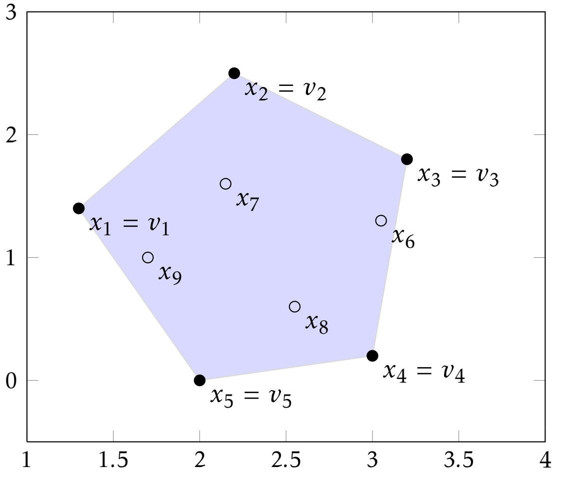

Figure 2.3: *An example of a set of nine points x_i, and the convex hull of those points (the shaded area). The points that constitute the vertices of the convex hull are the vertex representation of the shaded polytope.*

The task of going from the halfspace representation to the vertex representation of a polytope is known as the *vertex enumeration problem* and the opposite (vertex to halfspace representation) is known as the *facet enumeration problem*. These are complicated and very computational expensive tasks and in Avis and Fukuda [1992] the authors present one algorithm for performing those.

Let us now continue with some basic polytope operations which will be useful in the following chapters.

For two polytopic sets, $\mathcal{P}_1$ and $\mathcal{P}_2$, the *Minkovsky sum* is the set

$$\mathcal{P}_1 \oplus \mathcal{P}_2 = \{x_1 + x_2 \mid x_1 \in \mathcal{P}_1,\ x_2 \in \mathcal{P}_2\}$$

and correspondingly the *Pontryagin difference* of two sets is the set

$$\mathcal{P}_1 \ominus \mathcal{P}_2 = \{x_1 \mid x_1 + x_2 \in \mathcal{P}_1, \forall x_2 \in \mathcal{P}_2\}$$

An illustration of these two polytope operations are shown in Figure 2.4.

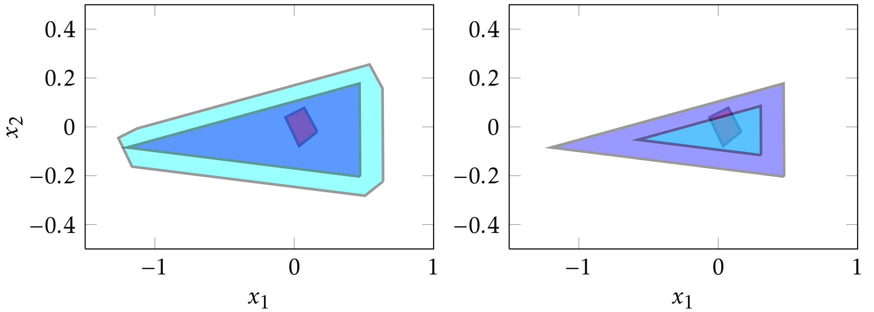

Figure 2.4: *The left axis shows an example of the Minkovsky sum. The two most inner polytopes are $\mathcal{P}_1$ and $\mathcal{P}_2$ respectively. The outer polytope is the Minkovsky sum of $\mathcal{P}_1$ and $\mathcal{P}_2$. The right axis shows the same two polytopes and the Pontryagin difference as the smaller triangular polytope.*

Furthermore, the *Cartesian product* of the two sets is

$$\mathcal{P}_1 \times \mathcal{P}_2 = \{(x_1, x_2) \mid x_1 \in \mathcal{P}_1,\ x_2 \in \mathcal{P}_2\}$$

If the dimension of $\mathcal{P}_1$ is $\mathbb{R}^n$ and $\mathcal{P}_2$ is $\mathbb{R}^m$, then the dimension of the product is $\mathbb{R}^{m+n}$.

2.10 Definition. For a polytopic set $\mathcal{P}$ with the halfspace representation

$$\mathcal{P} = \{x \mid Ax \leq b\}$$

we define

- the *scaling* of the set with a positive scalar α as

$$\alpha\mathcal{P} = \{x \mid Ax \leq \alpha b\}$$

- the *translation* of the set to an arbitrary point y as

$$\mathcal{P}(y) = \mathcal{P} \oplus y = \{x \mid A(x - y) \leq b\}$$

Note that the translation of a polytope to an arbitrary point is the same as the Minkovsky sum of the set, $\mathcal{P}$, and a set consisting of only one point, y. It is straightforward to derive the vertex representation of scaling and translation as

$$\alpha\mathcal{P} = \alpha\sum_{i=1}^{v_p}\beta_i v_i$$

and

$$\mathcal{P}(y) = y + \sum_{i=1}^{v_p}\beta_i v_i$$

Given a polytope, $\mathcal{P}$, the interior of the set, int($\mathcal{P}$), is defined as all those points $y \in \mathcal{P}$ for which there exists an $\epsilon > 0$ such that for any point x in $\mathcal{P}$, y is within an ϵ-radius of x.

$$\text{int}(\mathcal{P}) = \left\{y \mid \left\|x - y\right\|_2 \leq \epsilon, x \in \mathcal{P}\right\} \subseteq \mathcal{P}$$

One could also more loosely describe these points as all points in $\mathcal{P}$, except the border. Since the interior is an open set it is practical to work with the closely related ϵ-*interior* to a set.

2.11 Definition. For a given constant $0 \leq \epsilon \leq 1$ and set $\mathcal{P}$ containing the origin, let $\text{int}_\epsilon(\mathcal{P})$ denote the ϵ-interior of $\mathcal{P}$, i.e.,

$$\text{int}_\epsilon(\mathcal{P}) = (1-\epsilon)\mathcal{P} = \{x \mid Ax \leq (1-\epsilon)b\}$$

This is a key detail in the stability proof of our controller in Chapter 4.

The affine transformation, $F(\,\cdot\,)$, of a set, $\mathcal{P}$, defined by a matrix A and vector b is an affine map of all elements in $\mathcal{P}$.

$$F(\mathcal{P}) = \{Ax + b \mid x \in \mathcal{P}\}$$

and we will adopt the short hand notation $F(\mathcal{P}) = A\mathcal{P} + b$ for this. Additionally there exist an inverse affine mapping defined through

$$F^{-1}(\mathcal{P}) = \{x \mid Ax + b \in \mathcal{P}\}$$

This inverse mapping is central to the calculation of *invariant sets* which is an important detail of the stability of model predictive controllers.

Note that the scaling and translation defined above are just special cases of the affine mapping.

We argued in the beginning of this section that the intersection of two polytopes is a new polytope, but in fact, all the polytope operations we have defined will result in a new polytope [Boyd and Vandenberghe, 2004].

For our intended application it is also needed to be able to calculate the center point of a polytope. The center point that we consider is the so called *Chebychev center*, which is the center point of the largest ball (Chebychev ball) that can be inscribed in the polytope.

A *Ball* is all points x that are within a radius R from a center point x_c

$$\mathcal{B}(x_c, R) = \{x_c + x \mid \|x\|_2 \leq R\}$$

For a polytope $\mathcal{P}$ we find the Chebychev center from the following optimization problem

$$\begin{gathered} \underset{x_c, R}{\text{maximize}}\ R \\ \text{s.t.} \\ \mathcal{B}(x_c, R) \subseteq \mathcal{P} \end{gathered}$$

To see how this abstract formulation can be written as a simple linear program first note that if $\mathcal{B}(x_c, R) \subseteq \mathcal{P}$ it must hold for all facets of the polytope. This means that $a_i^T(x_c + x) \leq b_i,\ \forall i = 1, \ldots, m$, where m is the number of facets in $\mathcal{P}$. Also note that this constraint shall hold for all $\|x\|_2 \leq R$ and so it must hold for the worst case x. Thus we get

$$\sup_x \left\{a_i^T x \mid \|x\|_2 \leq R\right\} = \left\|a_i^T\right\|_2 R$$

and we can write the constraint as

$$a_i^T x_c + \left\|a_i^T\right\|_2 R \leq b_i \quad \forall\ i = 1, \ldots, m$$

The resulting LP becomes

$$\underset{x_c, R}{\text{maximize}}\ R \tag{2.10a}$$

$$\text{s.t.}$$

$$a_i^T x_c + \left\|a_i^T\right\|_2 R \leq b_i \quad \forall\ i = 1, \ldots, m \tag{2.10b}$$

3

Introduction to Model Predictive Control

In this chapter we give the background to the *Model Predictive Control* (MPC) problem formulation and give several variants and extensions of the theory. Section 3.1 gives a short historical background to the development of MPC. In Section 3.2 we derive the MPC controller formulation for linear systems and discuss stability and robustness issues as well as reference tracking. We also derive the explicit controller formulation. Finally in Section 3.3 we briefly discuss the MPC formulation for nonlinear systems.

3.1 Introduction

As described in the introduction, adding constraints to the infinite horizon linear quadratic optimal control problem makes it in general extremely difficult to solve explicitly, i.e., finding an explicit formulation for $\kappa(x)$. Model Predictive Control offers a way to overcome these difficulties by defining an open loop optimal control problem which is easier to solve than the original problem and then iteratively solve it online, using the current state of the system as initial condition, giving a closed loop feedback controller.

The MPC research originated in the early 1960's with the work of Propoi [1963] but in the 1970's and 1980's much of the development came from the process industry with methods such as DMC, GPC, IDCOM, QDMC and SMOC. These techniques were extensively implemented in industry and hence solved real world problems with constraints and nonlinearities, however they did not build on firm theoretical foundations, e.g., they did not guarantee stability and required careful tuning and stable models. In the 1990's, several influential papers were published that rigorously developed the stability theory and reference tracking formulation. A big leap in the theory development occurred in the early 2000's with the development of Explicit MPC which opened up a new area of research within the community. In the last two decades lots of research have been done on nonlinear MPC and its theoretical foundations, and today linear and nonlinear MPC is one of the most promising advanced control techniques to handle constrained multivariable systems.

3.2 Linear MPC

The most common formulation of MPC is the discrete time version and therefore will we limit our discussion to discrete time systems and the discrete time formulation of the MPC controller (with the exception of Chapter 5 where we also discuss continuous time nonlinear systems). Hence let us for the remainder of this section assume that the system dynamics can be described by a discrete time difference equation

$$x_{i+1} = Ax_i + Bu_i \tag{3.1}$$

and the subscript i is short for the time index, i.e., $x_i = x(iT_s)$ where T_s is the sample time of the system.

Consider a discrete time version of the constrained optimal control problem (1.7)

$$\begin{aligned}
\underset{u_i, x_i}{\text{minimize}} \quad & \sum_{i=0}^{\infty} \left(x_i^T Q x_i + u_i^T R u_i \right) && \text{(3.2a)} \\
\text{s.t.} \quad & \\
& x_{i+1} = Ax_i + Bu_i && \text{(3.2b)} \\
& x_i \in \mathcal{X} && \text{(3.2c)} \\
& u_i \in \mathcal{U} && \text{(3.2d)}
\end{aligned}$$

and assume that the sets $\mathcal{X}$ and $\mathcal{U}$ are convex. Then this can be viewed as a general convex optimization problem with an infinite number of decision variables $x_i,\ u_i\ \forall\ i > 0$. Instead of solving the infinite dimensional optimization problem (3.2) it is possible to recast it as a finite problem by splitting the objective function (3.2a) into two parts

$$\underset{u_i, x_i}{\text{minimize}} \quad \sum_{i=0}^{N-1} \left(x_i^T Q x_i + u_i^T R u_i \right) + \sum_{i=N}^{\infty} \left(x_i^T Q x_i + u_i^T R u_i \right)$$

and try to find a way to approximate the last part of the objective function, or at least to bound it from above by some function $\Psi(x_N)$

$$\sum_{i=N}^{\infty} \left(x_i^T Q x_i + u_i^T R u_i \right) \leq \Psi(x_N) \tag{3.3}$$

Then the resulting optimal control problem will have a finite dimension.

$$\underset{u_i, x_i}{\text{minimize}} \sum_{i=0}^{N-1} \left(x_i^T Q x_i + u_i^T R u_i \right) + \Psi(x_N) \tag{3.4a}$$

s.t.

$$x_{i+1} = A x_i + B u_i \quad \forall\ i = 0, \ldots, N-1 \tag{3.4b}$$

$$x_i \in \mathcal{X} \quad \forall\ i = 0, \ldots, N-1 \tag{3.4c}$$

$$u_i \in \mathcal{U} \quad \forall\ i = 0, \ldots, N-1 \tag{3.4d}$$

$$x_N \in \mathcal{T} \tag{3.4e}$$

In order to make the approximation or bound (3.3) valid there might be some extra constraints (3.4e) on the state at the end of the horizon, x_N. We will discuss further how to choose $\Psi(x_N)$ and $\mathcal{T}$ in Section 3.2.1 since it relates closely to stability of the closed loop system.

By solving the optimization problem (3.4) with the current state as x_0, the solution is a sequence of N optimal control inputs u_i^*, denoted $\{u_i^*\}_{i=0}^{N-1}$, for which the objective (3.4a) is minimized. Implementing this sequence of controls as inputs to the real system will result in an open loop controller. However it is widely known that closed loop control is preferred since all real systems suffer from disturbances and all models have errors [Skogestad and Postlethwaite, 2005].

To achieve closed loop control we implement, at each time step k, only the first control signal, u_0^*, in the optimal sequence $\{u_i^*\}_{i=0}^{N-1}$ and then in the next time step, $k+1$, measure the current state and redo the optimization with the new current state as x_0. This strategy is referred to as *Receding Horizon Control* since the control problem is solved over a future horizon which is progressing into the future as time evolves.

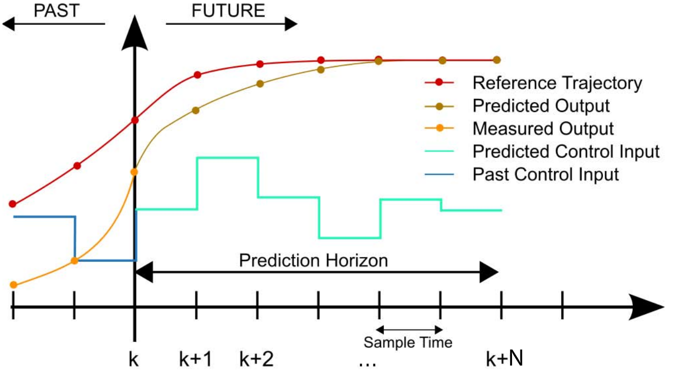

Figure 3.1: *The receding horizon concept of Model Predictive Control (By Martin Behrendt, via Wikimedia Commons).*

This could also be explained as at each discrete time instant $t = kT_s$ (where T_s

is the sample time of the controller) the current state of the system is measured $x(t) = x(kT_s) = x_k$ and the controller predicts the response of the system N time steps into the future (see Figure 3.1). The optimal control input is selected based on the predicted future behavior of the system by solving the optimization problem (3.4). In conclusion, the Model Predictive Control law, $\kappa(x_k)$, is defined by the solution to the following optimization problem

$$\mathcal{J}_k^* = \underset{u_{k+i}, x_{k+i}}{\text{minimize}} \sum_{i=0}^{N-1} \ell(x_{k+i}, u_{k+i}) + \Psi(x_{k+N}) \tag{3.5a}$$

s.t.

$$x_k = x(t) \tag{3.5b}$$

$$x_{k+i+1} = Ax_{k+i} + Bu_{k+i} \quad \forall\ i = 0, \dots, N-1 \tag{3.5c}$$

$$x_{k+i} \in \mathcal{X} \quad \forall\ i = 0, \dots, N-1 \tag{3.5d}$$

$$u_{k+i} \in \mathcal{U} \quad \forall\ i = 0, \dots, N-1 \tag{3.5e}$$

$$x_{k+N} \in \mathcal{T} \tag{3.5f}$$

and by applying the first element, u_k^*, of the optimal solution sequence $\{u_{k+i}^*\}_{i=0}^{N-1}$ as control input, i.e.,

$$\kappa(x_k) = u_k^* \tag{3.5g}$$

In (3.5) we have generalized the notation by replacing the quadratic cost function, $x_{k+i}^T Q x_{k+i} + u_{k+i}^T R u_{k+i}$, with the more general expression, $\ell(x_k, u_k)$. We have also introduced the notation $\mathcal{J}_k^*$ as the optimal objective function value. Note also that in (3.5), k denotes the current time step and i is used to denote the prediction steps into the future. This is only for notational convenience, in a real implementation there is no difference between (3.4) and (3.5). Although $\mathcal{X}$ and $\mathcal{U}$ can be any arbitrary convex sets we will in this thesis mainly consider polytopic constraint sets, i.e.,

$$\mathcal{X} = \{x \mid F_{\mathcal{X}} x \leq b_{\mathcal{X}}\}, \quad \mathcal{U} = \{u \mid F_{\mathcal{U}} u \leq b_{\mathcal{U}}\}$$

The formulation (3.5) is the standard linear MPC formulation and will, with a few exceptions, be used throughout this thesis, but there exist several other formulations with different objective functions and constraint formulations.

3.2.1 Stability

Due to the presence of constraints, the MPC formulation is a nonlinear control problem and to show stability for the closed loop system it is necessary to use *Lyapunov Stability Theory* [Khalil, 2002]. In this section we will give a brief outline of the stability properties for the MPC formulation. We assume in this section that the model of the system is perfect and that there are no disturbances acting on the system.

The infinite horizon MPC formulation (3.2), although it, in general, is impossible to solve, is stabilizing if $\ell(x_k, u_k) > 0\ \forall\ x_k \neq 0,\ u_k \neq 0$ and $\ell(0,0) = 0$. To see this we first observe that if we have an optimal solution sequence at time k, then following from the *Principle of Optimality* the optimal sequence at time $k+1$ is equal to the part of the optimal sequence at time k starting from index $k+1$. Then it follows that the optimal cost at time $k+1$, is

$$\begin{aligned}
\mathcal{J}^*_{k+1} &= \sum_{i=0}^{\infty} \ell(x^*_{k+1+i}, u^*_{k+1+i}) \\
&= \sum_{i=0}^{\infty} \ell(x^*_{k+i}, u^*_{k+i}) - \ell(x^*_k, u^*_k) \\
&= \mathcal{J}^*_k - \ell(x^*_k, u^*_k) \\
&\leq \mathcal{J}^*_k
\end{aligned}$$

From this we can conclude that if the stage cost $\ell(x_k, u_k)$ is strictly positive for all $x_k \neq 0$ and $u_k \neq 0$, then the objective function (3.2a) is strictly decreasing as k increases.

Furthermore since $\mathcal{J}^*_k$ is bounded below by zero it must follow that $\ell(x^*_k, u^*_k) \to 0$ as $k \to \infty$ and hence $x^*_k \to 0$ and $u^*_k \to 0$.

For the finite time horizon formulation (3.5) this type of argument can not be used since the horizon is progressing over time, taking new information into account. The solution at time $k+1$ can therefore be very different from that at time k. For this case one has to carefully chose the function $\Psi(x_{k+N})$ and the terminal constraint $x_{k+N} \in \mathcal{T}$ to ensure stability. There exist several such choices and we will give a brief review of some different techniques proposed in the literature. For a more extensive survey we refer to the excellent review paper by Mayne et al. [2000].

If the system is stable, a pragmatic approach would be to assume all control signals $u_{k+i} = 0$ for $i \geq N$ and calculate (3.3) as the uncontrolled response of the system starting from the state x_{k+N}. This is approach proposed in [Muske and Rawlings, 1993] and [Rawlings and Muske, 1993]. Similarly in [Bitmead et al., 1990] the authors use a quadratic norm cost as upper bound for (3.3)

$$\Psi(x_{k+N}) = x^T_{k+N} P x_{k+N}$$

where the matrix P is the stationary solution to the Riccati difference equation. Keerthi and Gilbert [1988] proposes instead to impose a new constraint, $x_{k+N} = 0$ (i.e., $\mathcal{T} = 0$), which for linear systems will give $\Psi(x_{k+N}) = 0$. With this setup they show that the closed loop system is stable. Additionally they show that the cost of the finite horizon optimal control problem (3.5) approaches that of the infinite horizon problem (3.2) as N increases.

A natural relaxation of this approach would be to constrain the final state x_{k+N}, not to the origin, but instead to some small region, $\mathcal{T}$, around the origin in which a local controller, $u_k = \kappa(x_k)$, can take over and drive the system to the origin. This approach is called *Dual Mode Control* since, in theory, the controller is divided in two modes, one solving the receding horizon control problem and one that is a local control law. However in practice the local controller is never implemented, it is only a theoretical construction to show stability. Combining this approach with a cost $\Psi(x_{k+N})$ which is the cost for the local controller to drive the states to the origin is, at least in academia, the most widely used formulation of the MPC controller (3.5).

Before we can state the formal stability theorem of the dual mode MPC, we first need a definition which can also be found in Blanchini [1999].

3.1 Definition. The set $\mathcal{T} \subset \mathbb{R}^n$ is said to be *positively invariant* for a system $x_{k+1} = f(x_k)$ if for all $x_k \in \mathcal{T}$ the solution $x_{k+i} \in \mathcal{T}$ for $i > 0$.

With this definition in place we are now ready to state the main stability theorem.

3.2 Theorem (Mayne et al. [2000]). *If the system (3.1) is controlled with the MPC law (3.5), where $\ell(x_k, u_k) \geq c(|(x_k, u_k)|)^2$ and $\ell(0,0) = 0$. Then the closed loop system is stable and asymptotically converges to the origin if the following conditions hold:*

1. *$\mathcal{T}$ is a positively invariant (see def. 3.1) set of the system (3.1) controlled with the feedback $u_k = \kappa(x_k)$ where $\kappa(x_k) \in \mathcal{U}\ \forall\ x_k \in \mathcal{T}$*
2. *$\mathcal{T} \subseteq \mathcal{X}$ with $0 \in \mathcal{T}$*
3. *$\Delta\Psi(x_k) + \ell(x_k, \kappa(x_k)) \leq 0,\ \forall\ x_k \in \mathcal{T}$, where $\Delta\Psi(x_k) = \Psi(x_{k+1}) - \Psi(x_k)$.*

The first two conditions ensures that the problem is *recursively feasible*, i.e., given that there exist a solution at one time, then there will exist a solution for all future time steps. The last condition ensures that the system asymptotically converges to the origin. The properties of Ψ that fulfill condition 3 can be obtained if Ψ is chosen to be a Lyapunov function upper bounding the infinite horizon cost when using the controller $\kappa(x_k)$.

To show the convergence let us assume that there exists a solution at time k with an optimal cost $\mathcal{J}_k^*$. Then, at the following time step we apply the feasible but possibly suboptimal control sequence $\{\hat{u}_{k+i}\}_{i=1}^{N} = \{u_{k+1}^*, \ldots, u_{k+N-1}^*, \kappa(x_{k+N}^*)\}$. Then the suboptimal cost, $\mathcal{J}_{k+1}$, is

$$
\begin{aligned}
\mathcal{J}_{k+1} &= \sum_{i=0}^{N-1} \ell(x_{k+1+i}^*, u_{k+1+i}) + \Psi(x_{k+1+N}) \\
&= \sum_{i=0}^{N-2} \ell(x_{k+1+i}^*, u_{k+1+i}^*) + \ell(x_{k+N}^*, u_{k+N}) + \Psi(x_{k+1+N}) \\
&\quad + \ell(x_k^*, u_k^*) - \ell(x_k^*, u_k^*) + \Psi(x_{k+N}^*) - \Psi(x_{k+N}^*)
\end{aligned}
$$

$$
\begin{aligned}
&= \underbrace{\sum_{i=0}^{N-1} \ell(x^*_{k+i}, u^*_{k+i}) + \Psi(x^*_{k+N})}_{\mathcal{J}^*_k} \\
&\quad - \ell(x^*_k, u^*_k) + \ell(x^*_{k+N}, u_{k+N}) + \Psi(x_{k+1+N}) - \Psi(x^*_{k+N}) \\
&= \mathcal{J}^*_k \underbrace{-\ell(x^*_k, u^*_k)}_{<0} + \underbrace{\ell(x^*_{k+N}, \kappa(x_{k+N})) + \Delta\Psi(x_{k+N})}_{\leq 0} \\
&< \mathcal{J}^*_k
\end{aligned}
$$

The last inequality follows from the third property of the theorem and from this we conclude that the optimal cost at time $k+1$ fulfills

$$\mathcal{J}^*_{k+1} \leq \mathcal{J}_{k+1} < \mathcal{J}^*_k$$

In other words, $\mathcal{J}^*_k$ is strictly decreasing as long as $x^*_k \neq 0$ and $u^*_k \neq 0$. Hence $x^*_k \to 0$ and $u^*_k \to 0$.

The above derivation assumes that there exists a solution to (3.5) at time $k+1$, i.e., that $\hat{\mathbf{u}}$ is feasible, this is not trivial and it must be proven that there always exist a solution to the open loop problem, so called *recursive feasibility*. To show this let $\mathcal{X}_N$ be the set of x where (3.5) is feasible. Assume $x_k \in \mathcal{X}_N$ with an optimal solution given by the sequence $\{u^*_{k+i}\}_{i=0}^{N-1}$ and with predicted optimal state trajectory $\{x^*_{k+i}\}_{i=1}^{N}$.

At the next time step $\{\hat{u}_{k+i}\}_{i=1}^{N} = \{u^*_{k+1}, u^*_{k+2}, \ldots, u^*_{k+N-1}, \kappa(x^*_{k+N})\}$ is a feasible control sequence, since $x^*_{k+N} \in \mathcal{T}$ and hence $\kappa(x^*_{k+N}) \in \mathcal{U}$ according to Condition 1 of Theorem 3.2. The new, possibly suboptimal, state sequence at time $k+1$ is $\{\hat{x}_{k+i}\}_{i=2}^{N+1} = \{x^*_{k+2}, x^*_{k+3}, \ldots, x^*_{k+N}, x_{k+N+1}\}$ where $x_{k+N+1} = Ax^*_{k+N} + B\kappa(x^*_{k+N})$. Since $\mathcal{T}$ is positively invariant w.r.t the system $x_{k+1} = Ax_k + B\kappa(x_k)$, it follows that x_{k+N+1} stays in $\mathcal{T}$, i.e., the terminal state constraint $x_{k+N+1} \in \mathcal{T}$ is satisfied. This argument can be used recursively and shows that the problem is feasible at all k.

It should be noted that the property of recursive feasibility require that the model of the system is perfect and that no disturbances act on the system. We will discuss more on how to ensure recursive feasibility in the presence of model uncertainties and disturbances in Section 3.2.4.

Sznaier and Damborg [1987] proposes a dual mode formulation for linear systems where the local controller $\kappa(x_k) = -Kx_k$ and $\Psi(x_{k+N}) = x^T_{k+N}Px_{k+N}$ is the solution to a discrete time version of the unconstrained infinite horizon LQ problem (1.5) and (1.6), i.e.,

$$K = (R + B^T PB)^{-1} B^T PA \tag{3.6a}$$

where P is the solution to the Riccati-equation

$$(A - BK)^T P(A - BK) - P = -Q - K^T RK \tag{3.6b}$$

The authors show that by iteratively solving (3.5) for an increasing horizon, N, until (3.4e) is satisfied (where $\mathcal{T}$ is an invariant set of the system controlled with the local controller, see 3.1) then the solution is also the solution to the constrained LQ problem (3.2). Later versions of this approach have explicitly incorporated (3.4e) as a constraint and use a fix horizon, N.

To see that this choice of local controller and terminal state penalty indeed satisfy the third property of Theorem 3.2 we note that since

$$\ell(x_k, u_k) = x_k^T Q x_k + u_k^T R u_k$$

$$\Psi(x_{k+N}) = x_{k+N}^T P x_{k+N}$$

$$x_{k+1} = (A - BK)x_k$$

we obtain

$$\begin{aligned}
&\ell(x_k^*, \kappa(x_k^*)) + \Psi(x_{k+1+N}) - \Psi(x_{k+N}^*) \\
&= x_{k+N}^{*T} Q x_{k+N}^* + (K x_{k+N}^*)^T R K x_{k+N}^* + x_{k+1+N}^T P x_{k+1+N} - x_{k+N}^{*T} P x_{k+N}^* \\
&= x_{k+N}^{*T} Q x_{k+N}^* + x_{k+N}^{*T} K^T R K x_{k+N}^* + x_{k+N}^{*T} (A - BK)^T P (A - BK) x_{k+N}^* \\
&- x_{k+N}^{*T} P x_{k+N}^* \\
&= x_{k+N}^{*T} \underbrace{\left(Q + K^T R K + (A - BK)^T P (A - BK) - P\right)}_{(3.6b)} x_{k+N}^* \\
&= 0
\end{aligned}$$

where the last equality follows from the fact that P and K are the solution to the LQ problem and hence satisfy (3.6b). This shows that the dual mode MPC formulation with the LQ solution as local controller and terminal state penalty is recursively feasible, stabilizes a linear system and is optimal in the sense that it solves the infinite horizon constrained LQ problem when the final state constraint is inactive. These desirable properties makes this formulation very attractive.

3.2.2 Reference tracking

When tracking a reference signal, i.e., the so called servo problem, the system shall not converge to the origin but settle at some steady state (x_r, u_r) different from the origin, yielding the desired output.

At this steady state it must hold that

$$y_k = r \tag{3.7}$$

where r is the external reference to be followed and since it is a steady state it must hold that

$$x_{k+1} = x_k = x_r \tag{3.8}$$

Given a controllable linear discrete time system

$$x_{k+1} = Ax_k + Bu_k \tag{3.9a}$$

$$y_k = Cx_k + Du_k \tag{3.9b}$$

inserting (3.7) and (3.8) into (3.9) we obtain

$$x_r = Ax_r + Bu_r$$

$$r = Cx_r + Du_r$$

Rearranging the equations gives the relation

$$\begin{bmatrix} A - I & B \\ C & D \end{bmatrix} \begin{bmatrix} x_r \\ u_r \end{bmatrix} = \begin{bmatrix} 0 \\ r \end{bmatrix} \tag{3.10}$$

This relation determine the steady state x_r and control u_r given a reference input r. However the combination of input and steady state that result in $y_k = r$ might be non-unique, i.e., if the matrix on the left hand side of (3.10) is singular. In that case, a reasonable choice is to choose a steady state which is the minimal norm input, which can be formulated as a convex problem [Muske and Rawlings, 1993, Meadows and Badgwell, 1998]

$$\underset{x_r, u_r}{\text{minimize}} \; u_r^T W u_r \tag{3.11a}$$

$$\text{s.t.}$$

$$\begin{bmatrix} A - I & B \\ C & D \end{bmatrix} \begin{bmatrix} x_r \\ u_r \end{bmatrix} = \begin{bmatrix} 0 \\ r \end{bmatrix} \tag{3.11b}$$

$$u_r \in \mathcal{U} \tag{3.11c}$$

$$x_r \in \mathcal{X} \tag{3.11d}$$

In the case that a reference signal r results in an infeasible optimization problem (3.11), then one can instead solve

$$\underset{x_r, u_r}{\text{minimize}} \; (y_k - r)^T W (y_k - r)$$

$$\text{s.t.}$$

$$\begin{bmatrix} A - I & B \end{bmatrix} \begin{bmatrix} x_r \\ u_r \end{bmatrix} = 0$$

$$u_r \in \mathcal{U}$$

$$x_r \in \mathcal{X}$$

resulting in the steady state that keeps the output as close as possible to r [Muske and Rawlings, 1993]. An alternative to this approach is to use a pseudo reference which we will discuss further later in this section.

Now a pragmatic approach to implement the reference tracking case is to simply shift the origin of the problem to the new setpoint and apply the standard MPC scheme on the translated system, i.e., in the original coordinate system, penalize deviations from the steady state setpoints.

$$\underset{u_{k+i},x_{k+i}}{\text{minimize}} \sum_{i=0}^{N-1} \ell(x_{k+i}-x_r, u_{k+i}-u_r) + \Psi(x_{k+N}-x_r) \tag{3.12a}$$

s.t.

$$x_{k+i+1} = Ax_{k+i} + Bu_{k+i} \tag{3.12b}$$

$$x_{k+i} \in \mathcal{X} \tag{3.12c}$$

$$u_{k+i} \in \mathcal{U} \tag{3.12d}$$

$$x_{k+N} \in \mathcal{T}(x_r) \tag{3.12e}$$

This formulation is the standard procedure of solving tracking problems in the MPC framework, see, e.g., Muske and Rawlings [1993], Rawlings et al. [1994], Lee and Cooley [1997], Meadows and Badgwell [1998], Rao and Rawlings [1999], Mayne et al. [2000], Rawlings [2000]. In this setup the terminal state constraint set now depends on the setpoint in the way that it shall be an invariant set for the translated system. This will in general lead to a terminal constraint set that has a different size and shape for every steady state setpoint. This is a complicating fact since it could require a recalculation of the terminal constraint set on-line. We will elaborate more on this topic in Chapter 4.

A further extension to the tracking concept is to use a so called *pseudo setpoint* or *pseudo reference* [Rossiter, 2006]. Instead of using the true reference r in (3.11) one introduces a new optimization variable $\bar{r}$ which gives a corresponding steady state and control, $\bar{x}$ and $\bar{u}$, in the optimization problem (3.12), and then penalize the deviation between the desired reference r and the pseudo reference $\bar{r}$ using a positive definite function $\phi(\bar{r}-r)$ in the objective function. By using this pseudo reference, the feasible region of the problem can be increased. The dual mode MPC problem when using a pseudo reference then has the form

$$\begin{aligned}\underset{u_{k+i},x_{k+i},\bar{x}_k,\bar{u}_k,\bar{r}_k}{\text{minimize}} \quad & \sum_{i=0}^{N-1} \ell(x_{k+i}-\bar{x}_k, u_{k+i}-\bar{u}_k) \\ & + \Psi(x_{k+N}-\bar{x}_k) + \phi(\bar{r}_k - r)\end{aligned} \tag{3.13a}$$

s.t.

$$x_{k+i+1} = Ax_{k+i} + Bu_{k+i} \tag{3.13b}$$

$$x_{k+i} \in \mathcal{X} \tag{3.13c}$$

$$u_{k+i} \in \mathcal{U} \tag{3.13d}$$

$$x_{k+N} \in \mathcal{T}(\bar{x}_k) \tag{3.13e}$$

$$\begin{bmatrix} A-I & B \\ C & D \end{bmatrix}\begin{bmatrix} \bar{x}_k \\ \bar{u}_k \end{bmatrix} = \begin{bmatrix} 0 \\ \bar{r}_k \end{bmatrix} \tag{3.13f}$$

Note that the only constraint on $\bar{r}$ is that it fulfills (3.13f), but since also $\bar{x}$ and $\bar{u}$ are variables in the optimization problem they can be chosen and the only constraint on how they must be chosen is (3.13e). The limiting constraint is that

$\bar{x}$ must be chosen such that the final state at the end of the prediction horizon is in the terminal set, $x_{k+N} \in \mathcal{T}(\bar{x})$. This means that an arbitrary reference, r, can not lead to infeasibility since we can always chose $\bar{r}$ such that $x_{k+N} \in \mathcal{T}(\bar{x})$ and the optimization problem remains feasible, but it might be at a high cost. In this way the pseudo reference acts as a prefiltering of the true reference such that feasibility can be guaranteed when changing references.

Both the formulation (3.12) and (3.13) requires a perfect model of the system otherwise there will be a nonzero steady state offset. Additionally, if there are disturbances acting on the system there will also be an offset between the desired output and the achieved output. Therefore one has to introduce integral action in the MPC controller for any practical purposes.

3.2.3 Integral control

A standard way of introducing integral control is to augment the system model with new states, ϵ, which is the integral of the tracking errors

$$\epsilon_k = \epsilon_{k-1} + r_k - y_k$$

and then penalize the integral state in the cost function. However this may not be a suitable approach in MPC since, firstly, the computational cost increases as the cube of the state dimension and secondly due to the presence of constraints there is a need for an anti-windup structure in order to avoid performance degradation [Muske and Badgwell, 2002].

Instead the standard procedure in MPC is to augment the system model with a constant disturbance, d_k,

$$x_{k+1} = Ax_k + Bu_k + Ed_k \tag{3.14a}$$
$$d_{k+1} = d_k \tag{3.14b}$$
$$y_k = Cx_k \tag{3.14c}$$

and use a disturbance observer to estimate this disturbance. Then the desired steady state is compensated for the disturbance.

$$\begin{bmatrix} A - I & B \\ C & 0 \end{bmatrix} \begin{bmatrix} x_r \\ u_r \end{bmatrix} = \begin{bmatrix} -E\hat{d}_k \\ r \end{bmatrix} \tag{3.15}$$

Many papers have been published related to integral control in MPC and almost all are variants of this setup. One of the pioneering contributions by Muske and Rawlings [1993] consider both input and output constant disturbances and use the Kalman filter to estimate the disturbances. In Meadows and Badgwell [1998] this concept is analyzed in the context of nonlinear MPC for constant output disturbances. Pannocchia and Rawlings [2003] and Pannocchia and Kerrigan [2003] extend the theory to a more general disturbance model and non square systems and Maeder and Morari [2010] extends the theory further to handle non constant disturbances such as sinusoids and ramps.

We will in this thesis only consider the standard setup with a constant disturbance model and following the derivation in Åkesson and Hagander [2003] we show that the use of a disturbance observer in MPC will, in fact, result in integral action.

For simplicity and clarity of the derivation we consider an integral form of MPC formulation (3.12) with a fixed reference, i.e., the terminal constraint set (3.12e) is a fixed set and can be described as a polytope. Instead of the state prediction used in (3.12) we use the state prediction equation $x_{k+i+1} = Ax_{k+i} + Bu_{k+i} + E\hat{d}_k$ with $x_k = \hat{x}_k$ where $\hat{x}_k$ and $\hat{d}_k$ are the observed state and disturbance. The disturbance observer for the system (3.14) is given by

$$\begin{bmatrix} \hat{x}_{k+1} \\ \hat{d}_{k+1} \end{bmatrix} = \begin{bmatrix} A & E \\ 0 & I \end{bmatrix} \begin{bmatrix} \hat{x}_k \\ \hat{d}_k \end{bmatrix} + \begin{bmatrix} B \\ 0 \end{bmatrix} u_k + L(y - C\hat{x}_k) \tag{3.16}$$

where L is the observer feedback gain.

We also assume that the steady state and control are given by

$$\begin{bmatrix} A-I & B \\ C & 0 \end{bmatrix} \begin{bmatrix} x_r \\ u_r \end{bmatrix} = \begin{bmatrix} -E\hat{d}_k \\ r \end{bmatrix}$$

$$\begin{bmatrix} x_r \\ u_r \end{bmatrix} = \begin{bmatrix} A-I & B \\ C & 0 \end{bmatrix}^{-1} \begin{bmatrix} -E\hat{d}_k \\ r \end{bmatrix} = \begin{bmatrix} \Pi_{xd} & \Pi_{xr} \\ \Pi_{ud} & \Pi_{ur} \end{bmatrix} \begin{bmatrix} \hat{d}_k \\ r \end{bmatrix} \tag{3.17}$$

By defining $\mathbb{X} = \begin{bmatrix} x_{k+1}^T & x_{k+2}^T & \dots & x_{k+N}^T \end{bmatrix}^T$ and $\mathbb{U} = \begin{bmatrix} u_k^T & u_{k+1}^T & \dots & u_{k+N-1}^T \end{bmatrix}^T$ we can write (3.12) as

$$\underset{\mathbb{U}}{\text{minimize}} \; (\mathbb{X} - \mathbb{1}x_r)^T \mathcal{Q} (\mathbb{X} - \mathbb{1}x_r) + (\mathbb{U} - \mathbb{1}u_r)^T \mathcal{R} (\mathbb{U} - \mathbb{1}u_r) \tag{3.18}$$

$$\text{s.t.} \tag{3.19}$$

$$\mathbb{X} = \mathcal{A}\hat{x}_k + \mathcal{B}\mathbb{U} + \mathcal{E}\hat{d}_k \tag{3.20}$$

$$\mathcal{F}_x \mathbb{X} \leq \Upsilon_x \tag{3.21}$$

$$\mathcal{F}_u \mathbb{U} \leq \Upsilon_u \tag{3.22}$$

where $\mathbb{1}$ is a vector of all ones and with the matrices

$$\mathcal{Q} = \begin{bmatrix} Q & & 0 & 0 \\ & Q & & 0 \\ 0 & & \ddots & \\ 0 & 0 & & P \end{bmatrix}, \quad \mathcal{R} = \begin{bmatrix} R & & 0 & 0 \\ & R & & 0 \\ 0 & & \ddots & \\ 0 & 0 & & R \end{bmatrix}$$

and

$$\mathcal{A} = \begin{bmatrix} A \\ A^2 \\ \vdots \\ A^N \end{bmatrix}, \quad \mathcal{B} = \begin{bmatrix} B & 0 & \dots & 0 \\ AB & B & \dots & 0 \\ \vdots & & \ddots & \\ A^{N-1}B & A^{N-2}B & \dots & B \end{bmatrix}, \quad \mathcal{E} = \begin{bmatrix} E \\ AE + E \\ \vdots \\ \sum_{j=0}^{N-1} A^j E \end{bmatrix}$$

and the state and control constraint matrices defined as

$$\Upsilon_x = [b_{\mathcal{X}}^T \; b_{\mathcal{X}}^T \; \dots \; b_{\mathcal{T}}^T]^T, \quad \Upsilon_u = [b_{\mathcal{U}}^T \; b_{\mathcal{U}}^T \; \dots \; b_{\mathcal{U}}^T]^T$$

$$\mathcal{F}_x = \begin{bmatrix} F_{\mathcal{X}} & & 0 & 0 \\ & F_{\mathcal{X}} & & 0 \\ 0 & & \ddots & \\ 0 & 0 & & F_{\mathcal{T}} \end{bmatrix}, \quad \mathcal{F}_u = \begin{bmatrix} F_{\mathcal{U}} & & 0 & 0 \\ & F_{\mathcal{U}} & & 0 \\ 0 & & \ddots & \\ 0 & 0 & & F_{\mathcal{U}} \end{bmatrix}$$

We can eliminate the state variables from the above formulation with the use of the state dynamic equation. This results in the following optimization problem

$$\begin{aligned} \underset{\mathbb{U}}{\text{minimize}}\ & \mathbb{U}^T \left(\mathcal{B}^T \mathcal{Q} \mathcal{B} + \mathcal{R}\right) \mathbb{U} + 2\mathbb{U}^T \mathcal{B}^T \mathcal{Q} \mathcal{A} \hat{x}_k + 2\mathbb{U}^T \mathcal{B}^T \mathcal{Q} \mathcal{E} \hat{d}_k \\ & - 2\mathbb{U}^T \mathcal{B}^T \mathcal{Q} \mathbb{1} x_r - 2\mathbb{U}^T \mathcal{R}^T \mathbb{1} u_r + \Phi(\hat{x}_k, \hat{d}_k, x_r, u_r) \\ \text{s.t.}\ & \\ & \begin{bmatrix} \mathcal{F}_u \\ \mathcal{F}_x \mathcal{B} \end{bmatrix} \mathbb{U} \leq \begin{bmatrix} \Upsilon_u \\ \Upsilon_x - \mathcal{F}_x \mathcal{A} \hat{x}_k \end{bmatrix} \end{aligned}$$

where we have collected all the terms of the objective function that are independent of u_{k+i} in $\Phi(\,\cdot\,)$.

From Section 2.1.1 we know that at the optimum it must hold that the gradient of the Lagrangian is equal to zero, hence form the Lagrangian and differentiate it w.r.t. $\mathbb{U}$.

$$\begin{aligned} \frac{d\mathcal{L}}{d\mathbb{U}} &= 2\left(\mathcal{B}^T \mathcal{Q} \mathcal{B} + \mathcal{R}\right) \mathbb{U}^* + 2\mathcal{B}^T \mathcal{Q} \mathcal{A} \hat{x}_k + 2\mathcal{B}^T \mathcal{Q} \mathcal{E} \hat{d}_k \\ & \quad - 2\mathcal{B}^T \mathcal{Q} \mathbb{1} x_r - 2\mathcal{R}^T \mathbb{1} u_r + \Lambda^{*T} \begin{bmatrix} \mathcal{F}_u \\ \mathcal{F}_x \mathcal{B} \end{bmatrix} = 0 \end{aligned}$$

From this we can easily solve for $\mathbb{U}$

$$\mathbb{U}^* = -\left(\mathcal{B}^T \mathcal{Q} \mathcal{B} + \mathcal{R}\right)^{-1} \left(\mathcal{B}^T \mathcal{Q} \mathcal{A} \hat{x}_k + \mathcal{B}^T \mathcal{Q} \mathcal{E} \hat{d}_k - \mathcal{B}^T \mathcal{Q} \mathbb{1} x_r - \mathcal{R}^T \mathbb{1} u_r + \frac{1}{2} \Lambda^{*T} \begin{bmatrix} \mathcal{F}_u \\ \mathcal{F}_x \mathcal{B} \end{bmatrix}\right)$$

and the optimal control, u_k^*, is defined by the first row of the equation as

$$u_k^* = -K_x \hat{x}_k - K_d \hat{d}_k + K_{x_r} x_r + K_{u_r} u_r - K_\lambda \lambda^* \tag{3.23}$$

for an appropriate definition of the matrices, $K_{(\,\cdot\,)}$. Using the relation (3.17) we can write u_k^* as

$$\begin{aligned} u_k^* &= -K_x \hat{x}_k - \underbrace{\left(K_d - K_{x_r} \Pi_{xd} - K_{u_r} \Pi_{ud}\right)}_{\tilde{K}_d} \hat{d}_k + \underbrace{\left(K_{x_r} \Pi_{xr} + K_{u_r} \Pi_{ur}\right)}_{\tilde{K}_r} r - K_\lambda \lambda^* \\ &= -K_x \hat{x}_k - \tilde{K}_d \hat{d}_k + \tilde{K}_r r - K_\lambda \lambda^* \end{aligned} \tag{3.24}$$

Inserting (3.24) into the observer equations (3.16) we obtain

$$\begin{aligned} \begin{bmatrix} \hat{x}_{k+1} \\ \hat{d}_{k+1} \end{bmatrix} &= \begin{bmatrix} A - BK_x - L_x C & E - B\tilde{K}_d \\ -L_d C & I \end{bmatrix} \begin{bmatrix} \hat{x}_k \\ \hat{d}_k \end{bmatrix} + \begin{bmatrix} B\tilde{K}_r \\ 0 \end{bmatrix} r + \begin{bmatrix} L_x \\ L_d \end{bmatrix} y_k + \begin{bmatrix} BK_\lambda \\ 0 \end{bmatrix} \lambda^* \\ &\triangleq \begin{bmatrix} \tilde{A}_{11} & \tilde{A}_{12} \\ -\tilde{A}_{21} & I \end{bmatrix} \begin{bmatrix} \hat{x}_k \\ \hat{d}_k \end{bmatrix} + \begin{bmatrix} B\tilde{K}_r \\ 0 \end{bmatrix} r + \begin{bmatrix} L_x \\ L_d \end{bmatrix} y_k + \begin{bmatrix} BK_\lambda \\ 0 \end{bmatrix} \lambda^* \end{aligned}$$

Rewriting the observer equations into the z-transform domain and into transfer matrix form yield

$$\begin{bmatrix}\hat{x}(z)\\ \hat{d}(z)\end{bmatrix} = \begin{bmatrix} z^{-1}I - \tilde{A}_{11} & -\tilde{A}_{12}\\ \tilde{A}_{21} & \frac{1-z}{z}I\end{bmatrix}^{-1}\left(\begin{bmatrix}B\tilde{K}_r\\ 0\end{bmatrix} r(z) + \begin{bmatrix}L_x\\ L_d\end{bmatrix} y(z) + \begin{bmatrix}BK_\lambda\\ 0\end{bmatrix}\lambda^*\right) \tag{3.25}$$

and the matrix inverse is

$$\begin{bmatrix} z^{-1}I - \tilde{A}_{11} & -\tilde{A}_{12}\\ \tilde{A}_{21} & \frac{1-z}{z}I\end{bmatrix}^{-1} = \begin{bmatrix}\Psi(z) & \frac{z}{1-z}\Psi(z)\tilde{A}_{12}\\ -\frac{z}{1-z}\tilde{A}_{21}\Psi(z) & \frac{z}{1-z}\left(I - \frac{z}{1-z}\tilde{A}_{21}\Psi(z)\tilde{A}_{12}\right)\end{bmatrix}$$

where, $\Psi(z) = z^{-1}I - \tilde{A}_{11} + \frac{z}{1-z}\tilde{A}_{12}\tilde{A}_{21}$, denotes the Schur complement to $\frac{1-z}{z}I$.

Inserting (3.25) into the z-transform of (3.24) we finally obtain

$$\begin{aligned} u(z) &= -K_x\hat{x}(z) - \tilde{K}_d\hat{d}(z) + \tilde{K}_r r(z) - K_\lambda\lambda^*\\ &= -K_x\Psi(z)L_x y(z) - \left(K_x\Psi(z)B - I\right)\tilde{K}_r r(z)\\ &+ \frac{z}{1-z}\Phi(z)(\Gamma(z)r(z) - y(z))\\ &- \left(K_x\Psi(z)B - \frac{z}{1-z}\tilde{K}_d\tilde{A}_{21}\Psi(z)B - I\right)K_\lambda\lambda^* \end{aligned} \tag{3.26}$$

with

$$\Phi(z) = K_x\Psi(z)\tilde{A}_{12}L_d - \tilde{K}_d\tilde{A}_{21}\Psi(z)L_x + \tilde{K}_d - \frac{z}{1-z}\tilde{K}_d\tilde{A}_{21}\Psi(z)\tilde{A}_{12}L_d$$

$$\Gamma(z) = \Phi^{-1}(z)\tilde{K}_d\tilde{A}_{21}\Psi(z)B\tilde{K}_r$$

Equation (3.26) shows that the MPC setup with a disturbance observer for a constant disturbance will result in an output feedback term, $-K_x(z)\Psi(z)L_x y(z)$, a reference feedforward term, $-\left(K_x(z)\Psi(z)\tilde{B}(z) - \tilde{K}_r(z)\right)r(z)$, a term depending on the optimal dual variables, $-\left(K_x\Psi(z)B - \frac{z}{1-z}\tilde{K}_d\tilde{A}_{21}\Psi(z)B - I\right)K_\lambda\lambda^*$, and an integral term of the output error, $\frac{z}{1-z}\Phi(z)(\Gamma(z)r(z) - y(z))$. It is worth noting that if no constraints are active then the complementary slackness condition (2.9) implies that the optimal dual variables, λ^*, are zero and hence that term has no affect on the control signal (as one would expect).

In Figure 3.2 we apply this integral control algorithm to the pitch dynamics of the same aircraft as in Section 4.3.1, but here subject to a constant state disturbance. The figure clearly shows the benefits of adding integral control since without the integral action there is a large offset from the reference signal which is completely eliminated with the use of integral control.

Note that we have not discussed anything about the stability properties of this output feedback MPC algorithm. As a matter of fact, the well known *principle of separation* does not hold and a separate state estimator and feedback design does not necessary stabilize nonlinear and constrained systems. We will not go into detail on the stabilizing properties of output feedback MPC in this thesis, instead we refer to Rawlings and Mayne [2009] for a more in depth treatment.

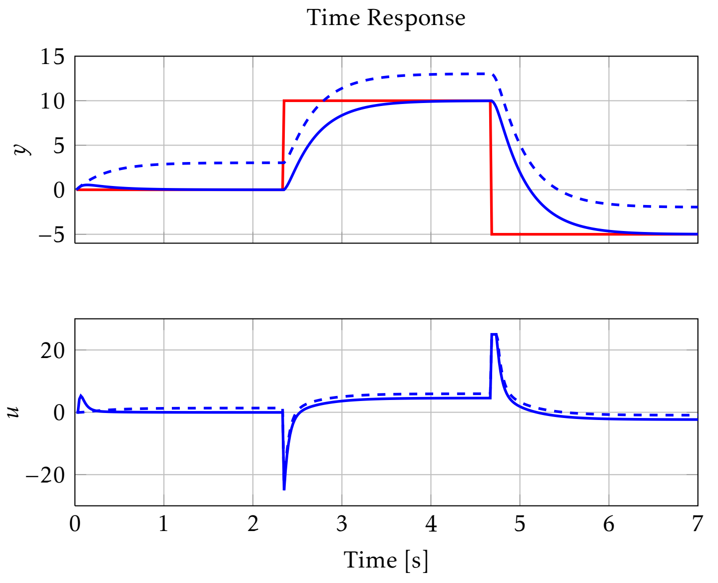

Figure 3.2: *The step respons of a system subject to a constant state disturbance when controlled with a MPC controller without integral action and with integral action. The dashed lines are the output and control signal of an MPC controller without integral action and the solid lines are the output and control signal when integral action is added.*

3.2.4 Slack variables

Due to the iterative nature of the MPC method it can happen, due to, e.g., model errors or disturbances, that at some point in time the optimization problem becomes infeasible. There exist several ways to handle this, one ad hoc way is to simply use the next control signal, u_{k+1}, from the optimal sequence in the previous time step; but there is no guarantee that the problem becomes feasible again at a later point in time.

A better and more systematic way to handle infeasibility issues is to add an extra optimization variable, a so called *slack variable*, ε, to the problem. The slack variable is a non-negative vector that is added to the right hand side of the state constraints

$$F_{\mathcal{X}} x \leq b_{\mathcal{X}} + \varepsilon$$

The effect of adding the slack variable to the right hand side of the inequality constraints is that if an element of the slack variable is larger than zero then the corresponding constraint is relaxed, i.e., feasibility can be regained.

Usually, the slack variables are only added to the state constraints and not to the input constraint [Maciejowski, 2002], since the input constraints usually are real hard constraints with physical limits, hence it makes no sense to relax those constraints. Note however that in theory one could also add slack to the input constraints if it make any sense, e.g., if the input signal limits have some safety margin to the real physical limit which one do not want to break unless it is extremely important.

In addition to adding the slack to the constraints it is also added in the cost function with some positive definite function, $\varphi(\varepsilon)$, that penalizes the constraint violations. In de Oliveira and Biegler [1994] it is suggested to use a quadratic cost function of the form

$$\varphi(\varepsilon) = \rho \,\|\varepsilon\|^2$$

The drawback with this type of penalty function is that if any constraints are active in the optimum, then a quadratic penalty on the slack variable will always cause the constraints to be violated to some small extent. This comes from the fact that the increased cost from the penalty function is of order $\mathcal{O}(\varepsilon^2)$ and smaller than the reduced cost from the other parts of the cost function, which is of order $\mathcal{O}(\varepsilon)$, for small values of ε [Maciejowski, 2002]. If instead a linear penalty function is used then the increase in cost is also of order $\mathcal{O}(\varepsilon)$ and if ρ is large enough, then the constraints will only be violated if there does not exist a feasible solution to the problem with slack variables equal to zero [Maciejowski, 2002].

This is known as an *exact penalty* but the question remains how to chose the gain ρ. Unfortunately, in order to have exact penalty the choice of gain, ρ, depends on the optimal dual variables (which are not known until the problem is solved). Fletcher [1987] shows that the gain must be larger than the dual norm of the Lagrange multipliers (the dual variables).

$$\rho > \|\lambda^*\|_D$$

In order to overcome this problem one has to estimate a bound on the dual variables and in Kerrigan and Maciejowski [2000] the authors derive a fixed lower bound on the gain.

Finally the resulting MPC problem with integral action, pseudo setpoint and slack variables can be formulated as

$$\underset{u_{k+i},x_{k+i},\bar{x}_k,\bar{u}_k,\bar{r}_k,\varepsilon}{\text{minimize}} \quad \sum_{i=0}^{N-1} \ell(x_{k+i} - \bar{x}_k, u_{k+i} - \bar{u}_k) + \Psi(x_{k+N} - \bar{x}_k) + \phi(\bar{r}_k - r) + \rho \,\|\varepsilon\|_\infty \tag{3.27a}$$

s.t.

$$x_{k+i+1} = Ax_{k+i} + Bu_{k+i} + E\hat{d}_k \tag{3.27b}$$

$$F_{\mathcal{X}} x_{k+i} \le b_{\mathcal{X}} + \varepsilon \tag{3.27c}$$

$$F_{\mathcal{U}} u_{k+i} \le b_{\mathcal{U}} \tag{3.27d}$$

$$F_{\mathcal{T}} x_{k+N} \le b_{\mathcal{T}} \tag{3.27e}$$

$$\begin{bmatrix} A - I & B \\ C & D \end{bmatrix} \begin{bmatrix} \bar{x}_k \\ \bar{u}_k \end{bmatrix} = \begin{bmatrix} -E\hat{d}_k \\ \bar{r}_k \end{bmatrix} \tag{3.27f}$$

where x_k, r and $\hat{d}_k$ are the input variables and $\hat{d}_k$ is estimated with a disturbance observer like (3.16). The applied control law is $\kappa(x_k) = u_k^*$.

3.2.5 The explicit solution

For many years it has been the general conception that an explicit solution to the constrained optimal control problem (3.2) is extremely difficult to calculate [Mayne et al., 2000]. However in the last decade this notion has been overthrown by what is called *Explicit Model Predictive Control*.

The development of explicit MPC has been motivated mainly by the need for MPC algorithms to control systems with very fast dynamics and a need for algorithms to be implemented on low cost hardware, where an online optimization is not possible. However yet another aspect is that of verifiability of the algorithm for safety critical applications; since in the recent years very fast online algorithms have been developed, verifiability might be the most important aspect today for using explicit MPC in aeronautical applications.

Explicit MPC builds on the concept of multi-parametric programming, which has been a research topic within the optimization community since the 1960's [Gal, 1980], but it were not until the papers of Bemporad et al. [2002a] and Bemporad et al. [2002b] that these techniques became widely used within the MPC community.

A multi-parametric program can in its general form be written as

$$z^*(\theta) = \arg\min f(z,\theta)$$

$$\text{s.t.} \quad g(z,\theta) \leq 0$$

where z is the optimization variable and θ is a vector of unknown parameters. Both the objective function and the constraint function can depend on the parameter vector and of course also the solution $z^*(\theta)$ is dependent on the parameter values.

We will in this section derive the explicit formulation for the quadratic program (3.5) where the constraint sets $\mathcal{X}$, $\mathcal{U}$ and $\mathcal{T}$ are polytopic sets.

In the same manner as we did in Section 3.2.3 we can rewrite the basic MPC formulation (3.5) by defining the concatenated vectors

$$\mathbb{X} = \begin{bmatrix} x_{k+1}^T & x_{k+2}^T & \dots & x_{k+N}^T \end{bmatrix}^T$$

and

$$\mathbb{U} = \begin{bmatrix} u_k^T & u_{k+1}^T & \dots & u_{k+N-1}^T \end{bmatrix}^T$$

which results in

$$\underset{\mathbb{U}}{\text{minimize}} \; \mathbb{X}^T \mathcal{Q} \mathbb{X} + \mathbb{U}^T \mathcal{R} \mathbb{U} \tag{3.28a}$$

$$\text{s.t.}$$

$$\mathbb{X} = \mathcal{A}x_k + \mathcal{B}\mathbb{U} \tag{3.28b}$$
$$\mathcal{F}_x\mathbb{X} \leq \Upsilon_x \tag{3.28c}$$
$$\mathcal{F}_u\mathbb{U} \leq \Upsilon_u \tag{3.28d}$$

From this we can again use the state dynamics equation (3.28b) to eliminate the state variable from (3.28) and the resulting optimization problem is a multi-parametric problem

$$\underset{\mathbb{U}}{\text{minimize}}\ \mathbb{U}^T\left(\mathcal{B}^T\mathcal{Q}\mathcal{B} + \mathcal{R}\right)\mathbb{U} + 2\mathbb{U}^T\left(\mathcal{B}^T\mathcal{Q}\mathcal{A}\right)x_k + x_k^T\left(\mathcal{A}^T\mathcal{Q}\mathcal{A}\right)x_k \tag{3.29a}$$
$$\text{s.t.} \begin{bmatrix}\mathcal{F}_u \\ \mathcal{F}_x\mathcal{B}\end{bmatrix}\mathbb{U} \leq \begin{bmatrix}\Upsilon_u \\ \Upsilon_x - \mathcal{F}_x\mathcal{A}x_k\end{bmatrix} \tag{3.29b}$$

where $\mathbb{U}$ is the optimization variable and x_k the unknown parameter.

If we introduce the variable

$$z = \mathbb{U} + \left(\mathcal{B}^T\mathcal{Q}\mathcal{B} + \mathcal{R}\right)^{-1}\mathcal{B}^T\mathcal{Q}\mathcal{A}x_k$$

we can rewrite (3.29) to

$$\underset{z}{\text{minimize}}\ z^THz \tag{3.30a}$$
$$\text{s.t.} \quad Mz \leq N + Sx_k \tag{3.30b}$$

with

$$H = \mathcal{B}^T\mathcal{Q}\mathcal{B} + \mathcal{R}, \quad M = \begin{bmatrix}\mathcal{F}_u \\ \mathcal{F}_x\mathcal{B}\end{bmatrix}$$

$$N = \begin{bmatrix}\Upsilon_u \\ \Upsilon_x\end{bmatrix}, \quad S = \begin{bmatrix}\mathcal{F}_u\left(\mathcal{B}^T\mathcal{Q}\mathcal{B} + \mathcal{R}\right)^{-1}\mathcal{B}^T\mathcal{Q}\mathcal{A} \\ -\mathcal{F}_x\mathcal{A} + \mathcal{F}_x\mathcal{B}\left(\mathcal{B}^T\mathcal{Q}\mathcal{B} + \mathcal{R}\right)^{-1}\mathcal{B}^T\mathcal{Q}\mathcal{A}\end{bmatrix}$$

where now only the right hand side of the constraints depend on the parameter.

Given a fixed value on the parameter x_k the solution to the multi-parametric QP (3.30) is given by the KKT conditions

$$2Hz + M^T\lambda = 0 \tag{3.31a}$$
$$Mz - (N + Sx_k) \leq 0 \tag{3.31b}$$
$$\lambda \geq 0 \tag{3.31c}$$
$$\lambda_i\Big(M_iz - (N + Sx_k)_i\Big) = 0 \tag{3.31d}$$

where the subscript i denotes the i:th row of the matrices. From (3.31a) we can solve for z

$$z^* = -H^{-1}M^T\lambda$$

which if we insert it into the complementary slackness condition (3.31d) gives

$$\lambda\Big(MH^{-1}M^T\lambda + (N + Sx_k)\Big) = 0$$

From Section 2.1.2 we know that for all active constraints it must hold that

$$M_{\mathcal{A}} H^{-1} M_{\mathcal{A}}^T \lambda_{\mathcal{A}} + N_{\mathcal{A}} + S_{\mathcal{A}} x_k = 0$$

where the subscript $\mathcal{A}$ denotes the part of the matrices for which the corresponding constraint is active. From this equation we can determine the dual variable and substitute it into the solution for, $z^*(x_k)$, yielding

$$z^* = H^{-1} M^T \left(M_{\mathcal{A}} H^{-1} M_{\mathcal{A}}^T \right)^{-1} (N_{\mathcal{A}} + S_{\mathcal{A}} x_k) \tag{3.32}$$

From (3.32) we observe that the optimal solution $z^*(x_k)$ to the KKT conditions (3.31) is an affine function of the parameter x_k

$$z^*(x_k) = F_i x_k + g_i$$

where F_i and g_i depends on the set of active constraints and hence the solution $z^*(x_k)$ is a different affine function depending on the set of active constraints. The set of the active constraints, $\mathcal{X}_i$, can easily be determined if we note that the optimal solution $z^*(x_k)$ must fulfill the primal feasibility constraint (3.31b), i.e.,

$$M H^{-1} M^T \left(M_{\mathcal{A}} H^{-1} M_{\mathcal{A}}^T \right)^{-1} (N_{\mathcal{A}} + S_{\mathcal{A}} x_k) \leq N + S x_k \tag{3.33}$$

The inequality constraint (3.33) specifies the polytopic subset, $\mathcal{X}_i$, of the state space where the optimal solution, $z^*(x_k)$, is valid.

It can be shown that the optimal solution $z^*(x_k)$ is, in fact, a continuous piecewise affine function (in the sense of definition 3.3) of the paramater x_k [Bemporad et al., 2002b]. Each affine function valid in a different polytopic subset, $\mathcal{X}_i$, of the state space where the active constraints does not change.

3.3 Definition. A function $z^*(x)$ is said to be piecewise affine if it is possible to partition the polyhedral state constraint set $\mathcal{X}$ into convex polyhedral regions $\mathcal{X}_i$ such that $z^*(x) = F_i x + g_i \;\; \forall \; x \in \mathcal{X}_i$

The calculations of the explicit controller is an iterative procedure where the entire feasible set is partitioned into convex polyhedral regions, each with an affine feedback solution. Here we will only outline the main structure of one such algorithm and refer to Baotic [2005] for the details. Once an initial x_k has been chosen and the affine solution and the polytopic subset, $\mathcal{X}_i$, has been calculated, then the remainder of the state space can be partitioned into convex regions, $\mathcal{P}_j$. In each convex region a new x_k is chosen and a new solution, $z^*(x_k)$ and the polytope of active constraints, $\mathcal{X}_i$ are calculated. The algorithm gradually explores the different regions $\mathcal{P}_j$, finding new partitions, $\mathcal{X}_i$, and dividing the remaining region into new smaller regions. This procedure continues until all regions have been divided into partions and no unexplored regions remain. Then the entire feasible set will be divided into convex polyhedral partitions, see Figure 3.3.

We summarize the above discussion in Algorithm 1.

Note that with the procedure for exploring the state space by dividing it into regions $\mathcal{P}_j$ which has been described in both Dua and Pistikopoulos [2001] and

Algorithm 1 Offline calculations of the explicit MPC feedback solution

1: Given an initial set of constraints (in the first iteration the state constraint set $\mathcal{X}$), select an initial state x_k belonging to the set, arguably the Chebychev center (2.10) of the set [Bemporad et al., 2002b].
2: Solve the KKT equations 3.31.
3: Determine the set of active constraints.
4: Calculate F_i and g_i from (3.32).
5: Determine the set $\mathcal{X}_i$ from (3.33).
6: Partition the remainder of the state constraint set into non overlapping convex regions $\mathcal{P}_j$.
7: Repeat from 1, now with the set $\mathcal{P}_j$ as initial set of constraints.

Bemporad et al. [2002b] it can happen, due to the nature of the algorithm, that several adjacent partitions have the same feedback solution. If the partitions constitute a convex set, then they can be merged into one partition, reducing the complexity of the controller, see Figure 3.3.

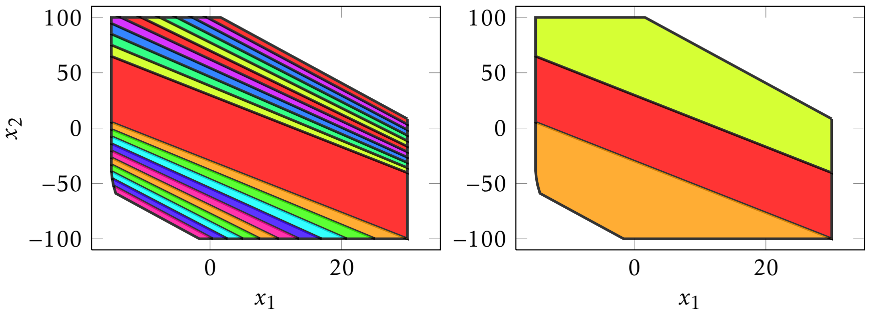

Figure 3.3: *In the left axis the total number of partitions, $\mathcal{X}_i$, generated by the algorithm and in the right axis the number of partitions after reduction.*

The online complexity of the explicit solution is mainly affected by the task of finding the partition, $\mathcal{X}_i$, in which the current state, x_k, is located (the so called *point location problem*) and hence dependent on the number of partitions, $N_{\mathcal{X}}$, that are required to describe the solution. The runtime complexity is in worst case $\mathcal{O}(N_{\mathcal{X}})$ if a straightforward simple search routine is implemented but with advanced search algorithms the runtime can be reduced to $\mathcal{O}(\log_2 N_{\mathcal{X}})$ [Kvasnica, 2009, Jones et al., 2006].

Since the number of state space partitions so greatly affect the applicability of explicit MPC, lots of research have been performed on the task of defining a simple state space partition structure in order to reduce the complexity in the point location problem, see, e.g., Rubagotti et al. [2012], Raimondo et al. [2012], Genuit et al. [2011].

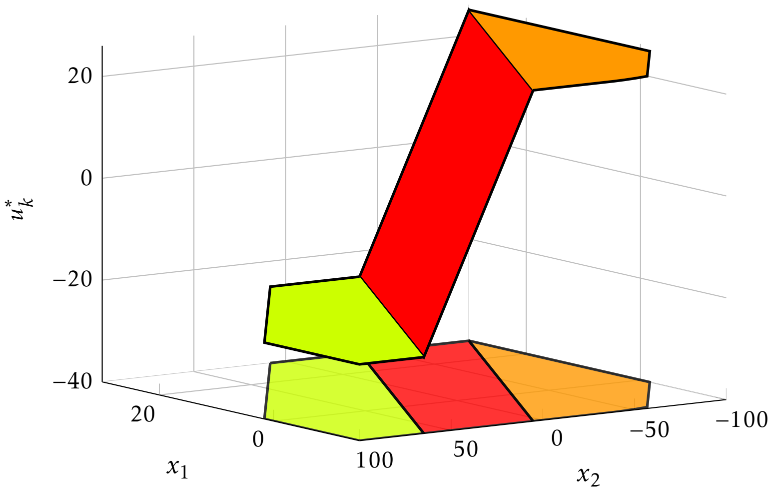

Figure 3.4: *The optimal explicit MPC feedback solution plotted over the polytopic regions where each affine solution is valid.*

In Figure 3.3 and 3.4 we have used the pitch dynamics of the same aircraft as in Section 4.3.1 to calculate a simple explicit MPC controller for stabilizing the unstable dynamics around the origin. In Figure 3.4 we can see that the optimal feedback solution is a piecewise affine function with three regions. One region where the solution is the discrete LQ solution and two regions where it is the saturated control input.

In Chapter 4 we develop a new MPC formulation for reference tracking purposes which has the property of greatly reducing the number of state space partitions needed to describe the explicit solution.

3.3 Nonlinear MPC

Even though all MPC formulations, because of the constraints, result in nonlinear controllers, the term nonlinear MPC usually refers to the case where the system dynamics is nonlinear

$$x_{k+1} = f(x_k, u_k)$$

and the MPC problem formulation (3.5) now looks like

$$\underset{u_{k+i}, x_{k+i}}{\text{minimize}} \sum_{i=0}^{N-1} \ell(x_{k+i}, u_{k+i}) + \Psi(x_{k+N}) \tag{3.34a}$$

s.t.

$$x_{k+i+1} = f(x_{k+i}, u_{k+i}) \tag{3.34b}$$

$$x_{k+i} \in \mathcal{X} \tag{3.34c}$$

$$u_{k+i} \in \mathcal{U} \tag{3.34d}$$

$$x_{k+N} \in \mathcal{T} \tag{3.34e}$$

Even though (3.34) looks very much like (3.5) and the analysis in Section 3.2.1, such as Theorem 3.2, can be extended to hold also for nonlinear systems there are some very important subtile differences.

In general it is extremely difficult to find the terminal state penalty, $\Psi(x_{k+N})$, and constraint set, $\mathcal{T}$, such that the problem corresponds to the infinite horizon optimal control problem. In the linear case this is relatively straightforward and a quadratic penalty term can be used with the unconstrained LQ solution, P, as weight.

Instead several different approaches on finding good approximate solutions to the infinite horizon problem has been proposed in the literature. Many of the methods, their stability properties, performance and implementation are analyzed in the easily accessible paper by Nicolao et al. [2000].

One popular approach proposed by, e.g., Michalska and Mayne [1993], Chen and Allgöwer [1998] and Magni et al. [2001], is to use a linear controller that is stabilizing the linearization of the system around the origin, to define terminal state constraint and cost. Another approach found in Primbs et al. [1999] and Jadbabaie et al. [2001] uses the theory of Control Lyapunov Functions to define an appropriate terminal cost function. Nicolao et al. [1996, 1998] propose to use a nonquadratic terminal state cost which is infinite outside an implicit terminal region, forcing the terminal state into a region where a linear state feedback stabilizes the system. In the paper by De Oliveira Kothare and Morari [2000] the authors introduce *contractive constraints* which forces the norm of the state at the prediction horizon to be smaller than the norm of the current state. Using this the authors can show, under feasibility assumptions, that the closed loop system is exponentially stable.

Other relevant papers analyzing stability of nonlinear MPC are Mayne and Michalska [1990], Michalska [1997], Henson [1998], Morari and Lee [1999], Mayne [2000] and Fontes [2001]. In Scokaert et al. [1999] two suboptimal MPC algorithms are analyzed which can guarantee stability even in the absence of an optimal solution. The main benefit of the proposed suboptimal schemes is the low computational complexity compared to other algorithms.

Regardless which of the above mentioned stabilizing algorithms one consider, the fact that the system dynamics is nonlinear makes the MPC formulation (3.34) a nonconvex program. In general, nonconvex programs are much harder to solve than the convex program that a linear dynamics and quadratic cost function results in [Boyd and Vandenberghe, 2004]. In the survey of Cannon [2004] several different techniques to solve the nonlinear MPC problem in a computational efficient way are reviewed. These methods include *Sequential Quadratic Programming*, (SQP), *Euler-Lagrange-* and *Hamilton-Jacobi-Bellman* approaches as well as *Cost and constraint approximation*.

The SQP method tries to find a solution for (3.34) by solving a sequence of approximations defined as quadratic optimization problems. The solution to each QP gives a search direction for the original problem (3.34). The main drawback with the SQP method is that the sequence of iterates can be extremely long and secondly that there are no guarantees that the sequence converges to the global minimum.

The Euler-Lagrange and Hamiltion-Jacobi-Bellman techniques circumvents the troublesome task of solving the nonconvex optimization problem by viewing the nonlinear MPC problem formulation in the light of optimal control. The Euler-Lagrange method numerically solves the two point boundary value problem that arise from Pontryagin's Maximum Principle and the Hamilton-Jacobi-Bellman approach tries to numerically solve the Hamilton-Jacobi-Bellman partial differential equation. These techniques either lack any stability guarantees or are extremely computational expensive.

Another approach to reduce the computational burden is to approximate the cost and constraint functions such that a convex program can be solved instead. This approach of course yields a suboptimal solution to the original problem (3.34) and for some approximation schemes not even recursive feasibility can be guaranteed.

One fairly popular approach to remove the nonlinearity of the system dynamics is to first design a linearizing inner feedback loop and then add a linear MPC controller for the linearized system. This alternative also has some major drawbacks and in Chapter 5 we investigate this approach further.

4

A Low Complexity Reference Tracking MPC Algorithm

Among the many different formulations of MPC with guaranteed stability, one that has attracted a lot of attention is the formulation with a terminal cost and terminal constraint set, i.e., the dual mode formulation. Despite its popularity only little has been published concerning stability properties for reference tracking applications. In this chapter we build on the dual mode formulation of MPC and our goal is to make minimal changes to this framework, for the linear polytopic case, in order to develop a flexible reference tracking algorithm with guaranteed stability and low complexity, which is intuitive and easily understood.

The main idea is to introduce a scaling variable that dynamically scales the terminal constraint set and therefore allows it to be centered around an arbitrary setpoint without violating the stability conditions. The main benefit of the algorithm is reduced complexity of the resulting QP compared to other state of art methods without loosing performance.

This chapter is an edited version of the journal paper:

> D. Simon, J. Löfberg, and T. Glad. Reference Tracking MPC using Dynamic Terminal Set Transformation. *IEEE Transactions on Automatic Control*, 2014. Provisionally accepted for publication.

which is an extension of the work previously published in the conference paper

> D. Simon, J. Löfberg, and T. Glad. Reference tracking MPC using terminal set scaling. In *51st IEEE Conference on Decision and Control (CDC)*, pages 4543–4548, Dec. 2012.

4.1 Introduction

As described in Section 3.2.2 it is often argued that with a simple change of coordinates the origin can represent any suitable setpoint for the controller. A pragmatic approach to implement the reference tracking case would therefore be to apply the standard MPC scheme (3.5) on the translated system, i.e., in the original coordinate system, penalize deviations from the steady state setpoint.

However for the dual mode formulation the possibility to guarantee recursive feasibility and stability when tracking a reference is dependent on the possibility to guarantee that the terminal set, that depends on the chosen setpoint, still is a valid positively invariant set. In order to guarantee stability it must hold that the translated terminal set still fulfills the conditions (1) and (2) in Theorem 3.2. What could easily happen if a simple change of coordinates is used is that $\mathcal{T}(x_r) \nsubseteq \mathcal{X}$, i.e., a translation of $\mathcal{T}$ moves parts of it outside $\mathcal{X}$, see Figure 4.1, thus invalidating any claim of positive invariance (and similarly w.r.t to control constraints for the nominal controller in $\mathcal{T}(x_r)$).

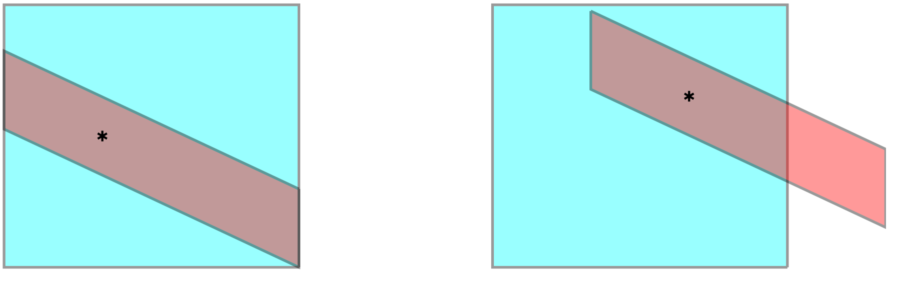

Figure 4.1: *Example of a terminal set $\mathcal{T}$ (dark shaded polytope) calculated with the setpoint at the origin (marked with a star) and the same terminal set shifted to an new setpoint x_r. When translating the terminal set, parts of it may leave the set $\mathcal{X}$ (the square polytope).*

Common assumptions when proving stability of linear MPC algorithms for tracking applications are to assume that the terminal constraint set is small enough and that the desired setpoints are located far into the interior of the feasible set such that the translated terminal constraint set still is positively invariant. This is a major drawback since many applications have optimal operating points on or close to the border of the feasible set.

To overcome the problem a straightforward way would be to recalculate the terminal constraint set online based on the current setpoint, see Figure 4.2, which is implicitly assumed in, e.g., Rao and Rawlings [1999], where the authors develop a method to handle active constraints at setpoints. To recalculate the terminal constraint set can however require complex calculations to be performed online and might thus not be suitable for systems where the setpoint changes often.

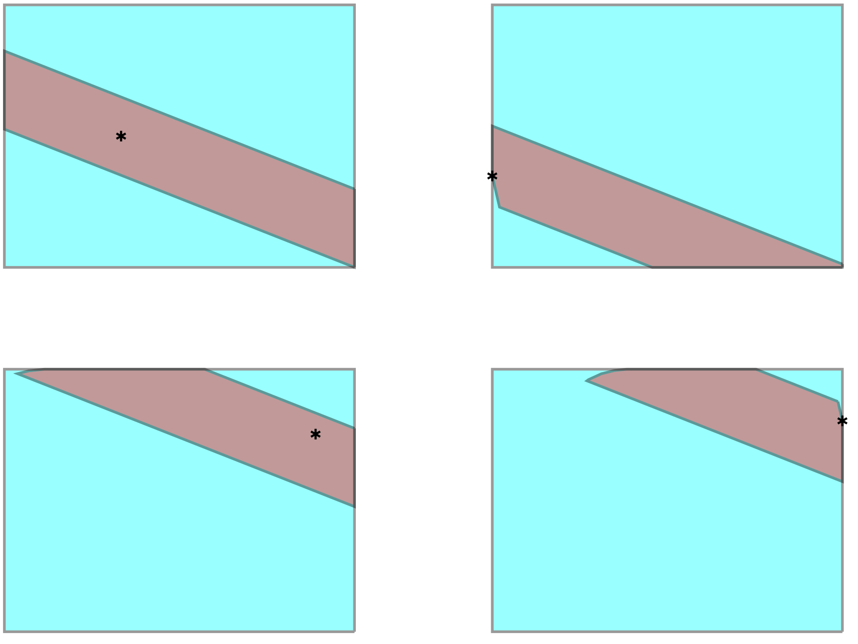

Figure 4.2: *Example of terminal state constraint set recalculation for different setpoints. Upper left axis is the terminal set calculated with the origin as the setpoint (marked with a star). The three other axis show the terminal constraint set calculated for three different setpoints other than the origin.*

Additionally when using the concept of pseudo setpoints as described in section 3.2.2, it is not possible to calculate a new terminal set online. Since the terminal set now depends on the value of pseudo setpoint which is a variable in the optimization problem and the value of the pseudo setpoint depends on the size and shape of the terminal set, this would result in a non tractable optimization problem.

A far better approach has been proposed in different forms by several authors Chisci and Zappa [2003], Bemporad et al. [1997], Limon et al. [2008], Ferramosca et al. [2009]. In Chisci and Zappa [2003] the authors augment the system with a new constant state which corresponds to the reference signal providing a terminal constraint set in the higher dimension (x, r) which contains the whole feasible equilibrium set for any given reference, i.e., the terminal constraint set is a fixed set which can be pre-calculated. Instead of using the reference as augmented state the authors of Limon et al. [2008], Ferramosca et al. [2009] introduce a new optimization variable, θ, which spans the null space of steady state equation (3.11b). The resulting controller guarantees feasibility and stability and has a larger domain of attraction than the standard MPC tracking controller.

The main drawback with these augmented state controllers is that they can be relatively complex even for small systems, making any explicit implementations impractical. This motivates an extension of the theory which is developed in detail in the remaining sections of this chapter. We develop a reference tracking algorithm with a modified terminal constraint set which can guarantee stability for arbitrary setpoints in the entire feasible set. The developed algorithm has the potential to be much simpler than existing methods.

4.2 The proposed controller

We will start with the MPC formulation (3.13) and in this section derive a high level representation of the proposed controller.

Instead of augmenting the system with a new state as in Chisci and Zappa [2003], Limon et al. [2008] the main concept here is to introduce an extra optimization variable, λ_k, which scales the terminal set, $\mathcal{T}$. The scaling allows us to move the terminal constraint set to an arbitrary setpoint, $\bar{x}_k$, since the terminal set can be scaled down to a single point, i.e., $\mathcal{T} = \bar{x}_k$, eliminating the need for any online recalculation of $\mathcal{T}$ as in Rao and Rawlings [1999] and the terminal constraint set can possibly be far simpler than the terminal set for an augmented system.

The proposed terminal state constraint is a scaled and translated version of the terminal constraint set (3.4e)

$$x_{k+N} \in \lambda_k \mathcal{T}(\bar{x}_k) \tag{4.1}$$

where λ_k is a non-negative scalar. The constraints on how λ_k can be chosen are such that the conditions of Theorem 3.2 hold, i.e., the terminal set must be positively invariant (which we will argue that it is in Lemma 4.5) and the state and control constraints must be satisfied in $\mathcal{T}$. To ensure this the following constraints must be added

$$\lambda_k \mathcal{T}(\bar{x}_k) \subseteq \mathcal{X} \tag{4.2a}$$

$$\kappa(x, \bar{x}_k, \bar{u}_k) \in \mathcal{U} \quad \forall\, x \in \lambda_k \mathcal{T}(\bar{x}_k) \tag{4.2b}$$

The first constraint states that the scaled and translated terminal set is a subset of the feasible state space, i.e., state constraints are satisfied in $\lambda_k \mathcal{T}(\bar{x}_k)$, and the second constraint states that the predefined stabilizing controller $\kappa(\,\cdot\,)$ fulfills the control constraints for all x in $\lambda_k \mathcal{T}(\bar{x}_k)$.

Note that λ_k is not necessarily less than 1, it might actually enlarge the terminal set if this is possible with respect to the conditions of Theorem 3.2. Let us illustrate this with an example.

4.1 Example

Consider a discrete time system (artificially constructed to illustrate the behavior with terminal set enlargement) with A and B matrices

$$A = \begin{bmatrix} 0.9 & 0.5 \\ 0 & -0.8 \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ 0.6 \end{bmatrix}$$

and with the state constraints

$$\mathcal{X} = \left\{ x \mid \|x - p\|_1 \leq 10, \quad p = (7.5 \ \ 0)^T \right\}$$

and the control constraints

$$\mathcal{U} = \{u \mid -3 \leq u \leq 5\}$$

The terminal constraint set, $\mathcal{T}$, is calculated as an invariant set for an LQ state feedback control law

$$\kappa(x_k) = -Kx_k$$

where the feedback gain is

$$K = -(I + B^T PB)^{-1} B^T PA$$

and P is the solution to the Riccati equation with $Q = I$ and $R = 1$

$$P = A^T P(I + BB^T P)^{-1} A + C^T C$$

The terminal set for the nominal case, i.e., when the setpoint is the origin, will for this problem setup result in a relatively small terminal set polytope, $\mathcal{T}$, located in the small left part of the state constraint set, see Figure 4.3.

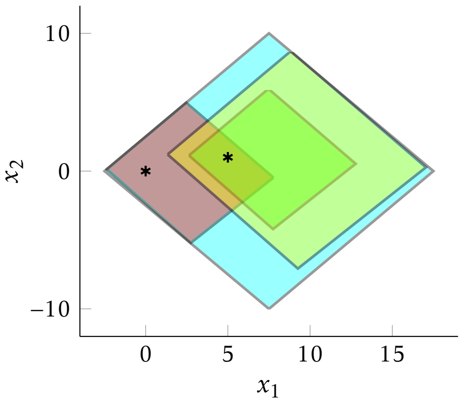

Figure 4.3: *Illustration of how the terminal set $\mathcal{T}$ can be scaled with a $\lambda > 1$. The state constraint set $\mathcal{X}$ is the outermost polytope. The original terminal state constraint set, $\mathcal{T}$, for zero reference, is the dark shaded polytope to the left which has the origin (marked with a black star) in its interior. This set translated to the new setpoint (also marked with a star) without any scaling is the inner most of the two polytopes located around $x_1 = 5$. The outer of the two polytopes is the translated set, scaled with $\lambda = 1.55$.*

When the system tracks a reference signal, the change of reference will then cause the terminal set to be shifted accordingly. From Figure 4.3 it is evident that $\mathcal{T}(\bar{x}) \subset \mathcal{X}$, i.e., the shifted terminal set lies in the strict interior of the feasible set. This means that it is possible to scale up the terminal set without violating the state constraints. However also the local control law needs to be feasible in the scaled terminal set, i.e., $\kappa(x) \in \mathcal{U} \;\; \forall \;\; x \in \lambda\mathcal{T}(\bar{x})$.

In this example $\mathcal{T}(\bar{x})$ can be scaled with a $\lambda \leq 1.55$ before the state constraints limit the scaling.

Modifying the MPC tracking algorithm (3.13) with the above constraints we arrive at the following high-level representation of the optimization problem

$$\underset{u,x,\lambda_k,\bar{x}_k,\bar{u}_k,\bar{r}_k}{\text{minimize}} \sum_{i=0}^{N-1} \ell(x_{k+i} - \bar{x}_k, u_{k+i} - \bar{u}_k) + \Psi(x_{k+N} - \bar{x}_k) + \phi(\bar{r}_k - r) \tag{4.3a}$$

$$\text{s.t.}$$

$$x_{k+i+1} = Ax_{k+i} + Bu_{k+i} \quad \forall \; i = 0, \ldots, N-1 \tag{4.3b}$$

$$x_{k+i} \in \mathcal{X} \quad \forall \; i = 0, \ldots, N-1 \tag{4.3c}$$

$$u_{k+i} \in \mathcal{U} \quad \forall \; i = 0, \ldots, N-1 \tag{4.3d}$$

$$x_{k+N} \in \lambda_k \mathcal{T}(\bar{x}_k) \tag{4.3e}$$

$$\lambda_k \mathcal{T}(\bar{x}_k) \subseteq \mathcal{X} \tag{4.3f}$$

$$\bar{u}_k - K(x - \bar{x}_k) \in \mathcal{U} \;\; \forall x \in \lambda_k \mathcal{T}(\bar{x}_k) \tag{4.3g}$$

$$\begin{bmatrix} A - I & B \\ C & D \end{bmatrix} \begin{bmatrix} \bar{x}_k \\ \bar{u}_k \end{bmatrix} = \begin{bmatrix} 0 \\ \bar{r}_k \end{bmatrix} \tag{4.3h}$$

$$\bar{x}_k \in \operatorname{int}_\epsilon(\mathcal{X}) \tag{4.3i}$$

$$\bar{u}_k \in \operatorname{int}_\epsilon(\mathcal{U}) \tag{4.3j}$$

In order to guarantee the convergence properties of the controller, our proof in section 4.2.4 requires the steady states and controls to be strictly feasible, i.e., the terminal set has to be placed in the strict interior of the feasible set. From a practical point of view we achieve this by constraining them to an ϵ-interior of the feasible set, constraints (4.3i) and (4.3j). Simulations indicate that for practical purposes any arbitrarily small ϵ can be chosen, even $\epsilon = 0$.

Unfortunately, as the problem is written in (4.3), it is not suitable for implementation. For example as the constraints (4.1) and (4.2a) are written now, it is not obvious that they are linear in λ_k and $\bar{x}_k$, and (4.2b) is an infinite-dimensional constraint. Hence, a reformulation is needed.

4.2.1 Vertex enumeration reformulation

As a first step we will show that the constraints in fact are linear in the variables and that the constraints can be formulated using a finite (although possibly large) number of linear constraints.

Let us begin with (4.1) which constrains the terminal state to be inside the scaled and translated terminal set. Since the terminal set is polytopic, we have a representation of the form $\mathcal{T} = \{x \mid F_{\mathcal{T}} x \leq b_{\mathcal{T}}\}$. With our definitions of translations and scaling from Chapter 2.2, we obtain a linear constraint.

$$F_{\mathcal{T}}(x_{k+N} - \bar{x}_k) \leq \lambda_k b_{\mathcal{T}} \tag{4.4}$$

The constraints (4.2a), which ensure the scaled and translated terminal set to be state feasible, and (4.2b), which ensures that the nominal control law $\bar{u}_k - K(x - \bar{x}_k)$ is feasible for any x in the scaled and translated terminal set, can be rewritten using the vertex form of the terminal set $\mathcal{T}$.

Let $\mathcal{T}$ have ν_p vertices v_j and it follows that $\lambda_k \mathcal{T}(\bar{x}_k)$ can be represented as the convex hull. By convexity, this polytope is a subset of $\mathcal{X}$ if and only if all vertices are. With $\mathcal{X} = \{x \mid F_{\mathcal{X}} x \leq b_{\mathcal{X}}\}$ we obtain

$$F_{\mathcal{X}} x \leq b_{\mathcal{X}}, \quad \forall \left\{ x \mid x = \bar{x}_k + \lambda_k v_j,\ \forall\ j = 1, \ldots, \nu_p \right\} \tag{4.5}$$

and we arrive at

$$F_{\mathcal{X}}(\bar{x}_k + \lambda_k v_j) \leq b_{\mathcal{X}}\ \forall\ j = 1, \ldots, \nu_p \tag{4.6}$$

Similarly for the control constraints, we have to ensure that

$$F_{\mathcal{U}}(\bar{u}_k - K(x - \bar{x}_k)) \leq b_{\mathcal{U}}, \quad \forall \left\{ x \mid x = \bar{x}_k + \lambda_k v_j,\ \forall\ j = 1, \ldots, \nu_p \right\} \tag{4.7}$$

Inserting the vertices leads to

$$F_{\mathcal{X}}(\bar{u}_k - \lambda_k K v_j) \leq b_{\mathcal{U}}\ \forall\ j = 1, \ldots, \nu_p \tag{4.8}$$

As the constraints (4.6) and (4.8) are derived now the main drawback is that they require the enumeration of all vertices in the terminal set. This can, even for simple polytopes, be extremely expensive [Grünbaum, 2003], e.g., in the case when $\mathcal{T}$ is a simple box in $\mathbb{R}^n$ it has $2n$ facets but 2^n vertices and for higher dimensions this difference becomes crucial from a computational complexity point of view. Therefore we seek to avoid performing the vertex calculations which can be done by rewriting the constraints (4.5) and (4.7) using duality theory.

4.2.2 Dual formulation of terminal set constraints

First note that the constraints (4.6) and (4.8) can be interpreted as uncertain constraints, i.e., they must hold for all $x \in \lambda_k \mathcal{T}(\bar{x}_k)$ and we can write them as

$$F_{\mathcal{U}}\left(\bar{u}_k - K\left(x - \bar{x}_k\right)\right) \leq b_{\mathcal{U}} \tag{4.9a}$$

$$F_{\mathcal{X}} x \leq b_{\mathcal{X}} \tag{4.9b}$$

for all x such that

$$\{x \mid F_{\mathcal{T}}\left(x - \bar{x}_k\right) \leq \lambda_k b_{\mathcal{T}}\}$$

Now consider a general uncertain affine constraint of the form

$$\left(c^T x + d\right) + (Ax + b)^T p \leq 0 \ \ \forall \ \left\{ x \mid F^T x \leq E \right\} \tag{4.10}$$

where x is an uncertain, but bounded, variable and p is the decision variable. This constraint must hold for all possible values of x, i.e., it must hold for the worst case x, in this case, the one that maximize the left hand side.

This maximum can be determined by the following optimization problem

$$\max_{x}. \ \left(A^T p + c\right)^T x + \left(b^T p + d\right) \ \text{ s.t. } F^T x \leq E \tag{4.11}$$

Additionally we know that for a solution p to satisfy (4.10) it must hold that the left hand side is negative and hence also gives a negative optimal cost for the optimization problem (4.11).

From duality theory, see, e.g., Boyd and Vandenberghe [2004] or Nesterov and Nemirovskii [1994], the dual problem of (4.11) can be formulated in the dual variable ζ as

$$\min_{\zeta}. \ E^T \zeta + \left(b^T p + d\right) \ \text{ s.t. } \ \zeta \geq 0, \ F\zeta = A^T p + c \tag{4.12}$$

Since the optimal value to the primal problem (4.11) is negative it then follows that the optimal value of the dual problem, which is a tight upper bound to the primal problem, is negative. Thus we can conclude that given a solution p and the existence of a dual variable $\zeta \geq 0$ such that

$$E^T \zeta + \left(b^T p + d\right) \leq 0, \ \ F\zeta = A^T p + c \tag{4.13}$$

is equivalent to that p satisfies the constraint (4.10) for all values of the uncertain variable x. This derivation has previously been presented in, e.g., Ben-Tal and Nemirovski [2002] in the context of robust optimization.

Let us now use this result to rewrite the constraints (4.6) and (4.8) into the dual form. Denote the decision variable $p_k = (\bar{x}_k, \bar{u}_k)^T$ and write each row of (4.9) generically as

$$f_x^T x + f_p^T p_k + d \leq 0 \ \ \forall \ \{x \mid F_{\mathcal{T}} (x - \bar{x}_k) \leq \lambda_k b_{\mathcal{T}}\}$$

where f_x^T denotes the rows of the matrix

$$F_x = \begin{bmatrix} -F_{\mathcal{U}} K \\ F_{\mathcal{X}} \end{bmatrix}$$

and f_p^T denotes the rows of

$$F_p = \begin{bmatrix} F_{\mathcal{U}} K & F_{\mathcal{U}} \\ 0 & 0 \end{bmatrix}$$

and d is the corresponding row element of the vector $\left(b_{\mathcal{U}}^T \ \ b_{\mathcal{X}}^T\right)^T$.

Writing this in the same structure as (4.10) gives

$$\left(f_x^T x + d\right) + \left(0x + f_p\right)^T p_k \leq 0 \ \ \forall \ \{x \mid F_{\mathcal{T}} x \leq \lambda_k b_{\mathcal{T}} + F_{\mathcal{T}} \bar{x}_k\}$$

and the dual form then becomes

$$(\lambda_k b_{\mathcal{T}} + F_{\mathcal{T}} \bar{x}_k)^T \zeta + f_p^T p_k + d \leq 0, \ \ F_{\mathcal{T}}^T \zeta = f_x, \ \zeta \geq 0$$

Expand the parenthesis and use the fact that $F_T^T \zeta = f_x$, this gives

$$\lambda_k b_T^T \zeta + \bar{x}_k^T f_x + f_p^T p_k + d \leq 0, \quad F_T^T \zeta = f_x, \ \zeta \geq 0$$

Hence, we should have a global solution (λ_k, p_k) such that there exist a ζ for every row satisfying this set of equations. Note that the term $b_T^T \zeta$ is non-negative (both b_T^T and ζ are non-negative) since the terminal set contains the origin in its interior. Hence, it is beneficial to make this term minimal for every row. Note also that this choice of $b_T^T \zeta$ is independent of the variables λ_k, p_k. Therefore let

$$\gamma = \min_{\zeta}.\ b_T^T \zeta \ \text{ s.t. } F_T^T \zeta = f_x, \ \zeta \geq 0 \tag{4.14}$$

and replace the term $b_T^T \zeta$ with the solution γ, finally giving

$$\lambda_k \gamma + \bar{x}_k^T f_x + f_p^T p_k + d \leq 0 \tag{4.15}$$

We can now replace each row of the uncertain constraints (4.6) and (4.8) with (4.15) and if we write this into a matrix form we obtain

$$\lambda_k \Gamma + \bar{x}_k^T \begin{bmatrix} -F_{\mathcal{U}} K \\ F_{\mathcal{X}} \end{bmatrix}^T + \begin{bmatrix} F_{\mathcal{U}} K & F_{\mathcal{U}} \\ 0 & 0 \end{bmatrix} \begin{pmatrix} \bar{x}_k \\ \bar{u}_k \end{pmatrix} \leq - \begin{pmatrix} b_{\mathcal{U}} \\ b_{\mathcal{X}} \end{pmatrix} \tag{4.16}$$

where the vector Γ has the solution γ to (4.14), for each f_x of F_x, as its elements.

The equations (4.4) and (4.16) gives a complete description of the terminal state constraint set.

4.2.3 The QP formulation

We can now summarize all the pieces into the quadratic program formulation of the proposed controller.

First select the different parts of the cost function in (4.3) as

$$\ell(x_{k+i} - \bar{x}_k, u_{k+i} - \bar{u}_k) = \|x_{k+i} - \bar{x}_k\|_Q^2 + \|u_{k+i} - \bar{u}_k\|_R^2 \tag{4.17a}$$

$$\Psi(x_{k+N} - \bar{x}_k) = \|x_{k+N} - \bar{x}_k\|_P^2 \tag{4.17b}$$

$$\phi(\bar{r}_k - r) = \beta \|\bar{r}_k - r\|_\infty \tag{4.17c}$$

In (4.17a), Q and R are positive definite weight matrices, used also to define the Lyapunov cost matrix P in (4.17b) and nominal state feedback K through

$$(A - BK)^T P (A - BK) - P = -Q - K^T R K \tag{4.18}$$

In (4.17c), β is a positive scalar, r is the desired reference to track and $\bar{r}_k$ is the pseudo reference variable. Later in this chapter we will discuss on how to choose the scalar β in order to achieve desired properties.

The proposed controller is then defined by the solution to the following quadratic programming problem

$$\underset{u,x,\lambda_k,\bar{x}_k,\bar{u}_k,\bar{r}_k}{\text{minimize}} \; \Psi(x_{k+N} - \bar{x}_k) + \phi(\bar{r}_k - r) + \sum_{i=0}^{N-1} \ell(x_{k+i} - \bar{x}_k, u_{k+i} - \bar{u}_k) \tag{4.19a}$$

s.t.

$$x_{k+i+1} = Ax_{k+i} + Bu_{k+i} \quad \forall\, i = 0, \dots, N-1 \tag{4.19b}$$

$$F_{\mathcal{X}} x_{k+i} \leq b_{\mathcal{X}} \quad \forall\, i = 0, \dots, N-1 \tag{4.19c}$$

$$F_{\mathcal{U}} u_{k+i} \leq b_{\mathcal{U}} \quad \forall\, i = 0, \dots, N-1 \tag{4.19d}$$

$$F_{\mathcal{T}}(x_{k+N} - \bar{x}_k) \leq \lambda_k b_{\mathcal{T}} \tag{4.19e}$$

$$\lambda_k \Gamma + \bar{x}_k^T F_x^T + F_p \begin{pmatrix} \bar{x}_k \\ \bar{u}_k \end{pmatrix} \leq -\begin{pmatrix} b_{\mathcal{U}} \\ b_{\mathcal{X}} \end{pmatrix} \tag{4.19f}$$

$$\begin{bmatrix} A - I & B \\ C & D \end{bmatrix} \begin{bmatrix} \bar{x}_k \\ \bar{u}_k \end{bmatrix} = \begin{bmatrix} 0 \\ \bar{r}_k \end{bmatrix} \tag{4.19g}$$

$$F_{\mathcal{X}}(\bar{x}_k) \leq (1-\epsilon)\, b_{\mathcal{X}}, \quad \epsilon > 0 \tag{4.19h}$$

$$F_{\mathcal{U}}(\bar{u}_k) \leq (1-\epsilon)\, b_{\mathcal{U}}, \quad \epsilon > 0 \tag{4.19i}$$

Solving the above problem gives an optimal control input sequence $\{u_{k+i}^*\}_{i=0}^{N-1}$ and the applied MPC law is the first element of this optimal sequence

$$\kappa(x_k) = u_k^* \tag{4.20}$$

It should be noted that the above formulation, in strict sense, is not a quadratic program on standard form since the cost function contains the infinity norm. However it is a trivial task to reformulate this into the standard QP form (2.2).

4.2.4 Stability and feasibility of the proposed algorithm

In the following theorem we establish the stability and convergence properties of the proposed controller that was derived in the previous section. Before we outline the stability properties of our proposed algorithm we must make some necessary assumptions and definitions.

4.2 Definition. $r_\perp$ is defined as the closest strictly feasible point to the reference r in a distance measure determined by the function $\phi(\,\cdot\,)$

$$r_\perp = \arg\min_{\bar{r}} \; \phi(\bar{r} - r)$$

$$\text{s.t.}$$

$$\begin{bmatrix} A - I & B \\ C & D \end{bmatrix} \begin{bmatrix} \bar{x} \\ \bar{u} \end{bmatrix} = \begin{bmatrix} 0 \\ \bar{r} \end{bmatrix}$$

$$\bar{x} \in \text{int}_\epsilon(\mathcal{X})$$

$$\bar{u} \in \text{int}_\epsilon(\mathcal{U})$$

4.3 Assumption. The matrix

$$\begin{bmatrix} A-I & B \\ C & D \end{bmatrix}$$

is such that there exists a solution to (3.11) for any reference r.

4.4 Assumption. $\mathcal{X}, \mathcal{U}$ and $\mathcal{T}$ contain the origin in their interior.

4.5 Lemma. *Let $\mathcal{T}$ be a positively invariant set for a stable discrete time linear system, $x_{k+1} = A_c x_k$. Then for any scalar $\lambda \geq 0$, the scaled set, $\lambda\mathcal{T}$ is also a positively invariant set for the system.*

Proof: The lemma follows directly from the scaling invariance of linear systems, see, e.g., Blanchini [1999]. □

With these definitions and assumptions in order, we are ready to formulate the main stability theorem for the proposed controller.

4.6 Theorem. *For any feasible initial state x_0 and Assumptions 4.3 and 4.4 , the MPC algorithm defined through* (4.19) - (4.20) *remains feasible and stabilizes the system* (3.9). *Additionally, x_k asymptotically converges to a setpoint given by the projection of the reference r onto the (ϵ-contracted) feasible set.*

The proof of the theorem is relatively straightforward and recursive feasibility and convergence of the state to a stationary point follow standard proofs found in the literature [Mayne et al., 2000]. In a second step, to show that the setpoint to which the state converges is the setpoint associated with the given reference r, if feasible, or the setpoint corresponding to the closest feasible reference has previously been proven in similar ways in Ferramosca et al. [2009].

Proof: Let $\mathcal{X}_N$ be the set of x where (4.19) is feasible. Assume $x_k \in \mathcal{X}_N$ with an optimal solution given by the sequence $\{u^*_{k+i}\}_{i=0}^{N-1}$ and λ^*_k and $\bar{r}^*_k$, with predicted state trajectory $\{x^*_{k+i}\}_{i=1}^{N}$.

At the next time step $\{\hat{u}_{k+i}\}_{i=1}^{N} = \{u^*_{k+1}, u^*_{k+2}, \ldots, u^*_{k+N-1}, \bar{u}^*_k - K(x^*_{k+N} - \bar{x}^*_k)\}$ is a feasible control sequence, since $\bar{u}^*_k - K(x^*_{k+N} - \bar{x}^*_k)$ is feasible according to (4.3g). Furthermore, we use $\lambda_{k+1} = \lambda^*_k$ and $\bar{r}_{k+1} = \bar{r}^*_k$. Keeping λ_{k+1} and $\bar{r}_{k+1}$ unchanged means that we keep the scaled and translated terminal set unchanged. The new, possibly suboptimal, state sequence is $\{\hat{x}_{k+i}\}_{i=2}^{N+1}$ where $x_{k+N+1} - \bar{x}^*_k = (A-BK)(x^*_{k+N} - \bar{x}^*_k)$. Since $\lambda^*_k\mathcal{T}$ is positively invariant w.r.t. the system $x_{k+1} = (A-BK)x_k$ according to Lemma 4.5, it follows that $x_{k+N+1} - \bar{x}^*_k$ stays in $\lambda^*_k\mathcal{T}$, i.e., the terminal state constraint $x_{k+N+1} \in \lambda^*_k\mathcal{T}(\bar{x}^*_k)$ is satisfied. Since $\lambda^*_k\mathcal{T}(\bar{x}^*_k) \subseteq \mathcal{X}$, state constraints are trivially satisfied.

Now assume that we have an optimal solution at time k and denote the optimal cost

$$\mathcal{J}_k^* = \sum_{i=0}^{N-1}\left(\left\|x_{k+i}^* - \bar{x}_k^*\right\|_Q^2 + \left\|u_{k+i}^* - \bar{u}_k^*\right\|_R^2\right) + \left\|x_{k+N}^* - \bar{x}_k^*\right\|_P^2 + \phi(\bar{r}_k^* - r)$$

Applying the control sequence $\{\hat{u}_{k+i}\}_{i=1}^N$ defined in the previous section gives the suboptimal cost $\mathcal{J}_{k+1}$ as

$$\begin{aligned}
&\sum_{i=0}^{N-1}\left(\left\|x_{k+1+i}^* - \bar{x}_k^*\right\|_Q^2 + \left\|u_{k+1+i}^* - \bar{u}_k^*\right\|_R^2\right) + \left\|x_{k+1+N} - \bar{x}_k^*\right\|_P^2 + \phi(\bar{r}_k^* - r)\\
&= \sum_{i=0}^{N-2}\left(\left\|x_{k+1+i}^* - \bar{x}_k^*\right\|_Q^2 + \left\|u_{k+1+i}^* - \bar{u}_k^*\right\|_R^2\right) + \left\|x_{k+N}^* - \bar{x}_k^*\right\|_Q^2\\
&+ \left\|\bar{u}_k^* - K(x_{k+N} - \bar{x}_k^*) - \bar{u}_k^*\right\|_R^2 + \left\|x_{k+1+N} - \bar{x}_k^*\right\|_P^2 + \phi(\bar{r}_k^* - r)\\
&+ \left\|x_k^* - \bar{x}_k^*\right\|_Q^2 + \left\|u_k^* - \bar{u}_k^*\right\|_R^2 + \left\|x_{k+N}^* - \bar{x}_k^*\right\|_P^2\\
&- \left\|x_k^* - \bar{x}_k^*\right\|_Q^2 - \left\|u_k^* - \bar{u}_k^*\right\|_R^2 - \left\|x_{k+N}^* - \bar{x}_k^*\right\|_P^2\\
&= \underbrace{\sum_{i=0}^{N-1}\left(\left\|x_{k+i}^* - \bar{x}_k^*\right\|_Q^2 + \left\|u_{k+i}^* - \bar{u}_k^*\right\|_R^2\right) + \left\|x_{k+N}^* - \bar{x}_k^*\right\|_P^2 + \phi(\bar{r}_k^* - r)}_{\mathcal{J}_k^*}\\
&\underbrace{+ \left\|x_{k+N}^* - \bar{x}_k^*\right\|_Q^2 + \left\|K(x_{k+N}^* - \bar{x}_k^*)\right\|_R^2 + \left\|x_{k+1+N} - \bar{x}_k^*\right\|_P^2 - \left\|x_{k+N}^* - \bar{x}_k^*\right\|_P^2}_{=0 \text{ due to } (4.18)}\\
&- \left\|x_k^* - \bar{x}_k^*\right\|_Q^2 - \left\|u_k^* - \bar{u}_k^*\right\|_R^2
\end{aligned}$$

and thus, it follows that the suboptimal cost is

$$\mathcal{J}_{k+1} = \mathcal{J}_k^* - \left\|x_k^* - \bar{x}_k^*\right\|_Q^2 - \left\|u_k^* - \bar{u}_k^*\right\|_R^2$$

which implies that the optimal cost $\mathcal{J}_{k+1}^*$ fulfills

$$\mathcal{J}_{k+1}^* \leq \mathcal{J}_{k+1} = \mathcal{J}_k^* - \left\|x_k^* - \bar{x}_k^*\right\|_Q^2 - \left\|u_k^* - \bar{u}_k^*\right\|_R^2 < \mathcal{J}_k^*$$

In other words, $\mathcal{J}_k^*$ is strictly decreasing as long as $x_k^* \neq \bar{x}_k^*$ and $u_k^* \neq \bar{u}_k^*$. Hence $x_k^* \to \bar{x}_k^*$ and $u_k^* \to \bar{u}_k^*$. Note that in the limit we have, since $\bar{x}_k^*$ and $\bar{u}_k^*$ represent a stationary pair, that $\bar{x}_k^* = x_k^* = x_{k+1}^* = \bar{x}_{k+1}^*$, i.e., the pseudo setpoint converges too.

To show convergence of $\bar{r}_k^* \to r_\perp$, assume that the system has settled at a setpoint given by $\bar{x}_k^*, \bar{u}_k^*$, defined by $\bar{r}_k^*$. The proof will proceed by contradiction, so we assume $\bar{r}_k^* \neq r_\perp$. Consider a perturbation $(0 \leq \gamma < 1)$ of the pseudo reference $\bar{r}_k^*$ towards $r_\perp$, given by

$$\bar{r}_\gamma = \gamma \bar{r}_k^* + (1-\gamma) r_\perp$$

Our first step is to show that this choice is feasible for γ sufficiently close to 1. By convexity, $\bar{r}_\gamma$ is feasible with respect to (4.19h) and (4.19i). We use the constant control sequence corresponding to the steady state control given by $\bar{r}_\gamma$, i.e. $u_{k+i} = \bar{u}_\gamma$ (which is also feasible by convexity) and the predicted states evolve according to

$$
\begin{aligned}
x_{k+1} &= A\bar{x}_k^* + B\bar{u}_\gamma \\
&= A\bar{x}_k^* + \gamma B\bar{u}_k^* + (1-\gamma)Bu_\perp \\
&= A(\gamma\bar{x}_k^* + (1-\gamma)x_\perp + \bar{x}_k^* - \gamma\bar{x}_k^* - (1-\gamma)x_\perp) + \gamma B\bar{u}_k^* + (1-\gamma)Bu_\perp \\
&= \gamma(A\bar{x}_k^* + B\bar{u}_k^*) + (1-\gamma)(Ax_\perp + Bu_\perp) + (1-\gamma)A(\bar{x}_k^* - x_\perp) \\
&= \gamma\bar{x}_k^* + (1-\gamma)x_\perp + (1-\gamma)A(\bar{x}_k^* - x_\perp) \\
&= \bar{x}_\gamma + (1-\gamma)A(\bar{x}_k^* - x_\perp)
\end{aligned}
$$

Applying the same manipulations recursively leads to

$$x_{k+i} = \bar{x}_\gamma + (1-\gamma)A^i(\bar{x}_k^* - x_\perp)$$

For γ sufficiently close to 1 all predicted states are feasible w.r.t. $\mathcal{X}$, since $\bar{x}_k^*$ and $x_\perp$ (and thus $\bar{x}_\gamma$) are strictly inside $\mathcal{X}$ and $(1-\gamma)A^i(\bar{x}_k^* - x_\perp)$ approaches 0 as γ goes to 1. What remains to show is that we can select λ_k such that $x_{k+N+1} \in \lambda_k\mathcal{T}(\bar{x}_\gamma)$ and $\lambda_k\mathcal{T}(\bar{x}_\gamma) \subseteq \mathcal{X}$.

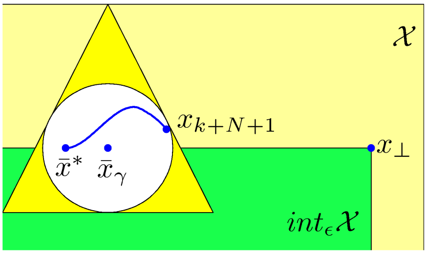

Figure 4.4: *Illustration of the components in the proof of convergence of the pseudo reference. The figure shows portions of the sets $\mathcal{X}$ and $\mathrm{int}_\epsilon(\mathcal{X})$, and the triangular set $\lambda_k\mathcal{T}(\bar{x}_\gamma)$ with its inscribed Euclidean ball.*

Since the pseudo reference is in the strict interior (defined by ϵ), it immediately follows that there exist a constant $\epsilon_\lambda > 0$, determined by the geometry of $\mathcal{T}, \mathcal{X}, \mathcal{U}$ and ϵ, such that $\epsilon_\lambda\mathcal{T}(\bar{x}) \subseteq \mathcal{X}$ for any strictly feasible $\bar{x}$. Let d denote the radius of the largest possible Euclidean ball centered at the origin which can be inscribed in $\mathcal{T}$, see Figure 4.4. Since $\mathcal{T}$ contains 0 in its interior by assumption, $d > 0$. The distance from the terminal state x_{k+N+1} to the new pseudo setpoint is given by $\left\|x_{k+N+1} - \bar{x}_\gamma\right\| = (1-\gamma)\left\|A^{N+1}(\bar{x}_k^* - x_\perp)\right\|$. If this distance is smaller than $\lambda_k d$, the terminal state is inside the scaled and translated terminal set. Hence, if $\gamma \geq 1 - \frac{\epsilon_\lambda d}{\left\|A^{N+1}(\bar{x}_k^* - x_\perp)\right\|}$ the terminal state constraint is fulfilled. Since $\mathcal{X}$ is polytopic, the denominator in the expression has an upper bound.

Returning back to the objective function for our proposed feasible solution, and using the notation $\Psi_i = A^i(\bar{x}_k^* - x_\perp)$, we arrive at

$$\begin{aligned}\mathcal{J}_k &= \left\|(1-\gamma)\Psi_N\right\|_P^2 + \sum_{i=0}^{N-1} \left\|(1-\gamma)\Psi_i\right\|_Q^2 + \underbrace{\left\|u_{k+i} - \bar{u}_\gamma\right\|_R^2}_{=0} + \phi\left(\gamma \bar{r}_k^* + (1-\gamma) r_\perp - r\right) \\ &= (1-\gamma)^2 \left\|\Psi_N\right\|_P^2 + (1-\gamma)^2 \sum_{i=0}^{N-1} \left\|\Psi_i\right\|_Q^2 + \phi\left(\gamma(\bar{r}_k^* - r_\perp) + (r_\perp - r)\right)\end{aligned}$$

Differentiate $\mathcal{J}_k$ with respect to the step size γ

$$\frac{\partial \mathcal{J}_k}{\partial \gamma} = -2(1-\gamma)\left(\left\|\Psi_N\right\|_P^2 + \sum_{i=0}^{N-1} \left\|\Psi_i\right\|_Q^2\right) + c^T(\bar{r}_k^* - r_\perp)$$

where c is the subgradient to the function $\phi(\,\cdot\,)$ at $\gamma = 1$. Evaluating this at $\gamma = 1$ gives

$$\left.\frac{\partial \mathcal{J}_k}{\partial \gamma}\right|_{\gamma=1} = c^T(\bar{r}_k^* - r_\perp)$$

If this inner product is positive, the cost function decreases as γ decreases which in turn implies that the cost can be reduced by moving $\bar{r}_k^*$ closer to $r_\perp$. From the definition of the subgradient it follows that

$$\phi(r_\perp - r) \geq \phi(\bar{r}_k^* - r) + c^T(r_\perp - \bar{r}_k^*)$$

which gives

$$c^T(\bar{r}_k^* - r_\perp) \geq \phi(\bar{r}_k^* - r) - \phi(r_\perp - r)$$

Since $r_\perp$ by definition is the closest feasible point to r in the chosen norm, the right hand side is strictly greater than zero unless $\bar{r}_k^* = r_\perp$. This means that the cost, $\mathcal{J}_k$, can be improved by making an arbitrarily small move of $\bar{r}_k^*$ towards $r_\perp$ and hence the only stationary point for $\bar{r}_k$ is $r_\perp$. Since we previously proved that x_k asymptotically converges to $\bar{x}_k$, and now proved that $\bar{x}_k$ converges to $x_\perp$ we can conclude that as x_k comes sufficiently close to $\bar{x}_k$ the cost will be reduced by moving $\bar{x}_k$ closer to $x_\perp$ and hence, x_k will asymptotically converge to $x_\perp$. $\square$

Note that it is not self-evident that this tracking MPC algorithm, using pseudo setpoints, is locally optimal in the sense that it minimizes the infinite horizon LQR cost in a vicinity of the setpoint, or in other words that it gives the same solution as the infinite LQ controller when possible.

However in the recent work by Ferramosca et al. [2011] the authors argue that under certain conditions a similar type of MPC algorithm has the local optimality property. In fact if the pseudo reference penalty is an exact penalty function, then it directly follows that the controller has the local optimality property. Since the same arguments can be used to show that also this algorithm possesses the local optimality property will we only briefly outline the train of thought and leave the details to the reader.

If $\phi(\bar{r}_k - r) = \beta \|\bar{r}_k - r\|_\infty$ and β is chosen such that $\phi(\bar{r}_k - r)$ constitutes an exact penalty function, as described in Section 3.2.4, then for all x_k where $\phi(\bar{r}_k - r) = 0$ is a feasible solution to (4.19), we will have $\mathcal{J}_k^* = \hat{\mathcal{J}}_k^*$, where $\hat{\mathcal{J}}_k^*$ is the solution to the related optimization problem

$$\underset{u,x,\lambda_k,\bar{x}_k,\bar{u}_k,\bar{r}_k}{\text{minimize}} \quad \Psi(x_{k+N} - \bar{x}_k) + \sum_{i=0}^{N-1} \ell(x_{k+i} - \bar{x}_k, u_{k+i} - \bar{u}_k) \tag{4.21}$$

subject to constraints (4.19b) - (4.19i) and the constraint $\phi(\bar{r}_k - r) = 0$ and where $\Psi(\cdot)$ is defined by (4.17b) and $\ell(\cdot)$ by (4.17a). Note that the optimization problem (4.21) is a dual mode formulation of the MPC controller (although translated to a new origin).

The results from Sznaier and Damborg [1987] show that the dual mode MPC formulation, with terminal state penalty equal to the infinite horizon unconstrained LQ cost and with a local controller that is the unconstrained LQ controller, has the local optimality property, i.e., the finite horizon cost equals that of the infinite horizon LQ problem, $\mathcal{J}_\infty$. From this it is clear that for all x_k where $\mathcal{J}_k^* = \hat{\mathcal{J}}_k^*$ it holds also that $\mathcal{J}_k^* = \mathcal{J}_\infty^*$. Note that the set of states, for which the local optimality results hold for the proposed controller, is not maximally large. This can be realized when considering that the scaled terminal set is not the maximum invariant terminal set that can be constructed for a given setpoint.

4.3 Examples from the aeronautical industry

In this section we illustrate the proposed method by two examples from the aeronautical industry. The first example considers maneuver limiting of an unstable fighter aircraft and in the second example we consider flight envelope protection for a small scale unmanned helicopter.

4.3.1 Maneuver limitations on a fighter aircraft

We will in this example compare the proposed controller to the reference tracking MPC algorithm first proposed in Limon et al. [2008] and then analyzed and extended in Ferramosca et al. [2009] and Ferramosca et al. [2011]. We will refer to this method as the *reference method*. Both controllers have been tuned using the same weighting matrices.

In this example we consider the short period dynamics (1.4) for the unstable generic fighter aircraft model ADMIRE [Forssell and Nilsson, 2005], discretized using a sample-time of 16ms.

$$\begin{bmatrix} \alpha_{k+1} \\ q_{k+1} \end{bmatrix} = \underbrace{\begin{bmatrix} 0.9719 & 0.0155 \\ 0.2097 & 0.9705 \end{bmatrix}}_{A} \begin{bmatrix} \alpha_k \\ q_k \end{bmatrix} + \underbrace{\begin{bmatrix} 0.0071 \\ 0.3263 \end{bmatrix}}_{B} \delta_k$$

$$y = \underbrace{\begin{bmatrix} 1 & 0 \end{bmatrix}}_{C} \begin{bmatrix} \alpha_k \\ q_k \end{bmatrix}$$

The maneuver limits for angle of attack and pitch rate have been set to

$$x_k \in \mathcal{X} = \left\{ (\alpha, q)^T \mid -15 \leq \alpha \leq 30, -100 \leq q \leq 100 \right\}$$

The elevator control surface angle deflection limits have been set to 25°

$$\delta_k \in \mathcal{U} = \{ \delta \mid -25 \leq \delta \leq 25 \}$$

The objective is to have the state α track a reference r given by the pilots control stick input, as close as possible to the boundary of the feasible set $\mathcal{X}$.

The weighting matrices in the costfunction (4.17a) have been chosen as

$$Q = \begin{bmatrix} 10 & 0 \\ 0 & 1 \end{bmatrix}, \quad R = 1$$

and P in (4.17b) is the Lyapunov cost of the corresponding LQ controller. The prediction horizon is chosen to $N = 20$ and ϵ is chosen small enough to not have any noticeable effect, $\epsilon = 10^{-5}$.

This example been implemented in Matlab with the use of the toolboxes YALMIP [Löfberg, 2004] and MPT [Herceg et al., 2013].

Nominal performance

The step response for both controllers are shown in Figure 4.5. The response in angle of attack is very similar between the proposed method and the reference method, only a slightly faster convergence can be observed for the proposed controller as the angle of attack approaches the border of the feasible set. Note that α converges to the desired reference if feasible. When setting the reference to $\alpha = 30°$, i.e., when it is located on the border of $\mathcal{X}$ the output will track the reference, but when the reference is set to $\alpha = -20°$, i.e., outside the feasible set, the output will track the pseudo reference that converges to the closest feasible point.

Since the terminal constraint set is a scaled version of the nominal invariant set, i.e., the one calculated for $r = 0$, it is clearly not the maximal invariant set and hence, it can result in a smaller domain of attraction. This drawback becomes evident only for shorter prediction horizons, e.g., for $N = 5$ the proposed controller has a smaller domain of attraction than the reference method, see Figure 4.6, while for $N = 10$ there is no difference between the two controllers. Therefore one must make a tradeoff between the complexity, i.e., the prediction horizon, and the needed domain of attraction.

Complexity of explicit solution

Comparing the complexity, i.e., the size of the QP, of the proposed controller with the reference method we can conclude that the proposed controller results in a

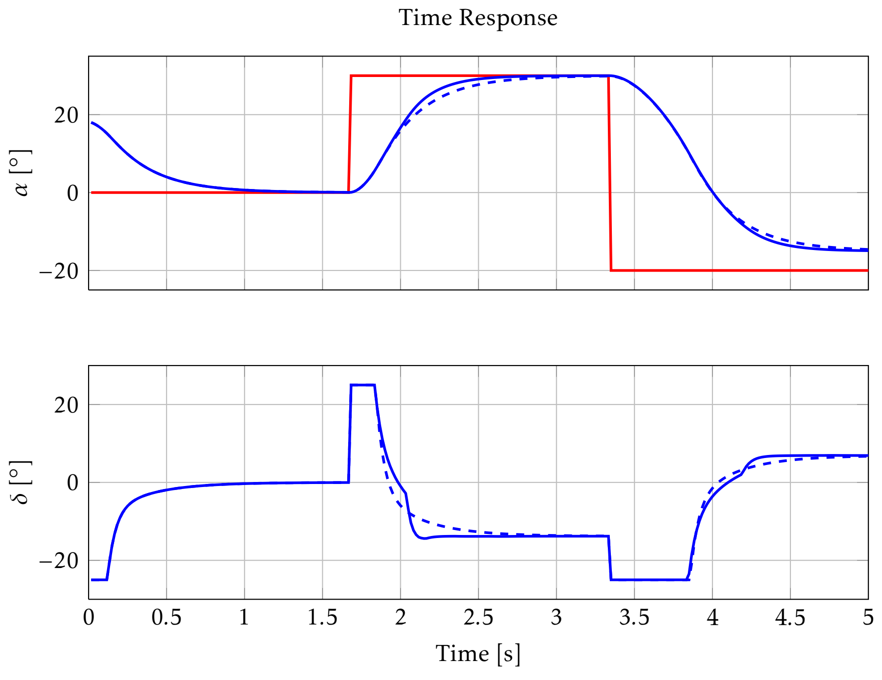

Figure 4.5: *The upper axis shows the pilot input reference, r (red, step shaped signal) and the output, α, for the two controllers. The proposed controller is the solid line and the reference controller is the dashed line. The lower axis shows the control signal for both controllers.*

large reduction in number of constraints. The reference method has 66 variables and 229 constraints while the proposed controller has 67 variables, but only 164 constraints. The significant difference in number of constraints comes from the terminal constraint set which is defined with 72 inequalities, in $\mathbb{R}^3$, for the reference method compared to only 4 inequalities, in $\mathbb{R}^2$, for the proposed controller. The large amount of inequalities needed to describe the terminal set for the reference method comes from the structure of the augmented system and has been noted by Limon et al. [2008]. In fact the authors state that it might not be possible to determine the terminal set with a finite number of constraints and if that is the case one has to shrink the terminal set such that the reference is constrained to an interior of the feasible set.

This difference in complexity will result in lower computational effort for online solution of the QP or a reduced complexity of an explicit implementation. When calculating the explicit MPC solution, the resulting controller for the reference method, with a prediction horizon of $N = 10$, has $N_{\mathcal{X}} = 6850$ partitions, compared to the explicit solution for the proposed method which has only $N_{\mathcal{X}} = 840$ partitions, i.e., a reduction of the number of partions with 87%. In Figure 4.7 the state space partitioning for a zero reference is shown. Compared to the pure

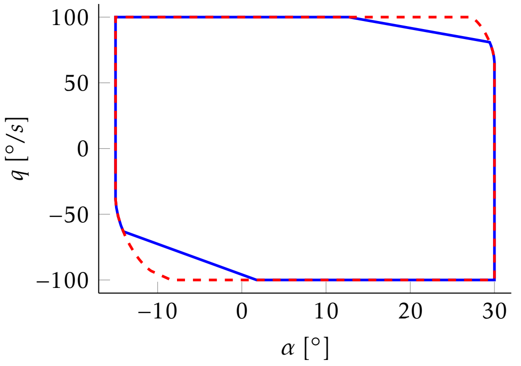

Figure 4.6: *The domain of attraction for the proposed controller (blue solid) and the reference method (red dashed line) for prediction horizon* $N = 5$.

stabilizing controller that was calculated in Section 3.2.5 we can see that adding the possibility of reference tracking increases the complexity of the solution also for zero references.

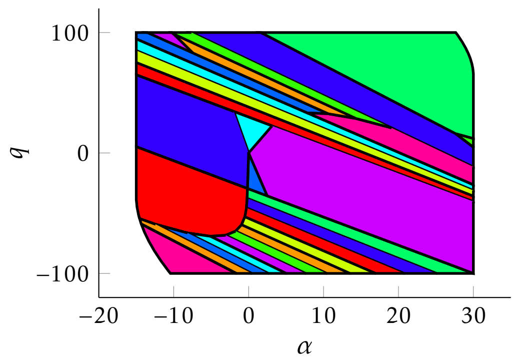

Figure 4.7: *State space partition of the explicit solution for reference,* $r = 0$.

Figure 4.8 shows the optimal control signal as a (piecewise affine) function of the states for different references. It is clear that the optimal feedback varies considerably with the reference. For zero reference (the upper left figure) one can recognize it as the saturated LQ solution from Chapter 3 and for reference values of $r = -10$ (upper right figure) and $r = 20$ (lower left figure) there are small regions, i.e., inside the invariant set, where the LQ solution is still valid. For the reference $r = 30$ (the lower right figure) we can see that the control signal is saturated for the most part of the state space and only for large angular rates there is a varying state feedback. For this reference value the terminal constraint set is scaled down to a single point and hence nothing of the original LQ solution is in the feedback.

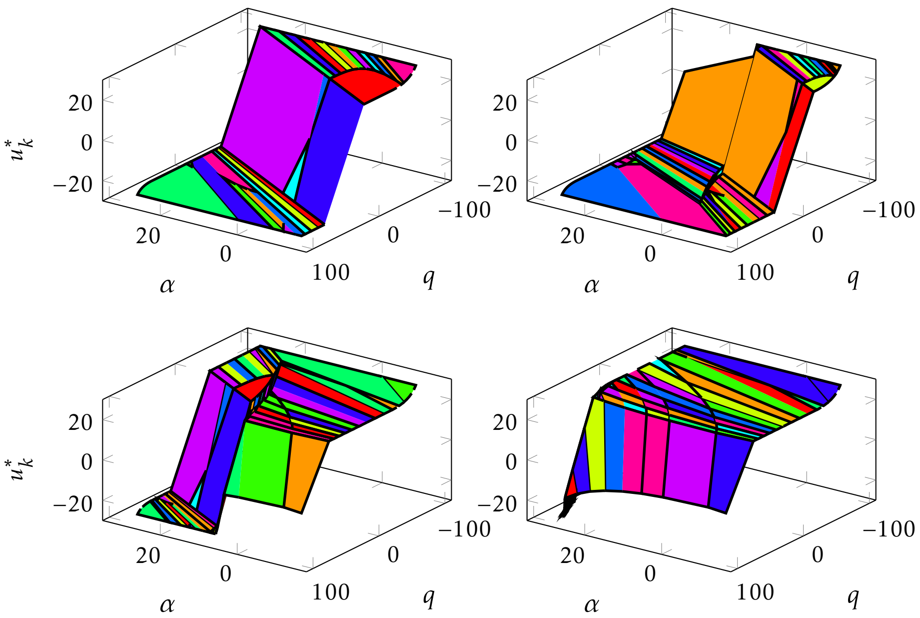

Figure 4.8: *Explicit feedback solution for different reference signals. Upper left axis show the feedback solution for $r = 0$, upper right axis shows the feedback solution for $r = -10$ and the two axis on the bottom shows the solution for $r = 20$ (left) and $r = 30$ (right).*

Robust performance and integral control

In this section we look at the performance of the proposed controller when the true system is different than that of the prediction model in the controller. Since both the proposed controller and the reference controller have similar performance characteristics we will restrict the discussion to the proposed method.

Let the true system have a 25% larger destabilizing pitching moment, i.e., the $(2, 1)$ element of the A-matrix is 25% larger than in the model. Additionally we let the true B-matrix be scaled with a factor of 0.8, i.e., the true system has 20% less control surface effectiveness than modelled.

The time response of the closed loop system with the proposed controller and the true system is shown in Figure 4.9. The initial response is slightly slower than for the case where there are no model errors and the angle of attack response overshoots the reference signal, settling at steady state value that is larger than the reference. This overshoot will result in violations of the state constraints when the reference approaches the boundary of constraint set.

To overcome this difficulty we add integral control as described in Section 3.2.3. We can see from Figure 4.9 that the performance is much better and the reference is attained without any steady state error. Even though the disturbance variable

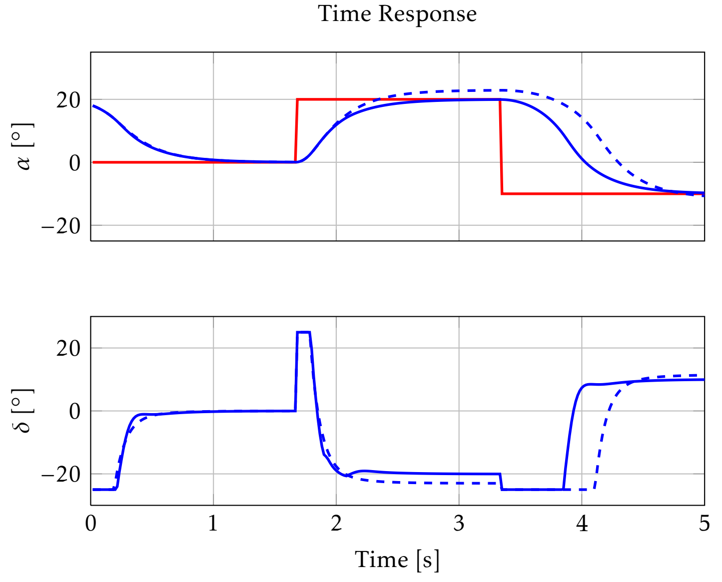

Figure 4.9: *Time response of the proposed controller (dashed) when the true system is different than that of the prediction model. The solid line is the time response of the proposed controller with added integral control.*

that is estimated in the disturbance observer is modelled as a constant disturbance, it can clearly catch dynamic behavior such as model errors.

In this thesis we do not investigate further on the theoretical properties of integral control, but leave this as an open topic for future research.

4.3.2 Helicopter flight envelope protection

In this example we apply the developed method to control the forward speed of an Yamaha R-max helicopter using the continuous five state model linearized around hover flight from Mettler [2002]

$$\begin{bmatrix} \dot{v} \\ \dot{q} \\ \dot{\theta} \\ \dot{a} \\ \dot{c} \end{bmatrix} = \begin{bmatrix} -0.0505 & 0 & -9.81 & -9.81 & 0 \\ -0.0561 & 0 & 0 & 82.6 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & -21.7391 & 14 \\ 0 & -1 & 0 & 0 & -0.342 \end{bmatrix} \begin{bmatrix} v \\ q \\ \theta \\ a \\ c \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ -2.174 \\ -0.7573 \end{bmatrix} \delta_s \tag{4.22}$$

which we have discretized with a sample rate of 60Hz. The states consist of the fuselage motion, i.e, the forward speed, v, the pitch rate, q and the pitch angle, θ, and also two states for the rotor dynamics. The first state of the rotor dynamics, a, is the pitch angle of the virtual rotor disc that is formed from the rotor blade rotation. The second state, c, is corresponding angle for the stabilizer bar. The control signal input, δ_s, is the so called swash plate angle.

The state constraints are formed by upper and lower limits on each variable

$$-5 \leq v \leq 10, \quad |q| \leq 5, \quad |\theta| \leq 3, \quad |a| \leq 1, \quad |c| \leq 2$$

additionally there are upper and lower limits on the control signal, $|\delta_s| \leq 5$.

In this example we have set the prediction horizon to $N = 10$, the weights in the cost function to

$$W = 10^4, \quad Q = \begin{bmatrix} 50 & 0 & 0 & 0 & 0 \\ 0 & 0.1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 10 & 0 \\ 0 & 0 & 0 & 0 & 0.1 \end{bmatrix}, \quad R = 1$$

and P again as the Lyapunov cost of the corresponding LQ controller. In this setup the QP problem has 72 variables and 296 constraints, out of which 96 constraints come from the terminal constraint set. As a comparison the terminal constraint set from the method of Limon et al. [2008] has 354 constraints.

Figure 4.10 shows how the variable λ_k varies over time when the system tracks a given reference input. From the figure we can see that the main changes in the

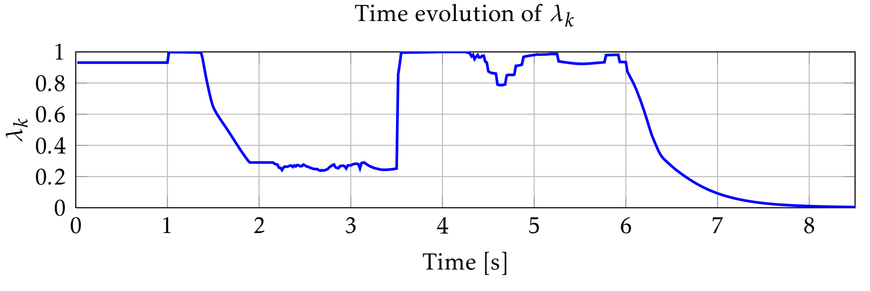

Figure 4.10: *The variation of the scaling variable, λ, over time as the system tracks a given speed reference.*

scaling variable come from the changes in the reference signal (compare with Figure 4.11). When the reference is changed from 0 to 9 m/s at time 1 second, then λ is increased to its maximum size in order to minimize the difference between the pseudo reference and the true reference. The same happens at time 3.5 seconds when the reference is set back to 0. Towards the end of the simulation the pseudo reference pushes the set point out to the border of the feasible set and hence λ is reduced to zero, scaling the terminal constraint set down to a single point. The small changes, or scattering, in λ that occur between 2 and 3.5 seconds and also between 4 and 6 seconds are due to the fact that the choice of λ is not unique and hence depends on the algorithm used to solve the QP. Note that this scattering in λ is internal in the controller and does not affect the applied control signal.

The step responses for the different state variables and also the optimal control signal are shown in Figure 4.11. The upper figure show the speed reference in red, the pseudo reference in green, the actual speed, v, in blue and the pitch angle, θ, and pitch rate, q, in magenta and cyan respectively. The middle figure shows the rotor disc pitch angle, a, in blue and the stabilizer bar pitch angle, c, in green. The bottom figure shows the control input, δ_s.

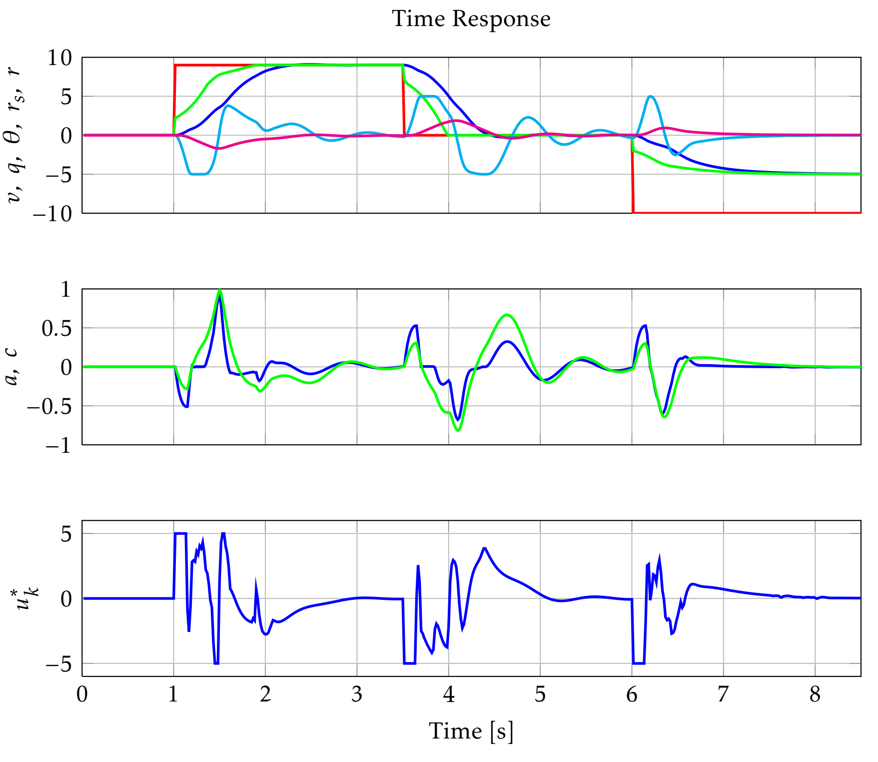

Figure 4.11: *Response of the helicopter subjected to step changes in the speed reference. The upper most axis show the fuselage states and the reference, the middle axis show the rotor states and the bottom axis show the control signal.*

The controller is able to track the speed reference as long as it is a feasible reference. At 6 seconds the speed reference changes to -10 m/s, which is outside the feasible set and then the pseudo reference converges to the closest feasible value.

On the contrary to the previous example it is not the output, v, that is the critical variable to limit. Instead it is the pitch angle, θ, and pitch rate, q, that we want to limit while tracking the speed as good as possible. From the figure we can see that the pitch rate reaches its upper and lower limits during the transients of the different step responses.

It is worth noting that the control signal seems to have a fairly agressive nature, but this comes primarily from the tuning of the controller and can be reduced if the controller is retuned. However we have tuned the controller such that the properties of the algorithm should be prominent.

5

Method for Guaranteed Stability and Recursive Feasibility in Nonlinear MPC

The main drawback with MPC is that it requires an iterative online solution of an optimization problem. This is in general fairly computationally expensive and has so far limited MPC's practical use for nonlinear systems.

To reduce the computational burden of nonlinear MPC, feedback linearization together with linear MPC has been successfully used to control nonlinear systems. The feedback linearization is used as an inner loop to obtain linear dynamics from input to output of the reference system in the MPC controller. The MPC controller is then used as an outer loop to obtain desired dynamics and constraint satisfaction. The main drawback is that this results in an optimization problem with nonlinear constraints on the control signal.

Several methods to approximate the nonlinear constraints have been proposed in the literature, many working in an ad hoc fashion, resulting in conservatism, or worse, inability to guarantee recursive feasibility. Also several methods work in an iterative manner which can be quit time consuming making them inappropriate for fast real time applications.

In this chapter we propose a method to handle the nonlinear constraints, using a set of dynamically generated local inner polytopic approximations. The main benefit of the proposed method is that while computationally cheap it still can guarantee recursive feasibility and convergence.

This chapter is an edited and extended version of the following conference paper

> D. Simon, J. Löfberg, and T. Glad. Nonlinear Model Predictive Control using Feedback Linearization and Local Inner Convex Constraint Approximations. In *Proceedings of the 2013 European Control Conference*, number 3, pages 2056–2061, 2013.

5.1 Introduction

As described in Section 3.3, when controlling nonlinear systems the resulting MPC optimization problem will be a nonconvex problem which in general is quite difficult to solve online at high frequency. One way to handle nonlinear systems in MPC without having to globally solve the complicated nonconvex optimization problem is to first create a linear response from (a virtual) input to the output of the system [Del Re et al., 1993]. This can be accomplished with inner loop *feedback linearization* [Khalil, 2002] of the form

$$u = \gamma(x, \tilde{u}) \tag{5.1}$$

resulting in a closed loop system which is linear from $\tilde{u}$ to y

$$\dot{z} = Az + B\tilde{u}, \quad y = Cz \tag{5.2}$$

We will outline the details of computing $\gamma(x, \tilde{u})$ in section 5.1.1, but we will first discuss around the main issues with the use of feedback linearization together with MPC.

The first main issue with this is that even if the original cost function (3.34a) is convex the resulting cost function expressed in $\tilde{u}$ can possibly be a nonconvex function. One could simply ignore this complication and formulate a new cost function which is quadratic in $\tilde{u}$. The performance trade off is analyzed in Primbs and Nevistic [1997] with the conclusion that this approximation is justified only when the complete problem can be formulated as a QP.

The second issue, and in our view a much more problematic issue, is that even simple control signal constraints as

$$\underline{u} \leq u_{k+i} \leq \overline{u} \tag{5.3}$$

will transform into a nonlinear and state dependent constraints on $\tilde{u}$ using (5.1) according to

$$\pi(x_{k+i}, \underline{u}) \leq \tilde{u}_{k+i} \leq \pi(x_{k+i}, \overline{u}) \tag{5.4}$$

Several different methods to handle this have been presented in the literature. In, e.g., Deng et al. [2009] the authors calculate the exact input constraints at time k and use them as constraints on the whole prediction horizon, an ad-hoc procedure which clearly does not guarantee recursive feasibility. Other authors such as Margellos and Lygeros [2010], Kothare et al. [1995], Nevistic and Del Re [1994], Kurtz and Henson [2010] propose to use the solution sequence from the previous time step to construct an approximation of the nonlinear constraints. These methods can in general guarantee stability under some strict assumptions, e.g., recursive feasibility. Despite this they can be quite computationally expensive if they work in an iterative manner, e.g., in Margellos and Lygeros [2010] this approximation is done by iteratively solving the linear MPC problem and in each iteration use the previous solution sequence $\{x^*_{k+i}\}_{i=0}^{N}$ to calculate the input constraints using (5.4). The iterations are cancelled when the solution sequence $\{u^*_{k+i}\}_{i=0}^{N-1}$ has converged. The authors of Kurtz and Henson [2010] use a non-iterative approach to construct the approximation of the nonlinear constraints. In the paper the authors propose to use the optimal solution sequence at time $k-1$, $\{u^*_{k-1+i}\}_{i=0}^{N-1}$ to construct a feasible solution at time k as $\{\hat{u}_{k-1+i}\}_{i=0}^{N} = \left\{\{u^*_{k-1+i}\}_{i=0}^{N-1}, 0\right\}$ which is used to predict the future trajectory $\{\hat{x}_{k+i}\}_{i=1}^{N}$ and from this reconstruct the nonlinear state dependent constraints.

In this chapter we will adopt a different approach to handle the nonlinear constraints based on using the exact constraints for the current time step and a set of inner polytope approximations for future time steps.

5.1.1 Feedback linearization

Let us, before we outline the proposed controller, describe the feedback linearization scheme.

If we consider the affine-in-control nonlinear system of the form

$$\dot{x} = f(x) + g(x)u, \quad y = h(x) \tag{5.5}$$

and define the *Lie derivative* in the direction of f as

$$L_f = \sum_{i=1}^{n} f_i(x) \frac{\partial}{\partial x_i}$$

then, if we repeatedly differentiate the output, we obtain $\dot{y} = L_f h$, $\ddot{y} = L_f(L_f h) = L_f^2 h$ etc.

If we assume that the system has $\dim y = \dim u = m$ and apply the Lie derivative then we obtain for the i:th output

$$\begin{aligned} \dot{y}_i &= L_{(f+gu)} h_i = L_{f+u_1 g_1 + u_2 g_2 + \ldots + u_m g_m)} h_i \\ &= L_f h_i + u_1 L_{g_1} h_i + u_2 L g_2 h_i + \ldots + u_m L_{g_m} h_i \end{aligned}$$

If all $L_{g_j} h_i = 0$ this means that $\dot{y}_i = L_f h_i$ and no control signal input affects the output derivative. Continuing the differentiation until one of the $L_{g_j} L_f^{\nu_i} h_i \neq 0$, then u_j affects $y_i^{(\nu_i)}$ and we say that the system has a *relative degree* of ν_i in x_0 [Khalil, 2002].

This procedure can be summarized in a *decoupling matrix* $R(x)$ according to

$$R(x) = \begin{bmatrix} L_{g_1} L_f^{\nu_1 - 1} h_i & \ldots & L_{g_m} L_f^{\nu_1 - 1} h_i \\ \vdots & & \vdots \\ L_{g_1} L_f^{\nu_m - 1} h_m & \ldots & L_{g_m} L_f^{\nu_m - 1} h_m \end{bmatrix} \tag{5.6}$$

which gives

$$\begin{bmatrix} y_1^{(\nu_1)} \\ \vdots \\ y_m^{(\nu_m)} \end{bmatrix} = R(x) \begin{bmatrix} u_1 \\ \vdots \\ u_m \end{bmatrix} + \begin{bmatrix} L_f^{\nu_1} h_1 \\ \vdots \\ L_f^{\nu_m} h_m \end{bmatrix}$$

If $R(x)$ is nonsingular then the control signal can be chosen as

$$u = R^{-1}(x) \left(- \begin{bmatrix} L_f^{\nu_1} h_1 \\ \vdots \\ L_f^{\nu_m} h_m \end{bmatrix} + \tilde{u} \right) \tag{5.7}$$

and the resulting closed loop system will be linear and decoupled from $\tilde{u}$ to y Khalil [2002].

This procedure works well when the zero dynamics is stable or when there is no zero dynamics, i.e., when the relative degree is equal to the state dimension. In these cases the system does not need to be transformed into a controllable canonical form [Khalil, 2002] and the original system states can be kept which is especially good if we have constraints on the states.

In the rest of this paper we restrict our discussion to these kinds of systems that allow us to keep our original states. The aircraft example in Section 5.3.2 motivates this assumption.

5.2 The proposed algorithm

First we make the following assumptions.

5.1 Assumption. The nonlinear system (5.5) is input-output feedback linearizable using the control (5.7) and the linearized system has a discrete-time state-space description

$$x_{k+1} = Ax_k + B\tilde{u}_k \tag{5.8}$$

with no unstable zero dynamics.

5.2 Assumption. The functions $\Psi(\cdot)$ and $\ell(\cdot)$, in (3.34a), are such that they satisfy the necessary conditions of Theorem 3.2, the sets $\mathcal{X}$ and $\mathcal{U}$, in (3.34c) and (3.34d), are convex polytopes and for simplicity we assume $\mathcal{U}$ to be simple bounds on the control signal.

Applying an inner loop feedback linearization as described in the previous section to a nonlinear system of the form (5.5) and then MPC as an outer loop controller, the resulting MPC problem setup is

$$\underset{\tilde{u}}{\text{minimize}}\ \Psi(x_{k+N}) + \sum_{i=0}^{N-1} \ell(x_{k+i}, \tilde{u}_{k+i}) \tag{5.9a}$$

$$
\begin{aligned}
\text{s.t.}& \\
x_{k+i+1} &= Ax_{k+i} + B\tilde{u}_{k+i} && (5.9b)\\
x_{k+i} &\in \mathcal{X} && (5.9c)\\
x_{k+N} &\in \mathcal{T} && (5.9d)\\
\tilde{u}_{k+i} &\in \Pi && (5.9e)
\end{aligned}
$$

where we have defined

$$\Pi = \{\tilde{u}_{k+i} \mid \pi(x_{k+i}, \underline{u}) \leq \tilde{u}_{k+i} \leq \pi(x_{k+i}, \overline{u})\}$$

where the functions $\pi(\,\cdot\,)$ are the nonlinear constraints that arise from the feedback linearization (5.7).

5.2.1 Nonlinear constraint approximations

To begin with, since the current state, x_k, is known, the exact nonlinear constraint on $\tilde{u}_k$ can be calculated as

$$\pi(x_k, \underline{u}) \leq \tilde{u}_k \leq \pi(x_k, \overline{u})$$

Obviously, since this is a linear constraint in $\tilde{u}_k$, our scheme should be able to use this exactly without resorting to any conservative approximation. It is thus our goal to derive an algorithm where this constraint is used exactly, and the constraints on future control signals are included in an as non-conservative fashion as possible, while guaranteeing stability and feasibility.

A first step to handle the future time steps of the nonlinear constraint (5.9e) is to simply replace it with a global inner convex polytopic approximation

$$\mathcal{G} = \{(x, \tilde{u}) \mid x \in \mathcal{X},\ g_l(\mathcal{X}) \leq \tilde{u} \leq g_u(\mathcal{X})\} \tag{5.10}$$

where $g_u(\mathcal{X})$ is a concave piecewise affine function such that $g_u(\mathcal{X}) \leq \pi(\mathcal{X}, \overline{u})$ and $g_l(\mathcal{X})$ is a convex piecewise affine function such that $g_l(\mathcal{X}) \geq \pi(\mathcal{X}, \underline{u})$. An example of an inner approximation, $\mathcal{G}$, is shown in Figure 5.1. Note that this approximation is not unique and the degree of suboptimality vary with the method of approximation.

If the nonlinear constraints (5.9e) form a highly nonconvex set, then it is fair to assume that $\mathcal{G}$ poorly approximates the true nonlinear constraints over the entire state space, i.e., it can only be close to the true constraints in some, possibly small, regions, cutting of control authority in other regions. An example of this is shown in Figure 5.2. This motivates us to not use a global approximation for all time steps in the control signal sequence.

If one makes use of the fact that the true constraints are known at time k it is easy to calculate the bounded evolution of the system to time $k+1$ and therefore all possible states, $\mathcal{X}_{k+1}$. It is then obvious that for this limited subset of the state-space there might exist a better inner convex approximation of the nonlinear constraints than the global approximation $\mathcal{G}$. Hence, we would like to construct a convex polytope, $\mathcal{I}$, over the set $\mathcal{X}_{k+1}$ and constrain $(x_{k+1}, \tilde{u}_{k+1})$ to this local approximation.

This procedure can of course be repeated for time step $k+2, k+3, \ldots, k+N-1$, generating a new local polytope for each $(x_{k+i}, \tilde{u}_{k+i})$. A significant problem will however occur if one tries to prove recursive feasibility of this scheme. Since we always use the exact constraint for the first control input, this conflicts with our inner approximation which was used for future control input, when we shift the horizon in standard MPC stability and recursive feasibility proofs. If we use the full control authority in the next time instant, the state predictions arising from that set will move outside the predictions that were used in the previous time instant when predictions were based on an inner approximation of the control input at $k+1$. To account for this, a scheme based on both inner approximations of control inputs for actual control decisions, and outer approximations of control inputs to perform the propagation of states, will be used.

The local constraint approximations are constructed as inner convex approximations of the nonlinear constraints based on reachable sets.

5.3 Definition. At time k, the outer approximation of the i:th step reachable set $\mathcal{X}_{k+i}$ is recursively defined as

$$\mathcal{X}_{k+i} = A\mathcal{X}_{k+i-1} + B\mathcal{C}_{k+i-1}$$

where

$$\mathcal{X}_k = \{x_k\}$$

The set $\mathcal{C}_{k+i}$ is an outer polytopic approximation of the nonlinear control constraints in the reachable set $\mathcal{X}_{k+i}$, i.e.,

$$\mathcal{C}_{k+i} = \left\{ \tilde{u}_{k+i} \mid \omega_l^{k+i}(\mathcal{X}_{k+i}) \leq \tilde{u}_{k+i} \leq \omega_u^{k+i}(\mathcal{X}_{k+i}) \right\}$$

where $\omega_u^{k+i}(\cdot)$ is a concave piecewise affine function such that

$$\omega_u^{k+i}(\mathcal{X}_{k+1}) \geq \pi(\mathcal{X}_{k+1}, \overline{u})$$

and $\omega_l^{k+i}(\cdot)$ is a convex piecewise affine function such that

$$\pi(\mathcal{X}_{k+1}, \underline{u}) \geq \omega_l^{k+i}(\mathcal{X}_{k+1})$$

The initial outer approximation, $\mathcal{C}_k$, is the exact control constraints, i.e.,

$$\mathcal{C}_k = \{u_k \mid \pi(x_k, \underline{u}) \leq \tilde{u}_k \leq \pi(x_k, \overline{u})\}$$

It should be noted here that we do not specify how to construct the sets $\mathcal{C}_{k+i}$, just constraints on how they can be constructed. The user is free to chose any method he or she may find suitable and the stability and feasibility of the algorithm holds regardless of the chosen method.

From the i:th step reachable set the local convex approximation, $\mathcal{I}_i^k$, i step ahead at time k can now be constructed as the polytope defined from the constraints

$$h_l^{k+i}(\mathcal{X}_{k+i} \cap \mathcal{X}) \leq \tilde{u}_{k+i} \leq h_u^{k+i}(\mathcal{X}_{k+i} \cap \mathcal{X})$$
$$x_{k+i} \in \mathcal{X}_{k+i} \cap \mathcal{X}$$

where $h_u^{k+i}(\,\cdot\,)$ is a concave piecewise affine function such

$$g_u(\mathcal{X}_{k+1}) \leq h_u^{k+i}(\mathcal{X}_{k+1}) \leq \pi(\mathcal{X}_{k+1}, \overline{u})$$

and $h_l^{k+i}(\,\cdot\,)$ is a convex piecewise affine function such

$$\pi(\mathcal{X}_{k+1}, \underline{u}) \leq h_l^{k+i}(\mathcal{X}_{k+1}) \leq g_l(\mathcal{X}_{k+1})$$

In other words, the local polytope, $\mathcal{I}_i^k$, shall be an inner approximation to the nonlinear constraints and on the subset $\mathcal{X}_{k+i}$ it shall hold that $\mathcal{G} \subseteq \mathcal{I}_i^k$, which can always be achieved.

Figure 5.1 shows an example that illustrates the relationship between the local polytopes, the global polytope and the nonlinear constraints. Note that as for the global inner convex approximation this construction is non unique, in this thesis we have used a tangent plane for the concave surfaces and a piecewise affine approximation of the convex surfaces (described further in the examples).

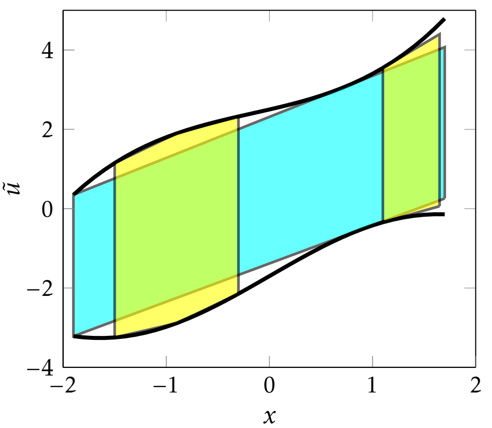

Figure 5.1: *Example showing the nonlinear constraints on $\tilde{u}$ as upper and lower bound, the global approximation, $\mathcal{G}$, in cyan (dark shaded) and two local approximations, $\mathcal{I}_1^k$, for different x_k in yellow.*

5.2.2 MPC receding horizon setup

Now let us summarize the discussion above into our proposed MPC algorithm. At each sample time k solve the NMPC problem given by Algorithm 2.

The state and control signals are constrained to the local polytopes up to horizon $N_l \leq N - 1$ and constrained to the global polytope for $N_l < i < N$. The introduction of the horizon N_l is to highlight that, depending on the problem, there might

Algorithm 2 Approximate NMPC algorithm

1: given measurement x_k, and a set of approximations, $\mathcal{I}_i^k$ and $\mathcal{G}$
2: Solve the QP

$$\underset{\tilde{u}}{\text{minimize}}\ \Psi(x_{k+N}) + \sum_{i=0}^{N-1} \ell(x_{k+i}, \tilde{u}_{k+i}) \tag{5.11a}$$

s.t.

$$x_{k+i+1} = Ax_{k+i} + B\tilde{u}_{k+i} \quad \forall\ i = 0, \dots, N-1 \tag{5.11b}$$

$$\pi(x_k, \underline{u}) \leq \tilde{u}_k \leq \pi(x_k, \overline{u}) \tag{5.11c}$$

$$(x_{k+i}, \tilde{u}_{k+i}) \in \mathcal{I}_i^k \ \ \forall\ i = 1, \dots, N_l \tag{5.11d}$$

$$(x_{k+i}, \tilde{u}_{k+i}) \in \mathcal{G} \ \ \forall\ i = N_l + 1, \dots, N-1 \tag{5.11e}$$

$$x_{k+N} \in \mathcal{T} \tag{5.11f}$$

3: calculate the control signal as $u_k = \gamma(x, \tilde{u}(1))$ from (5.7)
4: update the local approximations $\mathcal{I}_i^{k+1}$ as

$$\mathcal{I}_i^{k+1} = \mathcal{I}_{i+1}^k \ \ \forall\ i = 1, \dots, N_l - 1 \tag{5.12}$$

5: construct a new set $\mathcal{I}_{N_l}^{k+1}$ from the procedure in Section 5.2.1
6: repeat from 1.

not be any performance gain in using the local polytopes for the entire horizon, $N-1$, so instead the fixed global inner approximation is used for the last part of the horizon. We will see an example of this in the next section. The final constraint set $\mathcal{T}$ is an invariant set as defined in Definition 3.1 and it is calculated using the global inner polytope approximation, $\mathcal{G}$, as the initial bounds on x and $\tilde{u}$.

The practical advantage of this approach of shifting the local polytopes in time, as in (5.12), is that only one local approximation has to be calculated in each iteration. The theoretical advantage is that this also guarantees recursive feasibility.

We can now state the main stability result for this controller.

5.4 Theorem. *For any initially feasible state x_0, the MPC controller defined by Algorithm 2 remains feasible and stabilizes the system (5.5).*

Proof: To show recursive feasibility, let us denote the set of states where (5.11) is feasible with $\mathcal{F}$. Assume that $x_k \in \mathcal{F}$ and that (5.11) has the optimal solution sequence $\{\tilde{u}_{k+i}^*\}_{i=0}^{N-1}$.

We now claim that a feasible solution at time $k+1$ is to use the control sequence $\{\hat{u}_{k+i}^*\}_{i=1}^{N} = \left\{\{\tilde{u}_{k+i}^*\}_{i=1}^{N-1}, \kappa(x_N^*)\right\}$ where $\kappa(x)$ is a local controller as defined in Section 3.2.1.

To see that this is a feasible solution we first note that since $x_{k+N} \in \mathcal{T}$ we can select $\hat{u}_{k+N} = \kappa(x_N^*)$ since this will ensure that $x_{k+N+1} \in \mathcal{T}$ and all constraints are satisfied at $k+N+1$. Also we note that since $\tilde{u}_{k+1}^* \in \mathcal{I}_1^k \subset \Pi$ this means that $\pi(x_{k+1}, \underline{u}) \leq \tilde{u}_{k+1}^* \leq \pi(x_{k+1}, \overline{u})$ is feasible at time $k+1$.

Furthermore we have that all $\tilde{u}_{k+i}^*,\ i = 2, \ldots, N_l$ are feasible at time $k+1$ since the local approximations are shifted one time step (5.12). The control $\tilde{u}_{k+N_l+1}^* \in \mathcal{G}$ at time k are also feasible at time $k+1$ since $\tilde{u}_{k+N_l+1}^* \in \mathcal{I}_{N_l}^{k+1} \supseteq \mathcal{G}$ at time $k+1$ and $\mathcal{I}_{N_l}^{k+1}$ is always a nonempty set if the problem is initially feasible. All other $\tilde{u}_{k+N_l+i}^* \in \mathcal{G}$ are trivially feasible.

Convergence of the proposed algorithm is not affected by the local approximations and the standard proof from Section 3.2.1 hold without any change. This means that the controller defined through Algorithm 2 stabilizes the system (5.8) and by the Assumption 5.1 it also stabilizes the nonlinear system (5.5). □

5.3 Examples

In this section we present two examples to illustrate the properties of the proposed algorithm. In the first example we consider a fictitious nonlinear system whose purpose is to illustrate the generation and propagation of the local polytopes.

In the second example we consider the task of controlling a fighter aircraft which has nonlinear unstable dynamics. The purpose of this example is to illustrate the degree of suboptimality for the proposed method.

The implementation and simulation has been performed in Matlab with YALMIP [Löfberg, 2004] and MPT [Herceg et al., 2013].

5.3.1 Illustrative scalar example

Consider a nonlinear system given by

$$\begin{aligned} \dot{x} &= 1.8x + \left(0.2x^4 + 0.875\right) u \\ y &= x \end{aligned}$$

with the constraints $-2 \leq x \leq 2$, and $-2 \leq u \leq 2$. Following the procedure in Section 5.1.1 we obtain the feedback linearization control law

$$u = \frac{1}{0.2x^4 + 0.875} \left(\tilde{u} - 1.8x\right)$$

and the resulting linear system is an integrator from $\tilde{u}$ to y.

The nonlinear feedback gives the following nonlinear control constraints on $\tilde{u}$

$$-0.4x^4 + 1.8x - 1.75 \leq \tilde{u} \leq 0.4x^4 + 1.8x + 1.75 \tag{5.13}$$

shown in Figure 5.2 together with the global approximation $\mathcal{G}$. Note the massive loss of control authority at, e.g., $x_k = \pm 2$, if only the global approximation $\mathcal{G}$ is used to constrain $\tilde{u}$.

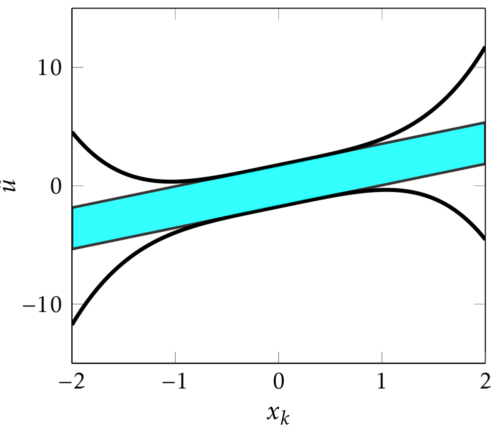

Figure 5.2: *Nonlinear constraints on $\tilde{u}$ due to feedback linearization and a global inner approximation $\mathcal{G}$.*

Algorithm 2 is applied to the discrete-time version of the integrator system with sample time 0.4s. Using $N = 5$ and $N_l = 4$, i.e., we use the global polytope $\mathcal{G}$ only to calculate the terminal constraint set, $\mathcal{T}$, and the objective is to control the system to the origin. The local polytopes are calculated from the tangent line at the center point in the reachable set.

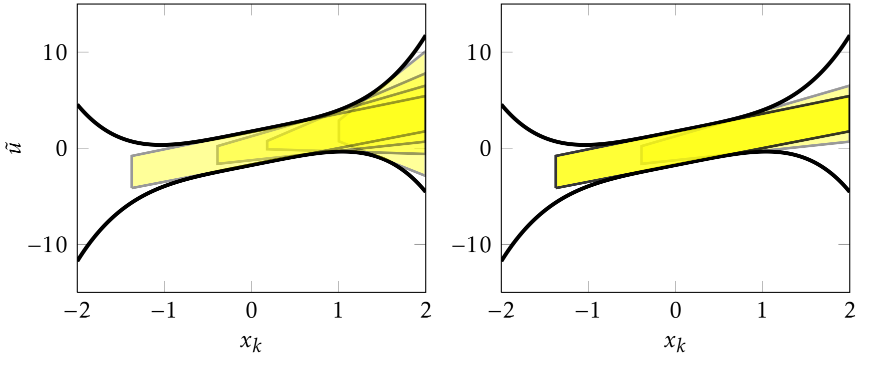

Figure 5.3: *The left axis shows the nonlinear constraints on $\tilde{u}$ and the local polytopes $\mathcal{I}_i^0$ for $i = 1, \ldots, 4$ at time $k = 0$. The right axis shows the local polytopes $\mathcal{I}_i^2$ at time $k = 2$.*

Starting in $x = 1.9$ the generated local polytopes at time $k = 0$, $\mathcal{I}_i^0$, are shown in the left part Figure 5.3. It clearly demonstrates the increased control signal ability compared to only using the global approximation as in Figure 5.2.

At the next time step the first local approximation, $\mathcal{I}_1^0$, is discarded. All other polytopes are shifted one step, i.e., $\mathcal{I}_1^1 = \mathcal{I}_2^0$, $\mathcal{I}_2^1 = \mathcal{I}_3^0$, etc. A new local polytope,

$\mathcal{I}_4^1$, is generated at the end of the sequence. The right part of Figure 5.3 shows the local approximations at time $k = 2$, when $x_k \approx 0.6$. In the figure one can see that the polytope $\mathcal{I}_3^0$ has been shifted and is now, at time $k = 2$, the polytope $\mathcal{I}_1^2$. The same holds for $\mathcal{I}_4^0 = \mathcal{I}_2^2$.

5.3.2 Nonlinear aircraft example

In this example we consider a continuous-time nonlinear model of the same aircraft as in Section 4.3.1. The nonlinearity consist of an extra term in the moment equation (5.14b) which is proportional to the square of the angle of attack.

$$\dot{\alpha} = -1.8151\alpha + 0.9605q \tag{5.14a}$$

$$\dot{q} = 0.15\alpha^2 + 12.9708\alpha - 1.8988q + 19.8474\delta_e \tag{5.14b}$$

$$y = \alpha \tag{5.14c}$$

The states are the same as in Section 4.3.1 where q is the angular rate in pitch and α the angle between the aircraft x-axis and the velocity vector, see Figure 1.1.

The coefficients of the linear terms have been selected to correspond to the linearized dynamics of the ADMIRE model at Mach 0.6 and altitude 1000 m, for details see Forssell and Nilsson [2005]. The coefficient for the α^2-term is selected to make the α-contribution to the moment equation in the nonlinear model approximately 15% larger at $\alpha = 30°$ compared to the linearized model. The constraints on the system are basic control surface deflection limits $|\delta_e| \leq 25°$ and also limits on the angle of attack $-10° \leq \alpha \leq 30°$.

For the system (5.14) it is easy to see that by selecting the nonlinear feedback as

$$\delta_e = \tilde{u} - 0.0076\alpha^2 \tag{5.15}$$

the closed loop system from MPC control input, $\tilde{u}$, to the output, α will be linear.

$$\dot{\alpha} = -1.8151\alpha + 0.9605q \tag{5.16a}$$

$$\dot{q} = 12.9708\alpha - 1.8988q + 19.8474\tilde{u} \tag{5.16b}$$

We can now formulate a MPC problem for the system (5.16) on the form (5.11) where we chose to use a cost function that is quadratic in $\tilde{u}$ since the goal is to end up in a standard QP problem. In the cost function we used the tuning parameters

$$Q = \begin{bmatrix} 5 & 0 \\ 0 & 1 \end{bmatrix}, \quad R = 2$$

and the sample time is 1/60 second.

Note that the state constraints are still linear after the feedback linearization but the control constraints are now nonlinear and state dependent.

$$-25° + 0.0076\alpha_{k+i}^2 \leq \tilde{u}_{k+i} \leq 25° + 0.0076\alpha_{k+i}^2 \tag{5.17}$$

In this case the nonlinearities are mild and the lower bound is convex while the upper bound is concave. It is therefore quite easy to make a good inner convex polytope approximation, $\mathcal{G}$, of these constraints, see Figure 5.4.

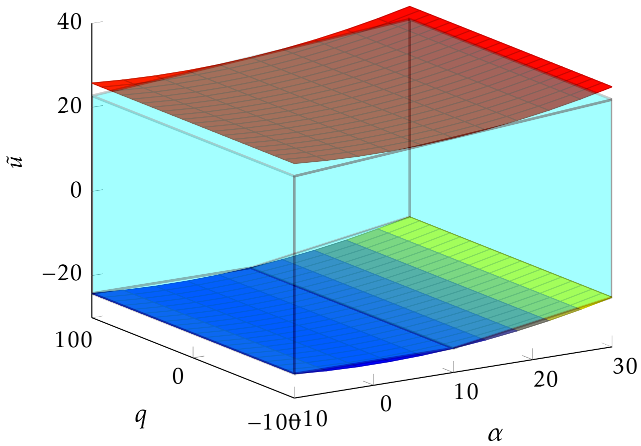

Figure 5.4: *Nonlinear control signal constraints* (5.17) *and a global inner convex polytope approximation,* $\mathcal{G}$

The lower bound can be approximated arbitrarily well with increasing complexity of the polytope while at the upper bound we cut away control performance around the α-limits with the global approximation. This opens up for the possibility that control performance could be gained using local approximations of the constraints when the current state is close to the α-limits.

If we compare Algorithm 2, with $N_l = 1$, with a global nonlinear solver (the standard branch and bound solver in YALMIP) for the problem (5.9), we obtain a measure of the performance loss, i.e., the suboptimality of our algorithm compared to using the exact nonlinear constraints. We have compared the open loop optimal cost of the two algorithms and as performance measure we use the relative error

$$\eta = \frac{\left|J^* - \hat{J}\right|}{|J^*|}$$

between the optimal cost of the proposed algorithm, $\hat{J}$, and the branch and bound solver, J^*. Figure 5.5 shows the variation of the relative error over the entire feasible state space.

From the figure we can conclude that for this example the maximum performance loss is approximately 4% and that it occurs in areas of the state space where maximum control signals are likely to be used, i.e., large (absolute value) combinations of angles and angular rates.

In the example we use only one step local approximations and a fair question

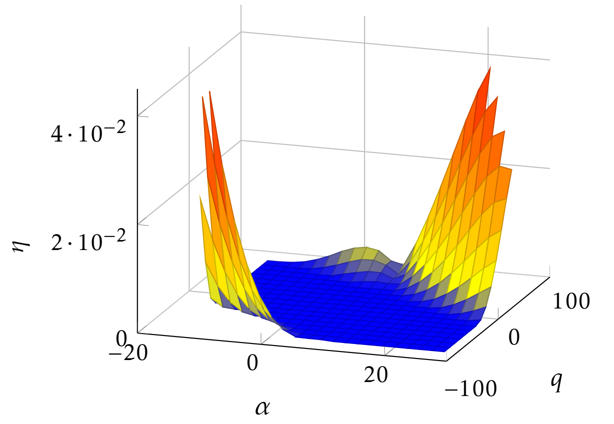

Figure 5.5: *Variation of the relative error, η, between proposed algorithm and global solver over the state space.*

to ask is if any more performance could be gained by increasing the number of local approximations. In Table 5.1 we compare, for a subset of the state space, the relative error for both $N_l = 1$ and $N_l = 10$.

Table 5.1: *A comparison of performance loss, η, when the number of local polytopes are increased from $N_l = 1$ to $N_l = 10$.*

x_0	$\begin{pmatrix}-14.0\\-26.5\end{pmatrix}$	$\begin{pmatrix}-5.0\\-9.5\end{pmatrix}$	$\begin{pmatrix}15.0\\28.3\end{pmatrix}$	$\begin{pmatrix}28.0\\52.9\end{pmatrix}$
$N_l = 1$	$4.1 \cdot 10^{-4}$	$3.9 \cdot 10^{-14}$	$7.7 \cdot 10^{-4}$	$2.5 \cdot 10^{-2}$
$N_l = 10$	$3.5 \cdot 10^{-11}$	$8.2 \cdot 10^{-16}$	$3.5 \cdot 10^{-5}$	$1.1 \cdot 10^{-2}$

The results in Table 5.1 indicate at least a 50% reduction in the relative error of the optimal cost when the number of local polytopes are increased. Although it is a significant decrease in the relative error the absolute values are in both cases relatively small and it is questionable if it gives any practical performance gain.

To evaluate the practical implications of the difference between the proposed controller and the branch and bound method we compare the time response of the closed loop system for the both controllers. We initiate the system at a steady state where it is expected to have a relatively large difference in optimal cost between the two controllers. However in Figure 5.6 we see that the time response of both closed loop systems are identical to the naked eye.

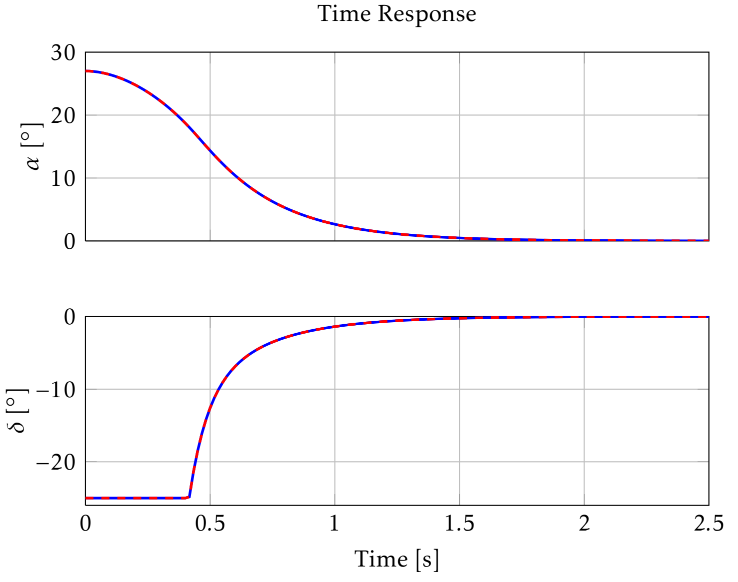

Figure 5.6: *Time response of the proposed controller (blue solid line) and the branch and bound controller (red dashed line) for the initial condition* $x_0 = [27 \quad 51]^T$*. The two responses appear identical.*

Since there appear to be no difference in the time response we also look at the error between the two time responses. Figure 5.7 reveal that there are very tiny differences in both state and control signals between the two closed loop systems. We can conclude that when there are only small nonlinearities, there is no practical gain in increasing the number of local polytopes, i.e., the important property is that we can use the full control authority for the current control input.

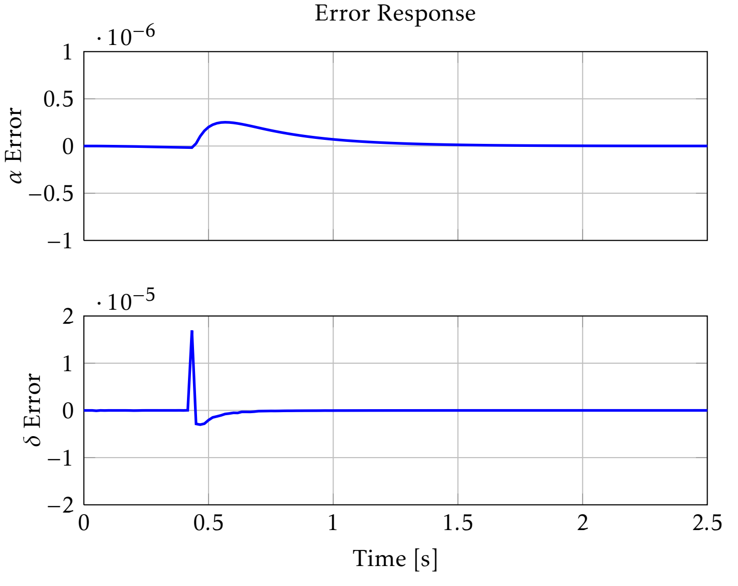

Figure 5.7: *Time response of the difference in state and control between the two controllers.*

6

Conclusions and Future Work

The overall objective of this thesis was to investigate Model Predictive Control and its applicability to the aeronautical industry. Properties such as stability, robustness, simplicity and verifiability are of special importance to the industry and should be considered when reviewing existing algorithms and developing new ones.

It has not been the focus of this thesis to survey all existing theory in the field and to make conclusive statements whether Model Predictive Control is a suitable choice for flight control design, but rather to investigate a few fundamental properties that are important and to develop new suitable algorithms.

In Chapter 4 we study the topic of reference tracking. Although reference tracking is a relatively mature topic in MPC, very little has been published regarding the stability properties when the terminal constraint set depends on the current reference, especially stability when the reference approaches the border of the feasible set. We can only speculate why it is so, but perhaps the traditional application areas have not motivated any more in depth analysis of this. The successful research that has been performed within this area is based on lifting the problem of calculating the terminal constraint set into a higher dimension in which the terminal constraint set is constant for an arbitrary reference. The main drawback with these algorithms is increased complexity of the resulting optimization problem.

In this thesis we instead make a simple extension to the standard dual mode MPC algoritm to allow for tracking arbitrary setpoints that approaches the boundary of the feasible set. This allows for, e.g., maneuver load limiting in fighter aircraft design. We prove that by scaling the terminal state constraint set with a positive scalar λ such that it is a subset of the feasible set for arbitrary references, both stability and recursive feasibility can be guaranteed. The performance and robustness of the proposed algorithm is comparable to the existing methods while the complexity of the proposed controller is significantly reduced. The complexity reduction partly comes from the fact that we utilize duality theory to rewrite complex set of constraints into a much simpler form and partly from the fact that the terminal constraint set is in a lower dimension.

The method developed in Chapter 4 is for linear models but in real life aircraft and helicopters do not have linear dynamics. The aerodynamic coefficients, see Section 1.1, are normally nonlinear functions that varies also with altitude and speed. In Chapter 5 we tackled the problem of nonlinear dynamics while trying to avoid complicated nonconvex optimization. We developed a new method of model predictive control for nonlinear systems based on feedback linearization and local convex approximations of the control constraints. We have shown that the proposed method can guarantee recursive feasibility and convergence to the origin. An example from the aircraft industry show that good performance can be attained and that the loss of optimality can be small with reasonable simple computational efforts.

There still remain numerous of things that are open topics for research which are important for the use of MPC in aeronautical applications. These concerns areas such as robustness to model errors and disturbances, controller structure and varying system dynamics.

Traditional robustness in MPC is concerned with guaranteeing constraint satisfaction despite disturbances and model errors. This normally result in a conservative constraint satisfaction which is not desirable, e.g., in the aircraft example when we want to track a reference out to the border of the feasible set. Instead constraint satisfaction from a statistical viewpoint might serve the purposes of the aeronautical industry better. Here integral control and the use of slack variables are crucial and their stability properties need to be further investigated.

Another topic that is interesting for research is the varying dynamics of aircraft and helicopters. The varying dynamics can be modeled with the use of LPV (Linear Parameter Varying) models. MPC has been applied to LPV models both in explicit form and as online optimization but since the terminal constraint set depends on the dynamics of the system it will in this setup be a parameter varying set and hence standard stability arguments do not hold. In fact guaranteed stability of reference tracking MPC with LPV models still remains an open issue.

In the introductory chapter we argued that the LQ design technique is widely used in the aeronautical industry and that adding constraints to this framework is a complicating factor. Therefore it is appealing to view the constraint satisfaction as an *add-on* feature to an existing LQ design. This can be done in the framework of *reference governors* or *command governors*. Investigating such different MPC structures, their stability and robustness properties and explicit formulations would really increase the applicability of MPC to aeronautical applications in the near future.

Bibliography

J. Åkesson and P. Hagander. Integral Action - a Disturbance Observer Approach. In *Proceedings of the 2003 European Control Conference*, 2003. Cited on page 34.

F. Almeida and D. Leissling. Fault-Tolerant Flight Control System Using Model Predictive Control. In *2009 Brazilian Symposium on Aerospace Eng. & Applications*, 2009. Cited on page 5.

D. Avis and K. Fukuda. A pivoting algorithm for convex hulls and vertex enumeration of arrangements and polyhedra. *Discrete & Computational Geometry*, 8 (1):295–313, 1992. Cited on page 19.

M. Baotic. *Optimal Control of Piecewise Affine Systems – a Multi-parametric Approach –*. Dissertation, Swiss Federal Institute of Technology, 2005. Cited on pages 19 and 41.

A. Bemporad, S. Member, A. Casavola, and E. Mosca. Nonlinear Control of Constrained Linear Systems via Predictive Reference Management. *IEEE Transactions on Automatic Control*, 42(3):340–349, 1997. Cited on page 49.

A. Bemporad, F. Borrelli, and M. Morari. Model predictive control based on linear programming˜the explicit solution. *IEEE Transactions on Automatic Control*, 47(12):1974–1985, Dec. 2002a. Cited on page 39.

A. Bemporad, M. Morari, V. Dua, and E. N. Pistikopoulos. The explicit linear quadratic regulator for constrained systems. *Automatica*, 38(1):3–20, Jan. 2002b. ISSN 00051098. Cited on pages 39, 41, and 42.

A. Ben-Tal and A. Nemirovski. Robust optimization methodology and applications. *Mathematical Programming*, 92(3):453–480, May 2002. Cited on page 54.

R. R. Bitmead, M. Gevers, and V. Wertz. Adaptive optimal control: the thinking man's GPC, 1990. Cited on page 27.

F. Blanchini. Set invariance in control. *Automatica*, 35:1747 – 1767, 1999. Cited on pages 28 and 57.

S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 1 edition, 2004. Cited on pages 9, 16, 21, 44, and 54.

M. Cannon. Efficient nonlinear model predictive control algorithms. *Annual Reviews in Control*, 28(2):229–237, Jan. 2004. Cited on page 44.

H. Chen and F. Allgöwer. A Quasi-Infinite Horizon Nonlinear Model Predictive Control Scheme with Guaranteed Stability. *Automatica*, 34(10):1205–1217, 1998. Cited on page 44.

L. Chisci and G. Zappa. Dual mode predictive tracking of piecewise constant references for constrained linear systems. *International Journal of Control*, 76 (1):61–72, Jan. 2003. Cited on pages 49 and 50.

M. de Berg, O. Cheong, M. van Kreveld, and M. Overmars. *Computational Geometry, Algorithms and Applications*. Springer London, third edit edition, 2008. ISBN 9783540779735. Cited on pages 18 and 19.

N. M. C. de Oliveira and L. T. Biegler. Constraint handing and stability properties of model-predictive control. *AIChE Journal*, 40(7):1138–1155, 1994. Cited on page 38.

S. L. De Oliveira Kothare and M. Morari. Contractive Model Predictive Control for Constrained Nonlinear Systems. *IEEE Transactions on Automatic Control*, 45(6):1053–1071, 2000. Cited on page 44.

L. Del Re, J. Chapuis, and V. Nevistic. Predictive Control with Embedded Feedback Linearization for Bilinear Plants with Input Constraints. In *Proceedings of the 32th IEEE Conference on Decision and Control (CDC)*, pages 2984–2989, 1993. Cited on page 72.

J. Deng, V. M. Becerra, and R. Stobart. Input Constraints Handling in an MPC/Feedback Linearization Scheme. *International Journal of Applied Mathematics and Computer Science*, 19(2):219–232, June 2009. Cited on page 72.

V. Dua and E. N. Pistikopoulos. An Algorithm for the Solution of Multiparametric Mixed Integer Linear Programming Problems. *Annals of Operations Research*, 99(i):123–139, 2001. Cited on page 41.

W. B. Dunbar, M. B. Milam, R. Franz, and R. M. Murray. Model Predictive Control Of A Thrust-Vectored Flight Control Experiment. In *IFAC World Congress*, 2002. Cited on page 5.

A. Ferramosca, D. Limon, I. Alvarado, T. Alamo, and E. Camacho. MPC for tracking with optimal closed-loop performance. *Automatica*, 45(8):1975–1978, Aug. 2009. Cited on pages 49, 57, and 61.

A. Ferramosca, D. Limon, I. Alvarado, T. Alamo, F. Castaño, and E. Camacho. Optimal MPC for tracking of constrained linear systems. *International Journal of Systems Science*, 42(8):1265–1276, Aug. 2011. Cited on pages 60 and 61.

R. Fletcher. *Practical Methods of Optimization*. John Wiley & Sons, 2nd edition, 1987. Cited on page 38.

F. A. Fontes. A general framework to design stabilizing nonlinear model predictive controllers. *Systems & Control Letters*, 42(2):127–143, Feb. 2001. Cited on page 44.

L. Forssell and U. Nilsson. ADMIRE The Aero-Data Model In a Research Environment Version 4.0, Model Description. Technical Report December, Swedish Defence Research Agency, 2005. Cited on pages 61 and 81.

T. Gal. A " Historiogramme " of Parametric Programming. *The Journal of the Operational Research Society*, 31(5):449–451, 1980. Cited on page 39.

B. A. G. Genuit, L. Lu, and W. P. M. H. Heemels. Approximation of PWA Control Laws Using Regular Partitions : An ISS Approach. In *Proceedings of the 18th IFAC World Congress*, pages 4540–4545, 2011. Cited on page 42.

A. H. Glattfelder and W. Schaufelberger. *Control systems with input and output constraints*. Springer Verlag, 2003. Cited on page 5.

S. Gros, R. Quirynen, and M. Diehl. Aircraft control based on fast non-linear MPC & multiple-shooting. In *2012 IEEE 51st IEEE Conference on Decision and Control (CDC)*, number 1, pages 1142–1147. Ieee, Dec. 2012. Cited on page 5.

B. Grünbaum. *Convex Polytopes*. Springer-Verlag, 1 edition, 2003. ISBN 0-387-00424-6. Cited on pages 18 and 53.

M. A. Henson. Nonlinear model predictive control: current status and future directions. *Computers & Chemical Engineering*, 23(2):187–202, Dec. 1998. Cited on page 44.

M. Herceg, M. Kvasnica, C. N. Jones, and M. Morari. Multi-Parametric Toolbox 3.0. In *Proceedings of the 2013 European Control Conference*, pages 502–510, 2013. Cited on pages 62 and 79.

A. Jadbabaie, J. Yu, and J. Hauser. Unconstrained receding-horizon control of nonlinear systems. *IEEE Transactions on Automatic Control*, 46(5):776–783, May 2001. Cited on page 44.

C. N. Jones, P. Grieder, and S. Raković. A logarithmic-time solution to the point location problem for parametric linear programming. *Automatica*, 42(12):2215–2218, Dec. 2006. Cited on page 42.

M. Kale and A. Chipperfield. Stabilized MPC formulations for robust reconfigurable flight control. *Control Engineering Practice*, 13(6):771–788, June 2005. Cited on page 5.

R. E. Kalman. Contributions to the Theory of Optimal Control. *Boletin de la Sociedad Matematica Mexicana*, 5:102–119, 1960. Cited on page 4.

S. S. Keerthi and E. G. Gilbert. Optimal Infinite-Horizon Feedback Laws for a General Class of Constrained Discrete-Time Systems : Stability and Moving-Horizon Approximations. *Journal of Otimization Theory and Applications*, 57 (2):265–293, 1988. Cited on page 27.

E. C. Kerrigan and J. M. Maciejowski. Soft Contraints and Exact Penalty Functions in Model Predictive Control. In *Proceedings of the UKACC International Conference*, 2000. Cited on page 38.

T. Keviczky and G. Balas. Receding horizon control of an F-16 aircraft: A comparative study. *Control Engineering Practice*, 14(9):1023–1033, Sept. 2006. Cited on page 5.

H. K. Khalil. *Nonlinear Systems third edition*. Prentice Hall, 2002. ISBN 0-13-067389-7. Cited on pages 26, 72, 73, and 74.

M. V. Kothare, V. Nevistic, and M. Morari. Robust Constrained Model Predictive Control for nonlinear Systems: A Comperative Study. In *Proceedings of the 34th IEEE Conference on Decision and Control (CDC)*, number December, pages 2884–2885, 1995. Cited on page 72.

M. J. Kurtz and M. A. Henson. Feedback linearizing control of discrete-time nonlinear systems with input constraints. *International Journal of Control*, 70:4: 603–616, 2010. Cited on pages 72 and 73.

M. Kvasnica. *Real-Time Model Predictive Control via Multi-Parametric Programming: Theory and Tools*. VDM Verlag, 2009. ISBN 9783639206449. Cited on page 42.

J. B. Lasserre. Global Optimization with Polynomials and the Problem of Moments. *SIAM Journal on Optimization*, 11:796–817, 2001. Cited on page 18.

H. J. Lee and B. Cooley. Recent advances in model preedictive control and other related areas. In *5th International conference on chemical process control*, pages 201–216, 1997. Cited on page 32.

D. Limon, I. Alvarado, T. Alamo, and E. Camacho. MPC for tracking piecewise constant references for constrained linear systems. *Automatica*, 44(9):2382–2387, Sept. 2008. Cited on pages 49, 50, 61, 63, and 67.

J. Löfberg. YALMIP: A Toolbox for Modeling and Optimization in MATLAB. In *Proceedings of the CACSD Conference*, Taipei, Taiwan, 2004. Cited on pages 62 and 79.

J. M. Maciejowski. *Predictive Control with Constraints*. Prentice Hall, 1 edition, 2002. ISBN 0-201-39823-0. Cited on pages 5, 37, and 38.

J. M. Maciejowski and C. N. Jones. MPC Fault-tolerant Flight Control Case Study: Flight 1862. In *IFAC Safeprocess Conference*, 2003. Cited on page 5.

U. Maeder and M. Morari. Offset-free reference tracking with model predictive control. *Automatica*, 46(9):1469–1476, Sept. 2010. Cited on page 33.

L. Magni, G. D. Nicolao, L. Magnani, and R. Scattolini. A stabilizing model-based predictive control algorithm for nonlinear systems. *Automatica*, 37:1351–1362, 2001. Cited on page 44.

K. Margellos and J. Lygeros. A simulation based MPC technique for feedback linearizable systems with input constraints. In *49th IEEE Conference on Decision and Control (CDC)*, pages 7539–7544. Ieee, Dec. 2010. Cited on page 72.

D. Mayne. Nonlinear Model Predictive Control: Challenges and Opportunities. In *Nonlinear model predictive control*, pages 23–44. Birkhäuser Verlag Basel, 2000. Cited on page 44.

D. Mayne and H. Michalska. Receding horizon control of nonlinear systems. *IEEE Transactions on Automatic Control*, 35(7):814–824, July 1990. Cited on page 44.

D. Mayne, J. B. Rawlings, C. V. Rao, and P. Scokaert. Constrained model predictive control: Stability and optimality. *Automatica*, 36(6):789–814, June 2000. ISSN 00051098. Cited on pages 27, 28, 32, 39, and 57.

E. S. Meadows and T. Badgwell. Feedback through steady-state target optimization for nonlinear model predictive control. *Journal of Vibration and Control*, 4:61–74, 1998. Cited on pages 31, 32, and 33.

B. Mettler. *Identification Modeling and Characteristics of Miniature Rotorcraft*. Springer, 2002. ISBN 9781402072284. Cited on page 66.

H. Michalska. A New Formulation of Receding Horizon Stabilising Control without Terminal Constraint on the State. *European Journal of Control*, 3:2–14, 1997. Cited on page 44.

H. Michalska and D. Mayne. Robust receding horizon control of constrained nonlinear systems. *IEEE Transactions on Automatic Control*, 38(11):1623–1633, 1993. Cited on page 44.

M. Morari and J. H. Lee. Model predictive control: past , present and future. *Computers & Chemical Engineering*, 23:667–682, 1999. Cited on page 44.

K. R. Muske and T. Badgwell. Disturbance modeling for offset-free linear model predictive control. *Journal of Process Control*, 12(5):617–632, Aug. 2002. Cited on page 33.

K. R. Muske and J. B. Rawlings. Model predictive control with linear models. *AIChE Journal*, 39(2):262–287, Feb. 1993. Cited on pages 27, 31, 32, and 33.

Y. Nesterov and A. Nemirovskii. *Interior-Point polynomial Algorithms in Convex Programming*. SIAM, 1 edition, 1994. Cited on page 54.

V. Nevistic and L. Del Re. Feasible Suboptimal Model Predictive Control for Linear Plants with State Dependent Constraints. In *Proceedings of the 1994 American Control Conference*, volume 2862, pages 2862–2866, 1994. Cited on page 72.

G. D. Nicolao, L. Magni, and R. Scattolini. Stabilizing Nonlinear Receding Horizon Control via a Nonquadratic Terminal State Penalty. In *Proceedings of the symposium on control, optimization and supervision CESA'96*, pages 185–187, 1996. Cited on page 44.

G. D. Nicolao, L. Magni, and R. Scattolini. Stabilizing Receding-Horizon Control of Nonlinear Time-Varying Systems. *IEEE Transactions on Automatic Control*, 43(7):1030–1036, 1998. Cited on page 44.

G. D. Nicolao, L. Magni, and R. Scattolini. Stability and robustness of nonlinear receding horizon control. In *Nonlinear model predictive control*, pages 3–22. Birkhäuser Verlag Basel, 2000. Cited on page 44.

NLR. Aircraft Accident Report 92-11, El Al Flight 1862. Technical report, Nederlands Aviation Safety Board, 1992. Cited on page 5.

NTSB. China Airlines Boeing 747-SP, N4522V, 300 Nautical Miles Northwest of San Fransisco, Callifornia, February 19, 1985. Technical report, 1986. Cited on pages 3 and 4.

NTSB. In-Flight Separation of Vertical Stabilizer American Airlines Flight 587 Airbus Industrie A300-605R, N14053 Belle Harbor, New York November 12, 2001. Technical report, 2004. Cited on page 3.

G. Pannocchia and E. C. Kerrigan. Offset-free control of constrained linear discrete-time systems subject to persistent unmeasured disturbances. In *Proceedings of the 42th IEEE Conference on Decision and Control*, number December, pages 1–6, 2003. Cited on page 33.

G. Pannocchia and J. B. Rawlings. Disturbance models for offset-free model-predictive control. *AIChE Journal*, 49(2):426–437, Feb. 2003. Cited on page 33.

J. A. Primbs and V. Nevistic. MPC Extensions to Feedback Linearizable Systems. In *Proceedings of the 1997 American Control Conference*, volume 0, pages 2073–2077, 1997. Cited on page 72.

J. A. Primbs, V. Nevistic, and J. C. Doyle. Nonlinear optimal control: A control Lyapunov function and receding horizon perspective. *Asian Journal of Control*, 1(1):14–24, 1999. Cited on page 44.

A. I. Propoi. Use of linear programming methods for synthesizing sampled-data automatic systems. *Automation and Rempote Control*, 24(7):837–844, 1963. Cited on page 23.

D. M. Raimondo, O. Huber, M. S. Darup, M. Mönningmann, and M. Morari. Constrained time-optimal control for nonlinear systems : a fast explicit approximation. In *Proceedings of the 4th IFAC Nonlinear Model Predictive Control Conference*, pages 113–118, 2012. Cited on page 42.

C. V. Rao and J. B. Rawlings. Steady states and constraints in model predictive control. *AIChE Journal*, 45(6):1266–1278, June 1999. Cited on pages 32, 48, and 50.

J. B. Rawlings. Tutorial Overview of Model Predictive Control. *IEEE Control Systems Magazine*, 20(3):38–52, 2000. Cited on page 32.

J. B. Rawlings and D. Q. Mayne. *Model Predictive Control: Theory and Design*. Nob Hill Publishing, first edition, 2009. ISBN 978-0-9759377-0-9. Cited on page 36.

J. B. Rawlings and K. R. Muske. The Stability of Constrained Receding Horizon Control. *IEEE Transactions on Automatic Control*, 38(10):1512–1516, 1993. Cited on page 27.

J. B. Rawlings, E. S. Meadows, and K. R. Muske. Nonlinear Model Predictive Control: A Tutorial and Survey. In *IFAC Advanced Control of Chemical Processes*, pages 185–197, 1994. Cited on page 32.

J. A. Rossiter. A Global Approach to Feasibility in Linear MPC. In *UKACC ICC*, 2006. Cited on page 32.

M. Rubagotti, D. Barcelli, and A. Bemporad. Approximate Explicit MPC on Simplicial Partitions with Guaranteed Stability for Constrained Linear Systems. In *Proceedings of the 4th IFAC Nonlinear Model Predictive Control Conference*, number 2011, pages 119–125, 2012. Cited on page 42.

P. Scokaert, D. Mayne, and J. Rawlings. Suboptimal model predictive control (feasibility implies stability). *IEEE Transactions on Automatic Control*, 44(3): 648–654, Mar. 1999. Cited on page 44.

D. Simon, J. Löfberg, and T. Glad. Reference tracking MPC using terminal set scaling. In *51st IEEE Conference on Decision and Control (CDC)*, pages 4543–4548, Dec. 2012. Not cited.

D. Simon, J. Löfberg, and T. Glad. Nonlinear Model Predictive Control using Feedback Linearization and Local Inner Convex Constraint Approximations. In *Proceedings of the 2013 European Control Conference*, number 3, pages 2056–2061, 2013. Not cited.

D. Simon, J. Löfberg, and T. Glad. Reference Tracking MPC using Dynamic Terminal Set Transformation. *IEEE Transactions on Automatic Control*, 2014. Provisionally accepted for publication. Not cited.

S. Skogestad and I. Postlethwaite. *Multivariable Feedback Control Analysis and Design*. John Wiley & Sons, 2 edition, 2005. ISBN 978-0-470-01168-3. Cited on page 25.

B. L. Stevens and F. L. Lewis. *Aircraft Control and Simulation*. John Wiley & Sons, 2 edition, 2003. ISBN 0-471-37145-9. Cited on page 2.

M. Sznaier and M. J. Damborg. Suboptimal control of linear systems With state and control inequality constriants. In *Proceedings of the 26th IEEE Confer-*

ence on Decision and Control (CDC), number December, pages 761–762, 1987. Cited on pages 29 and 61.

M. C. Turner and I. Postlethwaite. Output Violation Compensation for Systems With Output Constraints. *IEEE Transactions on Automatic Control*, 47(9): 1540–1546, 2002. Cited on page 5.

M. C. Turner and I. Postlethwaite. Multivariable override control for systems with output and state constraints. *International Journal of Robust and Nonlinear Control*, 14(1314):1105–1131, Sept. 2004. Cited on page 5.

Licentiate Theses
Division of Automatic Control
Linköping University

P. Andersson: Adaptive Forgetting through Multiple Models and Adaptive Control of Car Dynamics. Thesis No. 15, 1983.

B. Wahlberg: On Model Simplification in System Identification. Thesis No. 47, 1985.

A. Isaksson: Identification of Time Varying Systems and Applications of System Identification to Signal Processing. Thesis No. 75, 1986.

G. Malmberg: A Study of Adaptive Control Missiles. Thesis No. 76, 1986.

S. Gunnarsson: On the Mean Square Error of Transfer Function Estimates with Applications to Control. Thesis No. 90, 1986.

M. Viberg: On the Adaptive Array Problem. Thesis No. 117, 1987.

K. Ståhl: On the Frequency Domain Analysis of Nonlinear Systems. Thesis No. 137, 1988.

A. Skeppstedt: Construction of Composite Models from Large Data-Sets. Thesis No. 149, 1988.

P. A. J. Nagy: MaMiS: A Programming Environment for Numeric/Symbolic Data Processing. Thesis No. 153, 1988.

K. Forsman: Applications of Constructive Algebra to Control Problems. Thesis No. 231, 1990.

I. Klein: Planning for a Class of Sequential Control Problems. Thesis No. 234, 1990.

F. Gustafsson: Optimal Segmentation of Linear Regression Parameters. Thesis No. 246, 1990.

H. Hjalmarsson: On Estimation of Model Quality in System Identification. Thesis No. 251, 1990.

S. Andersson: Sensor Array Processing; Application to Mobile Communication Systems and Dimension Reduction. Thesis No. 255, 1990.

K. Wang Chen: Observability and Invertibility of Nonlinear Systems: A Differential Algebraic Approach. Thesis No. 282, 1991.

J. Sjöberg: Regularization Issues in Neural Network Models of Dynamical Systems. Thesis No. 366, 1993.

P. Pucar: Segmentation of Laser Range Radar Images Using Hidden Markov Field Models. Thesis No. 403, 1993.

H. Fortell: Volterra and Algebraic Approaches to the Zero Dynamics. Thesis No. 438, 1994.

T. McKelvey: On State-Space Models in System Identification. Thesis No. 447, 1994.

T. Andersson: Concepts and Algorithms for Non-Linear System Identifiability. Thesis No. 448, 1994.

P. Lindskog: Algorithms and Tools for System Identification Using Prior Knowledge. Thesis No. 456, 1994.

J. Plantin: Algebraic Methods for Verification and Control of Discrete Event Dynamic Systems. Thesis No. 501, 1995.

J. Gunnarsson: On Modeling of Discrete Event Dynamic Systems, Using Symbolic Algebraic Methods. Thesis No. 502, 1995.

A. Ericsson: Fast Power Control to Counteract Rayleigh Fading in Cellular Radio Systems. Thesis No. 527, 1995.

M. Jirstrand: Algebraic Methods for Modeling and Design in Control. Thesis No. 540, 1996.

K. Edström: Simulation of Mode Switching Systems Using Switched Bond Graphs. Thesis No. 586, 1996.

J. Palmqvist: On Integrity Monitoring of Integrated Navigation Systems. Thesis No. 600, 1997.
A. Stenman: Just-in-Time Models with Applications to Dynamical Systems. Thesis No. 601, 1997.
M. Andersson: Experimental Design and Updating of Finite Element Models. Thesis No. 611, 1997.
U. Forssell: Properties and Usage of Closed-Loop Identification Methods. Thesis No. 641, 1997.
M. Larsson: On Modeling and Diagnosis of Discrete Event Dynamic systems. Thesis No. 648, 1997.
N. Bergman: Bayesian Inference in Terrain Navigation. Thesis No. 649, 1997.
V. Einarsson: On Verification of Switched Systems Using Abstractions. Thesis No. 705, 1998.
J. Blom, F. Gunnarsson: Power Control in Cellular Radio Systems. Thesis No. 706, 1998.
P. Spångéus: Hybrid Control using LP and LMI methods – Some Applications. Thesis No. 724, 1998.
M. Norrlöf: On Analysis and Implementation of Iterative Learning Control. Thesis No. 727, 1998.
A. Hagenblad: Aspects of the Identification of Wiener Models. Thesis No. 793, 1999.
F. Tjärnström: Quality Estimation of Approximate Models. Thesis No. 810, 2000.
C. Carlsson: Vehicle Size and Orientation Estimation Using Geometric Fitting. Thesis No. 840, 2000.
J. Löfberg: Linear Model Predictive Control: Stability and Robustness. Thesis No. 866, 2001.
O. Härkegård: Flight Control Design Using Backstepping. Thesis No. 875, 2001.
J. Elbornsson: Equalization of Distortion in A/D Converters. Thesis No. 883, 2001.
J. Roll: Robust Verification and Identification of Piecewise Affine Systems. Thesis No. 899, 2001.
I. Lind: Regressor Selection in System Identification using ANOVA. Thesis No. 921, 2001.
R. Karlsson: Simulation Based Methods for Target Tracking. Thesis No. 930, 2002.
P.-J. Nordlund: Sequential Monte Carlo Filters and Integrated Navigation. Thesis No. 945, 2002.
M. Östring: Identification, Diagnosis, and Control of a Flexible Robot Arm. Thesis No. 948, 2002.
C. Olsson: Active Engine Vibration Isolation using Feedback Control. Thesis No. 968, 2002.
J. Jansson: Tracking and Decision Making for Automotive Collision Avoidance. Thesis No. 965, 2002.
N. Persson: Event Based Sampling with Application to Spectral Estimation. Thesis No. 981, 2002.
D. Lindgren: Subspace Selection Techniques for Classification Problems. Thesis No. 995, 2002.
E. Geijer Lundin: Uplink Load in CDMA Cellular Systems. Thesis No. 1045, 2003.
M. Enqvist: Some Results on Linear Models of Nonlinear Systems. Thesis No. 1046, 2003.
T. Schön: On Computational Methods for Nonlinear Estimation. Thesis No. 1047, 2003.
F. Gunnarsson: On Modeling and Control of Network Queue Dynamics. Thesis No. 1048, 2003.
S. Björklund: A Survey and Comparison of Time-Delay Estimation Methods in Linear Systems. Thesis No. 1061, 2003.

M. Gerdin: Parameter Estimation in Linear Descriptor Systems. Thesis No. 1085, 2004.

A. Eidehall: An Automotive Lane Guidance System. Thesis No. 1122, 2004.

E. Wernholt: On Multivariable and Nonlinear Identification of Industrial Robots. Thesis No. 1131, 2004.

J. Gillberg: Methods for Frequency Domain Estimation of Continuous-Time Models. Thesis No. 1133, 2004.

G. Hendeby: Fundamental Estimation and Detection Limits in Linear Non-Gaussian Systems. Thesis No. 1199, 2005.

D. Axehill: Applications of Integer Quadratic Programming in Control and Communication. Thesis No. 1218, 2005.

J. Sjöberg: Some Results On Optimal Control for Nonlinear Descriptor Systems. Thesis No. 1227, 2006.

D. Törnqvist: Statistical Fault Detection with Applications to IMU Disturbances. Thesis No. 1258, 2006.

H. Tidefelt: Structural algorithms and perturbations in differential-algebraic equations. Thesis No. 1318, 2007.

S. Moberg: On Modeling and Control of Flexible Manipulators. Thesis No. 1336, 2007.

J. Wallén: On Kinematic Modelling and Iterative Learning Control of Industrial Robots. Thesis No. 1343, 2008.

J. Harju Johansson: A Structure Utilizing Inexact Primal-Dual Interior-Point Method for Analysis of Linear Differential Inclusions. Thesis No. 1367, 2008.

J. D. Hol: Pose Estimation and Calibration Algorithms for Vision and Inertial Sensors. Thesis No. 1370, 2008.

H. Ohlsson: Regression on Manifolds with Implications for System Identification. Thesis No. 1382, 2008.

D. Ankelhed: On low order controller synthesis using rational constraints. Thesis No. 1398, 2009.

P. Skoglar: Planning Methods for Aerial Exploration and Ground Target Tracking. Thesis No. 1420, 2009.

C. Lundquist: Automotive Sensor Fusion for Situation Awareness. Thesis No. 1422, 2009.

C. Lyzell: Initialization Methods for System Identification. Thesis No. 1426, 2009.

R. Falkeborn: Structure exploitation in semidefinite programming for control. Thesis No. 1430, 2010.

D. Petersson: Nonlinear Optimization Approaches to $\mathcal{H}_2$-Norm Based LPV Modelling and Control. Thesis No. 1453, 2010.

Z. Sjanic: Navigation and SAR Auto-focusing in a Sensor Fusion Framework. Thesis No. 1464, 2011.

K. Granström: Loop detection and extended target tracking using laser data. Thesis No. 1465, 2011.

J. Callmer: Topics in Localization and Mapping. Thesis No. 1489, 2011.

F. Lindsten: Rao-Blackwellised particle methods for inference and identification. Thesis No. 1480, 2011.

M. Skoglund: Visual Inertial Navigation and Calibration. Thesis No. 1500, 2011.

S. Khoshfetrat Pakazad: Topics in Robustness Analysis. Thesis No. 1512, 2011.

P. Axelsson: On Sensor Fusion Applied to Industrial Manipulators. Thesis No. 1511, 2011.

A. Carvalho Bittencourt: On Modeling and Diagnosis of Friction and Wear in Industrial Robots. Thesis No. 1516, 2012.

P. Rosander: Averaging level control in the presence of frequent inlet flow upsets . Thesis No. 1527, 2012.

N. Wahlström: Localization using Magnetometers and Light Sensors. Thesis No. 1581, 2013.

R. Larsson: System Identification of Flight Mechanical Characteristics. Thesis No. 1599, 2013.

Y. Jung: Estimation of Inverse Models Applied to Power Amplifier Predistortion. Thesis No. 1605, 2013.

M. Syldatk: On Calibration of Ground Sensor Networks. Thesis No. 1611, 2013.

M. Roth: Kalman Filters for Nonlinear Systems and Heavy-Tailed Noise. Thesis No. 1613, 2013.