Suboptimal Nonlinear Model Predictive Control

by

Rolf Heinz Findeisen

A thesis submitted in partial fulfillment

of the requirements for the degree of

Master of Science

(Chemical Engineering)

at the

UNIVERSITY OF WISCONSIN–MADISON

1997

Suboptimal Nonlinear Model Predictive Control

Approved by
James B. Rawlings:

____

To my Parents

Acknowledgments

I would like to express my thanks and appreciation to my advisor, Dr. James B. Rawlings. I thoroughly enjoyed working together with him and would wish everybody to have such an insightful and inspiring advisor. It is mostly due to his patience and guidance that I could reach my goal. Additionally I would like to say thank you to the following people.

Professor M. Zeitz who made it possible so that I could take part on, in my opinion, a very successful exchange program between the Universität Stuttgart and the University of Wisconsin–Madison. I hope this program will be also offered in the future, to allow other exchange students to enjoy and use the great facilities and possibilities here in Madison.

I would also like to express my gratitude to all students and former student of the Rawlings research group, who provided a great working and social environment: John Eaton, who created a fabulous computing environment and never complained about my frequent questions; Rahul Bindlish who proof-read major parts of this thesis and was always helpful and very polite, Rock Matthews and John Campbell, for the many interesting discussions; Chris Rao which helped me with many interesting discussions about MPC; Pierre Scokaert who once I started helped me to understand some of the basic MPC concepts; Claire Epley, our formerly group secretary; Scott Middlebrooks, Daniel Patience and Jianfeng Zhang.

Last but not least my brother Peter, I could have never finished without his steady support, help and often difficult and demanding questions, concerning his or my own research.

Rolf Heinz Findeisen

University of Wisconsin–Madison
February 1997

Suboptimal Nonlinear Model Predictive Control

Rolf Heinz Findeisen
Under the supervision of Professor James B. Rawlings
at the University of Wisconsin–Madison

In this work we consider the nonlinear model predictive control problem. The success of model predictive control for constrained linear systems is largely due to the fact that the resulting on-line optimization problem is a quadratic program for which reliable software exists. However, most of the existing model predictive control algorithms for nonlinear systems require the *exact global* solution of a *non-convex* nonlinear optimization problem to guarantee stability. In addition, most algorithms force the system to reach the origin after the end of the prediction horizon to guarantee a feasible solution at the next step. Even with state–of–the–art optimization algorithms, this seems to be practically impossible in finite time. Therefore it is important to modify the existing nonlinear model predictive algorithms to facilitate the solution of the resulting online optimization problem.

The first part of this thesis contains a review of the existing Model Predictive Control concepts for linear and nonlinear systems. We will see that for the nonlinear case controllers have been proposed, that overcome some of the stated problems. They permit feasible suboptimal solutions and allow the last prediction state to lie in a region instead of constraining it to zero. This can simplify the optimization problem significantly, however it is often difficult to check a priori if prerequisite assumptions are satisfied or if additional information for the algorithm must be provided.

In this thesis we propose a suboptimal model predictive control scheme for a specific class of nonlinear systems, namely systems for which the linearized system around the origin is stabilizable. This controller removes the terminal state constraint and requires only that we find solutions that are feasible and decrease the cost function. This scheme provides an asymptotically stabilizing controller while reducing the computational cost. An algorithm that uses a special optimizer that guarantees descent feasible sub steps is outlined.

In summary, this thesis reviews the state of model predictive control for nonlinear systems and proposes a new model predictive control algorithm that does not demand the *global* solution of a *non-convex nonlinear optimization problem.* This leads to a computationally more practical model predictive control algorithm.

Contents

List of Tables

List of Figures

Notation

Upper case symbols

A, B	State transition and input distribution matrices for linear systems.
C, D	Output matrices for linear systems.
B_r	$:= \{x : \|x\| \leq r\},\ r \in \mathbb{R}^+$. Ball with radius r around origin.
$\mathcal{B}, \hat{\mathcal{B}}$	Closed regions for stabilizing controller.
$\mathbf{C}^j$	Denotes the function space of the j times continuously differentiable functions.
G^x, G^u	Linear constraint matrixes for x and u, e.g. $G^x x \leq g^x \in \mathbb{R}^n$.
$\mathcal{K}^0$	A function $\alpha : \mathbb{R}^+ \to \mathbb{R}^+$ belongs to the class $\mathcal{K}^0$ if it is (1) continuous and (2) $\alpha(x) = 0 \Leftrightarrow x = 0$ (Def. 1.3).
$\mathcal{K}^+$	α belongs to $\mathcal{K}^+$ if (1) $\alpha \in \mathcal{K}^0$ and (2) it is nondecreasing (Def. 1.4).
$\mathcal{K}^\infty$	α is in the class $\mathcal{K}^\infty$ if (1) $\alpha \in \mathcal{K}^+$ and (2) $\alpha(x) \to \infty$ when $x \to \infty$ (Def. 1.5).
$\mathcal{K}$	α is called a $\mathcal{K}$-function if (1) $\alpha \in \mathcal{K}^+$ and (2) it is strictly increasing (Def. 1.6).
K	Gain/filter matrix or nonnegative integer in constrained asymptotic stability definition (Def. 1.9).
L	Stage cost function.
L^P	Final state penalty
M	$\in \mathbb{N}^+$ Positive integer used in Lyapunov theorem (1.1).
N^C	Control horizon, discrete time.
N^P	Prediction horizon, discrete time.
N	Horizon if prediction and control horizon coincide, discrete time.
P	Solution of algebraic riccati equation
Q, R	Weighting matrices.
$\tilde{Q}$	Final state penalty matrix

P	Solution of algebraic matrix Riccati equation.
T^P	Prediction horizon, continuous time.
T^C	Control horizon, continuous time.
T	Horizon if prediction and control horizon coincide, continuous time. Upper block traingular part in Schure decomposition, section 2.2.4
ΔT	Sampling time.
U	Orthogonal part in Schur decomposition, section 2.2.4
$\mathcal{U}$	Constraint input set.
V	Lyapunov function (Theo. 1.1).
$\mathcal{W}$	Region in which $L(x, Kx) \leq x^T Q x + x^T K^T R K x \quad \forall x \in \mathcal{W}$, section 3.1.
$\mathcal{X}$	Constraint state set.
$\mathbb{X}_K$	Positive invariant set for which the linear system with the LQR law $u_k = -Kx_k$ satisfies all the state and input constraints over an infinite horizon.

Lower case symbols

a	$\in \mathbb{R}^+$. Positive constant used in exponential stability definition (Def. 1.10).
f	State transition function.
g	Output function in dynamic system equations.
g^x, g^u	Constant vectors for linear x and u constraints.
h	Stabilizing control law around origin.
j, k, l	Integer valued time indices.
k_1, k_2, k_3	System constants, example 1, chapter 4.
m	Number of system inputs.
n	Number of system states.
p	Number of system outputs.
r	$\in \mathbb{R}^+$, radius.
s, t	Continuous valued time variable.
u	$\in \mathbb{R}^m$. Input, manipulated variable.
u_k	Input at discrete time k.

$u_{j\|k}$	Hypothetical input at discrete time j given actual time k.
$\bar{u}(t)$	Input at continuous time t.
$\bar{u}(s\|t)$	Hypothetical input at continuous time s given actual time t
$\bar{u}_k$	Discrete time input $\bar{u}(t) := \bar{u}_k$, $t \in [t_0 + k\Delta T, t_0 + (k+1)\Delta T]$ for continous time system
$\bar{u}_{j\|k}$	Discrete time discrete hypothetical input $\bar{u}(t\|t_0 + k\Delta T) := \bar{u}_{j\|k}$, $t \in [t_0 + j\Delta T, t_0 + (j+1)\Delta T]$ given actual time $t = t_0 + k\Delta T$.
Δu_k	Rate of change for u_k, $\Delta u_k := u_k - u_{k-1}$.
x	$\in \mathbb{R}^n$. State space variable.
x^e	Equilibrium or steady state, defined by $x^e := f(x^e)$.
x^1, x^2	States $c_A = x^1$ and $x^2 = c_B$ in example 1, chapter 4.
x_F	Feedstream, example 1, chapter 4.
x_k	State at discrete time k.
$x^u(j;k,x_k)$	Solution of $x_{l+1} = f(x_l, u_l, l)$ at time $j \geq k$ under the inputs $\{u_k, \dots, u_{j-1}\}$ with the initial state x_k at time k.
$x_{j\|k}$	$:= x^u(j;k,x_k)$. Prediction of x at time $j \geq k$ given $x_{k\|k}$ and the hypothetical input sequence $\{u_{k\|k}, \dots, u_{j-1\|k}\}$.
$\bar{x}(t)$	State at continuous time t.
$\bar{x}^u(s;t,\bar{x}(t))$	Solution of $\dot{\bar{x}}(\tau) = \bar{f}(\bar{x}(\tau), \bar{u}(\tau), \tau)$ at time $s \geq t$ under the input $\bar{u}(\tau)$ $\tau \in [t,s]$ with initial state $\bar{x}(t)$.
$\bar{x}(s\|t)$	$:= \bar{x}^u(s;t,\bar{x}(t)$. Prediction of $\bar{x}$ at time $s \geq t$ given $\bar{u}(\tau\|t)$ $\tau \in [t,s]$.
Δx_k	Rate of change for x_k, $\Delta x_k := x_k - x_{k-1}$.
y	$\in \mathbb{R}^p$. Output, measured variable.
y_k	Output at discrete time k.
$\bar{y}(t)$	Output at continuous time t.

Greek symbols

α, β	$\mathcal{K}^+$ functions used in Lyapunov theorem (Theo. 1.1).
$\gamma, \delta, \sigma, \epsilon$	$\in \mathbb{R}^+$. Positive real numbers used in stability definitions (Def. 1.8, Def. 1.9, Def. 1.10).
γ	$\mathcal{K}^+$ function
λ	Exponential factor used in exponential stability definition (Def. 1.10).

μ	Constant $\in (0, 1]$.
π_k	Hypothetical input sequence $\pi_k := \{u_{k\|k}, u_{k+1\|k}, \dots, u_{k+N^P-1\|k}\}$.
π_k^K	Hypothetical input sequence$\pi_k^K := \{Kx_{k\|k}, \dots, Kx_{k+N^P-1\|k}\}$
Φ	Cost function.
Φ^K	Cost under linear controller $u = Kx$, section 3.2.
τ	Continuous valued time variable.
ζ, ς	Additional state variables.

Other symbols

$\tilde{(\cdot)}$	Denotes feasible, but not optimal variables, chapter 3.
$\bar{(\cdot)}$	Denotes continuous time variables and symbols.
$(\cdot)^e$	Equilibrium values.
$(\cdot)^P$	Identifier for plant.
$(\cdot)^\star$	Denotes variables or sequences with optimal value.
$(\cdot)'$	Augmented variables or modified regions.
$(\cdot)^T$	Trasnsposed of matrix or vector.
$:=$	Left hand side is defined as.
$=:$	Right hand side is defined as.
$\\|x\\|$	$:= \sqrt{x^T x}$ Euclidean or l_2 norm of a vector $x \in \mathbb{R}^n$.
$\\|x\\|_P$	$:= \sqrt{x^T P x}$ weighted Euclidean norm of a vector $x \in \mathbb{R}^n$ with respect to a positive definite matrix $P \in \mathbb{R}^{n \times n}$
$\\|A\\|$	$:= \sup_{x \neq 0, x \in \mathbb{R}^n} \frac{\\|Ax\\|}{\\|x\\|}$. Induced Euclidean matrix norm of a matrix $A \in \mathbb{R}^{m \times n}$
$\mathbb{N}$	The set of nonnegative integers.
$\mathbb{N}^+$	The set of positive integers.
$\mathbb{R}^n$	Euclidean space of dimension n.
$\mathbb{R}^{n \times m}$	Matrix of size $n \times m$.
$\mathbb{R}^+$	Denotes the nonnegative reals.

Chapter 1

Introduction

1.1 Introduction and Motivation

Over the last decade model predictive control has emerged as a powerful and widely used control technique, especially in the process industries. Model predictive control (MPC) stands for a class of control approaches in which a model of the controlled system is used to predict the effect of the control input on the system and to select with this prediction the optimal control action with respect to a predefined control objective.

Most of the industrially applied MPC schemes work with a linear model of the process and a quadratic cost function subject to additional constraints on the states and inputs. The implicit inclusion of constraints might be the major reason that MPC has such great success in industrial applications. The necessity to satisfy constraints on the states and inputs arises naturally in the process industries. The maximum opening of a valve is limited due to the design. Environmental and safety consideration might make it necessary to avoid high temperatures or concentrations of side products. This seems to be the key feature, since other methods can hardly consider constraints while guaranteeing the local or global stability of the resulting closed loop system. Another feature of linear MPC is that the resulting optimization problem is a convex quadratic program. This class of optimization problems is well investigated and an online solution for medium sized systems can be computed online. If the optimizer fails, we know that it is not possible to solve this problem with the given constraints or horizon.

Besides this great success of linear MPC in the process industries there are still many open questions. Especially, the application of MPC to nonlinear processes is in an early stage. The transfer of the mathematical controller framework from the linear case to nonlinear case is surprisingly simple. However the resulting optimization problem is in most cases numerical intractable. To guarantee stability we would have to solve a global non-convex nonlinear optimization problem. Solutions that partially overcome this problem have been proposed. However, it seems that at this time there is no general and industrially implementable solution available. To find such a practical and computationally feasible solution method is desirable. Economical considerations force the industry to run processes more and more near or at their "optimum." This often means to operate the plant in a region of highly nonlinear behavior. A combination of a linear controller and a linear model resulting from the linearization of the nonlinear state equations might lead to instability or unsatisfying behavior. The cost due to application of a linear controller might be high compared to the theoretical achievable since the linear controller does not directly account for existing nonlinearities. The nonlinear MPC approach in contrast takes the nonlinearity of the model into account and allows for a wider range of operating conditions and a lower resulting cost. One industrial solution to chive a wider range of operating conditions is the design of more than one linear controller for different operation regions and to switch between them dependent on the actual system state. A sound proof of stability for this concept exist only for special cases. Another major point for the use of a nonlinear controller arrives from the modelling side. The modelling of a system from first principles often leads to highly nonlinear coupled equations. The application of a linear controller makes it necessary to linearize these equations around a steady state. It would be natural to remove this step and allow the control engineer to directly use the modelling equations.

It is the goal of this thesis to develop the theory for a new nonlinear model predictive Controller which seems to overcome some of the problems of existing NMPC concepts. We especially try to keep the solution of the resulting optimization and computational problems practical.

1.2 Thesis Overview

This work is organized as follows:

- The remainder of Chapter 1 contains some comments about the notation. The fact that MPC calculations contain a prediction of future states of the systems under a proposed input sequence makes it necessary to distinguish between the open and the closed loop system values. Hereafter we introduce the system models under consideration. We mostly deal with nonlinear time-invariant discrete time systems, but since major parts of the existing literature consider continuous time models, we also present them for necessary notation.

- In Chapter 2 we state the general MPC framework for discrete and continuous time. We proceed with a review of the major results of MPC for linear systems. The bulk of this chapter is devoted to the study of existing nonlinear

MPC algorithms. We try to clarify the major differences between these existing schemes and show their advantages and disadvantages.

- Chapter 3. During this part of the work we derive a new nonlinear MPC strategy. This strategy is based on the "ideal" MPC algorithm, which contains an infinite control and prediction horizon. A solution of this problem is not possible in general, so we limit ourselves to systems with a stabilizable linearization around the steady state. This allows us to modify the algorithm to achieve a computationally tractable problem. We prove the convergence and stability once a initial feasible solution is found.

- Chapter 4 begins with the discussion of the encountered optimization problems and a presentation of possible solution methods. We consider optimization algorithms that can make use of the inherent structure of the system and optimizers that can guarantee feasibility at every sub-step. We conclude this chapter with two illustrative examples.

- The final chapter serves as a review of the presented work and discusses future research directions.

1.3 Comments about the Notation

Throughout the thesis the symbol ":=" means that the left side is defined to be equal to the right hand side; similarly "=:" means that the right hand side is defined as the left hand side.

Most of our work is concerned with the discrete time case, but sometimes it is not possible to avoid continuous time formulations. For simplicity, we attempt to use similar symbols for the continuous and discrete time variables. Continuous time variables are differentiated from discrete time variables by a bar above the symbols. $\bar{u}$ means the input for a continuous time system, whereas u stands for the discrete time counterpart. Subscripts like x_k account for a discrete or integer time variable. In the case of a real time variable we use the standard $\bar{x}(t)$ notation. We also use integer valued time variables for continuous time systems. One example is the use of time discontinuous inputs $\bar{u}_k$. By this we mean a constant input $\bar{u}(t) := \bar{u}_k \ \forall\ t \in [t_0 + k\Delta T, t_0 + (k+1)\Delta T]$ over the sampling time ΔT. Here t_0 denotes an initial time and ΔT a fixed constant sampling interval.

Further $\bar{x}^u(s; t, x(t))$ denotes the solution of $\dot{\bar{x}} = \bar{f}(\bar{x}, \bar{u}, t)$ at time s under the input $u(\tau)$ $(\tau \in [t, s])$ with the initial state $\bar{x}(t)$ at time t. Similar $\bar{x}^u(j; l, x_l)$ denotes the solution of $x_{k+1} = f(x_k, u_k, k)$ at time j under the input u_k $(k = l, \ldots, j-1)$ with the initial state x_l at time l.

Mathematical representations of MPC concepts have to distinguish between the calculated open-loop control sequence, state trajectories and the actual response of the system, the closed-loop values. In the discrete case, we distinguish between these by using double subscripts such as $x_{j|k} := x^u(j; k, x_k)$ $j \geq k$. This stands for the prediction of x at time j given the value $x_{k|k} := x_k$. We implicit assume hereby the input sequence $\{u_{k|k}, u_{k+1|k}, \dots , u_{j-1|k}\}$. In the continuous time case we use $\bar{x}(s|t) := \bar{x}^u(s; t, \bar{x}(t))$ with the open-loop input $\bar{u}(\tau|t)$ $\tau \in [t, s]$.

We consider only the nominal case. This means that the model (of the plant) and the actual or real plant coincide and that we are not concerned about disturbances. Whenever it is necessary to differentiate the actual plant and the model, we use a superscript $(\cdot)^P$ to identify the plant. If we do not use $(\cdot)^P$ the reader can assume that the plant and the model coincide.

The $(\cdot)^\star$ superscript denotes variables or sequences with optimal values, so for example $x^\star = \min_x f(x)$.

1.4 System Models

During the whole presentation we assume for simplicity the absence of time-delays.

Continuous Time Nonlinear Systems: In its most general form, continuous time systems can be given by the following differential equations.

$$\dot{\bar{x}}(t) = f(\bar{x}(t), \bar{u}(t), t) \qquad \bar{x}_0 =: \bar{x}(0) \quad \text{given} \tag{1.1}$$

$$\bar{y}(t) = \bar{g}(\bar{x}(t), \bar{u}(t), t) \tag{1.2}$$

Here $t \in \mathbb{R}$ represents the continuous time.

Discrete Time Nonlinear Systems:

$$x_{k+1} = f(x_k, u_k, k) \qquad x_0 \quad \text{given} \tag{1.3}$$

$$y_k = g(x_k, u_k, k) \tag{1.4}$$

Here $k \in \mathbb{N}$ stands for the integer valued time.
In both cases $x \in \mathbb{R}^n$ denotes the state, $u \in \mathbb{R}^m$ the input or manipulated variable and $y \in \mathbb{R}^p$ the output or controlled variable. If $\bar{f} : \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R} \to \mathbb{R}^n$ (or $f : \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{N} \to \mathbb{R}^n$ in the discrete case) and $\bar{g} : \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R} \to \mathbb{R}^p$ ($g : \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{N} \to \mathbb{R}^p$) do not explicit depend on t (k) we talk about time invariant systems. In the following g and f are time invariant.
Furthermore we assume that the origin is a steady state for the given system, $f(0, 0) = 0$. Additional we require that f and g are continuous, $f, g \in \mathbf{C}^0$.
Remarks: In general it is not possible to go from the continuous time model to the

discrete time model. We would need an explicit solution of the state equations with respect to $\bar{u}(t)$. $\triangle$

If f, g are linear in x and u we talk about linear systems.

$$\dot{\bar{x}} = \bar{A}(t)\bar{x} + \bar{B}(t)\bar{u}(t) \qquad \bar{x}_0 := \bar{x}(0) \quad \text{given} \tag{1.5}$$
$$\bar{y} = \bar{C}(t)\bar{x} + \bar{D}(t)\bar{u}(t) \tag{1.6}$$
$$\tag{1.7}$$

or

$$x_{k+1} = A_k x_k + B_k u_k \qquad x_0 \quad \text{given} \tag{1.8}$$
$$y_k = C_k x_k + B_k u_k \tag{1.9}$$

For physical continuous time systems $\bar{u}$ normally vanishes from $\bar{y} = \bar{y}(\bar{x}(t), t)$. This is in general not true for the discrete time case, especially if the equations result from the exact integrated continuous equations [46].

1.5 Important Mathematical Definitions

During the rest of the thesis $\mathbb{R}^+$ denotes the nonnegative reals.

Definition 1.1 (Norms) *The $\|x\|$ denotes the Euclidean or l_2 norm of an vector $x \in \mathbb{R}^n$, whereas $\|x\|_P$ denotes the weighted Euclidean norm of a vector with respect to a positive definite matrix $P \in \mathbb{R}^{n \times n}$. In the case of a matrix $A \in \mathbb{R}^{m \times n}$ $\|A\|$ means the induced matrix norm:*

$$\begin{aligned} \|x\| &:= \sqrt{x^T x} \\ \|x\|_P^2 &:= x^T P x \\ \|A\| &:= \sup_{x \neq 0, x \in \mathbb{R}^n} \frac{\|Ax\|}{\|x\|} \end{aligned}$$

Definition 1.2 (Region $\boldsymbol{B_r}$) *The set B_r with $r \in \mathbb{R}^+$ denotes the ball with radius r in $\mathbb{R}^n$.*

$$B_r := \{x : \|x\| \leq r\}$$

Definition 1.3 ($\boldsymbol{\mathcal{K}^0}$ function) *A function $\alpha : \mathbb{R}^+ \to \mathbb{R}^+$ belongs to the class $\mathcal{K}^0$ if (1) it is continuous and (2) $\alpha(x) = 0 \Leftrightarrow x = 0$.*

Definition 1.4 ($\boldsymbol{\mathcal{K}^+}$ function) *α belongs to $\mathcal{K}^+$ if (1) $\alpha \in \mathcal{K}^0$ and (2) it is nondecreasing*

Definition 1.5 ($\mathcal{K}^\infty$ function) *α is in the class $\mathcal{K}^\infty$ if (1) $\alpha \in \mathcal{K}^+$ and (2) $\alpha(x) \to \infty$ when $x \to \infty$*

Definition 1.6 ($\mathcal{K}$-function) *A function α is called a $\mathcal{K}$-function if (1) $\alpha \in \mathcal{K}^+$ and (2) it is strictly increasing*

Remarks: Notice the difference between a $\mathcal{K}$-function (Definition 1.6) and a function which belongs to the class $\mathcal{K}^+$ (Definition 1.4). A $\mathcal{K}$-function is strictly increasing whereas a function of the class $\mathcal{K}^+$ is only nondecreasing. $\triangle$

Since we deal with constrained systems, it is necessary to slightly modify the usual Lyapunov stability definitions as used in Kalman and Bertram ([23],[24]), Kwakernaak and Sivan [29] or Vidyasagar [68]. The stability definitions and theorems are modified versions of the ones Keerthi and Gilbert [27] presented. We consider time-invariant, nonlinear discrete time systems:

$$x_{k+1} = f(x_k), \qquad x_k \in \mathcal{X} \subset \mathbb{R}^n \tag{1.10}$$

with $f : \mathcal{X} \to \mathcal{X}$, where $\mathcal{X}$ denotes the constrained subspace of $\mathbb{R}^n$.

Definition 1.7 (Steady State for Constrained Systems) *A state $x^e \in \mathbb{R}^n$ is called an equilibrium or steady state for the constrained system (1.10) if (1) $x^e \in \mathcal{X}$ and (2) $x^e := f(x^e)$.*

Remarks: From now on we assume that $x = 0$ is a steady state (s.s.) for the system (1.10) and that $x = 0, u = 0$ is a steady state of (1.1,1.3). $\triangle$

Definition 1.8 (Constrained Stability) *The steady state $x_0 = 0$ is stable if, for every $\epsilon > 0$, $\exists\, \delta(\epsilon) > 0$, such that*

$$\|x_j\| \leq \epsilon, \qquad \forall j \geq k,\ x_k \in \mathcal{X},\ x_k \in B_\delta$$

Remarks: The major difference between this stability definition and the traditional stability definition is that the considered states must lie not only in the region B_δ but also in the set $\mathcal{X}$. $x_k \in \mathcal{X}$ enforces $x_j \in \mathcal{X}\ \forall j \geq k$ since $f : \mathcal{X} \to \mathcal{X}$. $\triangle$

Definition 1.9 (Constrained Asymptotic Stability) *The origin is asymptotically stable, if (1) it is stable, (2) $\exists\, \gamma > 0$ and, for any $\sigma > 0$ $\exists\, K(\sigma) \in \mathbb{N}$ such that*

$$\|x_j(x_k)\| \leq \sigma \qquad \forall j \geq k + K, \quad x_k \in \mathcal{X}, \quad x_k \in B_\gamma$$

Remarks: *Notice that the second requirement implies for all $\|x_k\| \in B_\gamma$ that $\|x_j\| \to 0$ as $j \to \infty$.* $\triangle$

Definition 1.10 (Constrained Exponential Stability) *The origin is exponentially stable if there exists $\delta > 0, a > 0,$ and $0 \leq \lambda < 1$ such that*

$$\|x_j(x_k)\| \leq a\|x_k\|\lambda^{j-k} \qquad \forall j \geq k,\ \|x_k\| < \delta,\ x_k \in \mathcal{X}$$

The following theorem is taken form Keerthi [27].

Theorem 1.1 (Lyapunov) *Given a $V : \mathbb{R}^n \to \mathbb{R}$ that satisfies the following: There exist $\alpha, \beta \in \mathcal{K}^+$, $\gamma \in \mathcal{K}^0$, $r \in \mathbb{R}^+$ and a positive integer $M \in \mathbb{N}^+$, such that*

1. $V(x) \leq \beta(\|x\|), \quad \forall x \in \mathcal{X},\ x \in B_r$
2. $\alpha(\|x\|) \leq V(x), \quad \forall x \in \mathcal{X}$
3. $V(x_j) - V(x_{j+M}(x_j)) \geq \gamma(\|x_j\|),\ V(x_j) - V(x_{j+1}(x_j)) \geq 0, \quad \forall x_j \in \mathcal{X}$

Then the zero steady-state solution of (1.10) is locally asymptotically stable with the region of attraction $\mathcal{X}$.

Chapter 2

MPC Review

This chapter contains a presentation of the general MPC framework and a review of the main results for linear and nonlinear MPC. It is divided in the following three subparts:

- In the first section we outline a generic scheme which seems to be inherent in most MPC strategies. Hereafter we give the basic mathematical MPC formulation for the continuous and discrete time cases.

- The second part of this chapter gives a short review of the main results of linear model predictive control. This is important since many of the basic ideas used in Nonlinear Model Predictive Control have their origins in existing linear MPC control concepts. We limit our presentation to MPC in the state space formulation.

- The third part contains the most important material for the subsequent chapters. We discuss the various existing nonlinear MPC approaches. A somewhat surprising outcome of this discussion is that certain questions such as nominal stability are not much more difficult to answer than in the linear case. It is more difficult to implement these concepts, since most of them require the on-line solution of a non-convex optimization problem.

2.1 A General MPC Framework

2.1.1 What is MPC?

In this work, model predictive control is defined as a control strategy in which the controller determines an input profile. This profile optimizes a given performance objective over a time interval, that spans from the current time to a time given by the prediction horizon in the future. The first part of the resulting manipulated variable profile is implemented and the feedback is incorporated by repeating the optimization for the next control step. This scheme is illustrated in Figure 2.1.

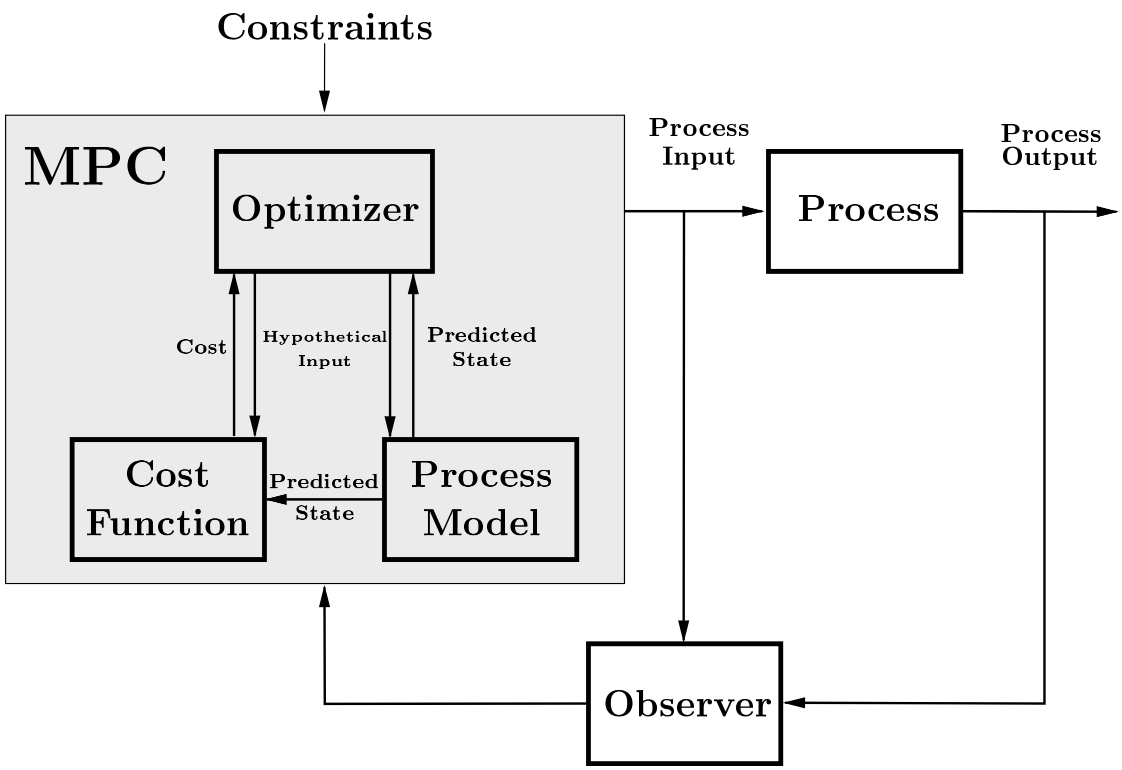

Figure 2.1: Basic MPC structure

Remarks: The above definition is standard, but other definitions have appeared in the literature. Garcia, Prett and Morari [16] refer to MPC as “that family of controllers in which there is a direct use of an explicit and separately identifiable model.” This is much broader and it misses the important feature of a prediction horizon that might be infinite or finite, fixed or variable.
Often MPC is referred to as “based on the solution of a open-loop optimal control problem at each sample time, and *no explicit provision* is made in the problem formulation to *ensure closed-loop stability*” [62]. This definition does not allow the inclusion of explicit stability constraints in the optimization problem. Under this definition predictive controllers with an explicit stability constraint are referred to as receding horizon control (RHC). In general MPC is used in the process and chemical engineering literature whereas RHC or moving horizon control (MHC) is used in the electrical and control literature. For a more complete review of this topic see Kwon [31]. △

It is important to recognize that in our generic scheme the estimation of the current system state is not a part of the MPC controller. In Figure 2.1 the observer or state estimator is not included in the MPC controller. For simplicity we in general assume that an exact state measurement is available. A presentation of state estimation methods would exceed the scope of this work.

The defining features of the model predictive control scheme can be summarized as follows:

- a supplied *process model* is used to predict the future behavior of the system based on the current state information and a *hypothetical* input profile.
- a user defined *cost or objective function* gives quantitative information about the performance of the controlled system over the prediction horizon.
- the control profile is determined by *optimizing* the cost function over the hypothetical input profile with respect to additional constraints on the inputs and states. This leads to an *open-loop optimal* control sequence.
- feedback is introduced by repeating this optimization step after applying the first part of the determined input profile.

As mentioned earlier, the cost function gives quantitative information about the system over the prediction horizon T^P (or N^P in the discrete case). Many MPC strategies introduce an additional horizon, the control horizon T^C or N^C respectively. After the control horizon the input profile is set to 0 or fixed to the last calculated value at the end of the control horizon.

$$\begin{array}{rcl} u_{k+j|k} := u_{k+N^C-1|k} & N^C - 1 < j \leq N^P - 1 & \text{discrete time case} \\ u(t+\tau|t) := u(t+T^C|t) & T^C < \tau \leq T^P & \text{continous time case} \end{array}$$

Here t or k stands for the current time (Figure 2.2). When the control and prediction horizons coincide we refer to the horizon as T or N.

The prediction and control horizon can both be finite or infinite. For computational reasons the control horizon is usually finite. Otherwise the resulting optimization problem has, even in the discrete case, an infinite number of decision variables. For linear systems with a quadratic cost function, it is possible to keep an infinite prediction horizon [45] whereas this is in general not possible for nonlinear systems. Lately it has been shown that an infinite prediction and control horizon for linear systems can be implemented ([9],[57]). Further both horizons can be time-varying ([36],[43],[38]). This variable horizon formulation has recently gained more interest since it has several important advantages over a fixed horizon formulation. It removes the necessity to know in advance the fixed horizon length. It can reduce the computational effort in the disturbance free case since an initial feasible trajectory is also feasible at the next time-step. Additionally the variable horizon offers in some cases a solution to the problem of infeasiblity with respect to state and input constraints. If necessary the horizon can be increased or decreased to lead to a feasible problem or reduce the necessary calculations.

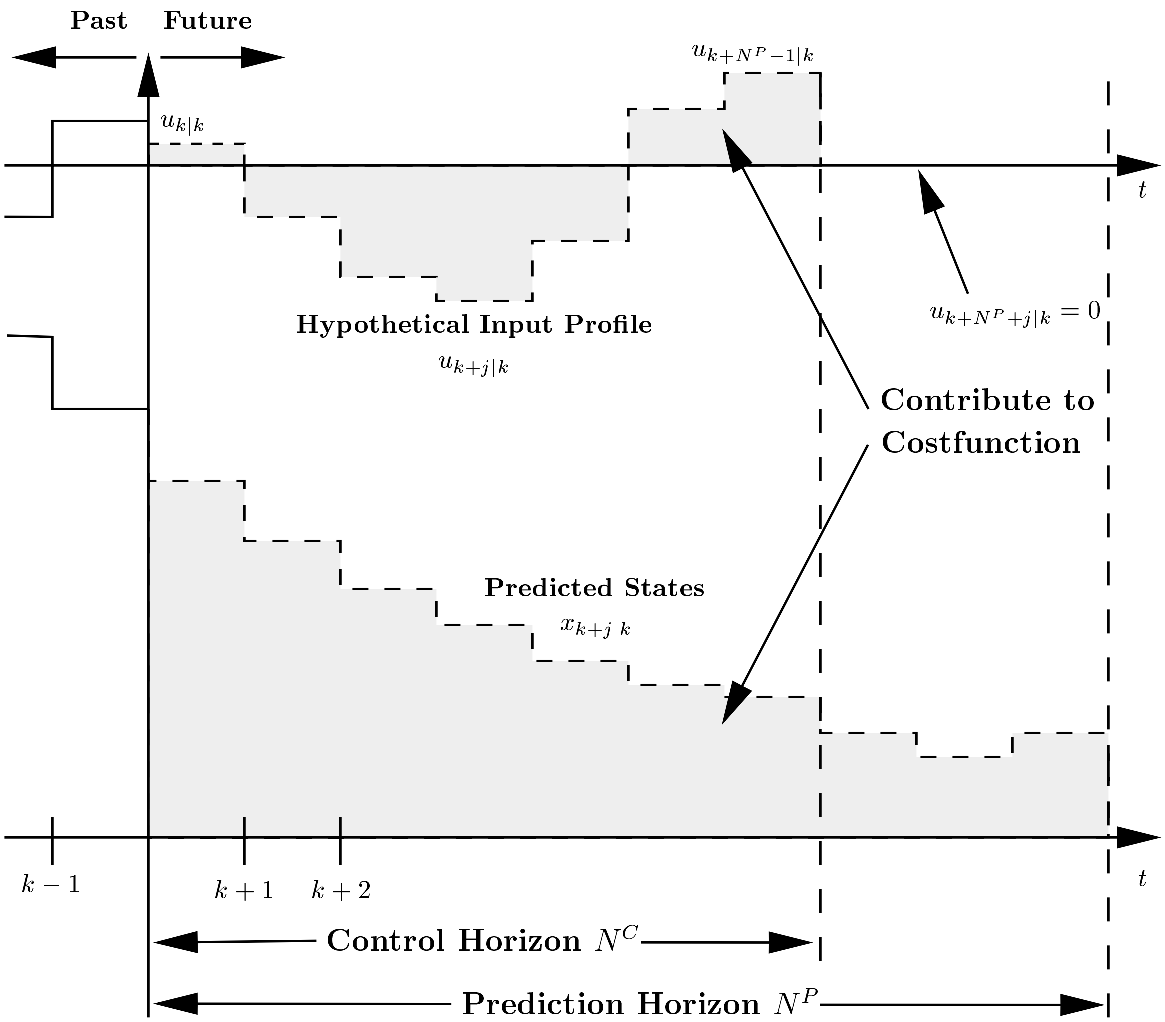

Figure 2.2: Prediction for discrete time systems

2.1.2 A Mathematical Prototype

A mathematical formulation of continuous time MPC can be given as follows:

$$\min_{\substack{\bar{u}(\tau|t)\\ \tau\in[t,t+T^P]}} \quad \bar{\Phi}(t, \bar{x}(t); \bar{u}(\tau|t)) \tag{2.1}$$

s.t.

$$\begin{aligned} \dot{\bar{x}}(\tau|t) = \bar{f}(\bar{x}(\tau|t), \bar{u}(\tau|t)) \quad \tau \in [t, t+T^P] \\ \bar{x}(t|t) := \bar{x}(t) \end{aligned} \tag{2.2}$$

with the additional input and state constraints

$$\bar{u}(\tau|t) \quad \in \mathcal{U} \subset \mathbb{R}^m \tag{2.3}$$

$$\bar{x}(\tau|t) \quad \in \mathcal{X} \subset \mathbb{R}^n \tag{2.4}$$

Both sets $\mathcal{X}$ and $\mathcal{U}$ contain the origin. Usually these sets are be described by a series of inequality equations:

$$\bar{G}(\tau, \bar{u}(\tau|t), \bar{x}(\tau|t)) \leq 0 \tag{2.5}$$

The control loop is closed by applying the first part of this hypothetical input to the system

$$\bar{u} := \bar{u}(\tau|t) \quad \tau \in [t, t + \Delta T] \tag{2.6}$$

Here:

$\bar{\Phi}$	objective function, performance criteria
$\bar{f}$	model dynamics
$\bar{x}$	model state vector
$\bar{u}$	input or control vector
T^P	prediction horizon
$\mathcal{U}$	constraint input set
$\mathcal{X}$	constraint state set
$\bar{G}$	inequality constraint function

For simplicity we have not included constraints on the rates of change $\dot{\bar{x}}$ and $\dot{\bar{u}}$.

The difference between the classical optimal control and this approach is that we compute the whole input sequence $\bar{u}(\tau|t)$ $(\tau \in [t, t + T^P])$ but only apply the first part of it $(\tau \in [t, t + \Delta T])$. Here $T \leq T^P$ is called the sampling period and a new control sequence under the usage of new state estimates is hereafter calculated. It is worth pointing out that for $n = m$ (number of inputs = number of outputs) and for the special system $\dot{\bar{x}} = \bar{u}$ our control problem for one sampling period becomes basically the "classical" calculus of variations. Often many results of the optimal control theory are generalizations or special cases of older facts of the calculus of variations, especially for linear systems with a quadratic cost function ([63],[67]). From this we can suspect the difficulties that we might encounter during the solution process of 2.1.
usually $\bar{\Phi}$ is given by

$$\bar{\Phi}(t, \bar{x}(t); \bar{u}(\tau|t)) := \int_{\tau=t}^{t+T^P} \bar{L}(t; x(\tau|t), u(\tau|t))d\tau + \bar{L}^P(t; x(t + T^P|t)) \tag{2.7}$$

with the positive definite functions $\bar{L} : \mathbb{R} \times \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$ and $\bar{L}^P : \mathbb{R} \times \mathbb{R}^n \to \mathbb{R}$. $\bar{L}^P$ is called the final state cost or final state penalty. Often $\bar{L}$ and $\bar{L}^P$ be quadratic time invariant in $\bar{x}$ and $\bar{u}$.

$$\begin{aligned} \bar{L}(\bar{x}(\tau|t), \bar{u}(\tau|t)) &= \|\bar{x}(\tau|t)\|_{\bar{Q}} + \|\bar{u}(\tau|t)\|_{\bar{R}} && (2.8) \\ \bar{L}^P(\bar{x}(t + T^P)) &= \|\bar{x}\left(\tau + T^P\right)\|_{\bar{Q}^P} && (2.9) \end{aligned}$$

The resulting optimization problem is intractable; we have to calculate a time continuous input trajectory $u(\tau|t)$. Many MPC implementations try to overcome this problem by generating a piecewise constant control sequence,
$\bar{\pi} := \{\bar{u}_{k|k}, \dots, \bar{u}_{k+N^P-1|k}\}$; the control input is fixed over the sampling time ΔT, $\bar{u}(t) := \bar{u}_k \ \forall \ t \in [t_0 + k\Delta T, t_0 + (k+1)\Delta T]$. During the optimization we have to integrate the system equation with the above given sequence as the input. The optimizer normally calculates a derivative of the cost function with respect to the inputs using finite difference methods. This can lead to numerical instability, since the cost function was already calculated via numerical integration. If a closed solution for the system is available, it is possible to transfer the continuous time problem to a discrete one for which no integration is necessary.

For the discrete time case we get

$$\min_{\pi_k} \Phi(k, x_k; \pi_k) \tag{2.10}$$

s.t.

$$\begin{aligned} x_{j+1|k} &= f(x_{j|k}, u_{j|k}) \\ x_{k|k} &:= x_k \end{aligned} \tag{2.11}$$

with the input and state constraints

$$u_{j|k} \in \mathcal{U} \subset \mathbb{R}^m \tag{2.12}$$

$$x_{j|k} \in \mathcal{X} \subset \mathbb{R}^n \tag{2.13}$$

or

$$G(j, u_{j|k}, x_{j|k}) \leq 0 \tag{2.14}$$

π_k denotes the hypothetical input sequence $\pi_k := \{u_{k|k}, u_{k+1|k}, \dots, u_{k+N^P-1|k}\}$.
We close the control loop similar to the continuous case:

$$u_k := u_{k|k} \tag{2.15}$$

All symbols have the same meaning as in the continuous time case. The cost function is a sum over all stage costs plus an additional final state penalty term.

$$\Phi(x_k; u_{j|k}) := \sum_{j=k}^{k+N^P-1} L_k(x_{j|k}, u_{j|k}) + L^P(x_{k+N^P|k}) \tag{2.16}$$

with $L : \mathbb{N} \times \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$ and $L^P : \mathbb{N} \times \mathbb{R}^n \to \mathbb{R}$ positive definite functions.

To keep the presentation clear we have not included rate of change variables in the cost function; neither have we allowed constraints on them. With rate of change we mean $\dot{x}$ and $\dot{u}$ in the continuous case and $\Delta u_k := u_k - u_{k-1}$, $\Delta x_k := x_k - x_{k-1}$ in the discrete case. However constraining these variables is often required in practice. The closing speed of a valve could be limited to avoid heavy wear or the maximum temperature change might be limited due to safety reasons. For the discrete case this is no real limitation. We can augment the state equations so that it is possible to formulate these changes as functions of the resulting augmented state vector:

$$\begin{aligned} \underbrace{\begin{bmatrix} x_{k+1} \\ \zeta_{k+1} \\ \varsigma_{k+1} \end{bmatrix}}_{} &= \underbrace{\begin{bmatrix} f(x_k, u_k) \\ x_k \\ u_k \end{bmatrix}}_{} \\ x'_{k+1} &= f'(x_k, u_k) \end{aligned} \tag{2.17}$$

This also leads to a new region $\mathcal{X}'$ and stage cost functions L', Ld'^P.

2.2 Linear MPC, Short Review

In the following we review the major results of linear MPC. This presentation is not intended to be complete, because we limit ourselves to state space descriptions. For a more complete review of linear MPC including the finite impulse response formulations like the famous IDCOM [55] and DMC [11] see ([31], [16], [51], [32]).

2.2.1 The Linear Quadratic Regulator (LQR)

Modern linear MPC has its roots in the linear quadratic regulator (LQR) problem, which was solved by Kalman [22] over 3 decades ago. For the LQR the process model is linear and the objective takes the form of a quadratic function in x_k and u_k must be minimized over an infinite horizon. If there are no constraints then the resulting problem takes the form:

$$\min_{\pi} \Phi_\infty(x_k, \pi) \tag{2.18}$$

with

$$\Phi_\infty(x_k, \pi) = \sum_{j=0}^{\infty} x_j^T Q x_j + u_j^T R u_j \quad Q \geq 0,\ R > 0 \tag{2.19}$$

$$x_{k+1} = A x_k + B u_k \tag{2.20}$$

We omit here double indices because for the infinite horizon case open loop control is the same as closed loop control:

$$u_k(x_k) = -Kx_k \tag{2.21}$$
$$K := -(B^TPB + R)^{-1}B^TPA \tag{2.22}$$

Here P is the solution of an algebraic Riccati equation.

Kalmans original work considered continuous time processes. He showed that the resulting cost is finite and that the LQR guarantees stability if the process (A, B) is stabilizable. The stability domain of the linear system with the linear controller is the whole state space. For the undisturbed case the open and closed loop optimal solution coincide since the horizon is infinite.

For the theoreticians LQR was a big step, but the practitioners struggled with its use. First LQR seemed to offer only a small amount of improvement compared to the widely accepted and well understood PID type controllers. This is mainly due to the fact that the usage of LQR requires tuning expertise to get the desired results. A second more important point is the lack of constraint handling. The "pure" PID controller cannot handle simple constraints as input saturation, however ad-hoc solutions such as anti-windup were available which partly overcome these problems.

2.2.2 Finite Control and Prediction Horizon Linear MPC

A logical expansion of the existing LQR concept was the inclusion of constraints on states and inputs. This inclusion however made it necessary to use a finite horizon since the optimization problem would otherwise be off limits. The resulting optimization is a quadratic program (QP):

$$\min_{\substack{\pi_k \\ u_{j|k} \in \mathcal{U} \\ x_{j|k} \in \mathcal{X}}} \Phi_{N^C}(x_k, \pi_k) \tag{2.23}$$

where $\mathcal{X}$ and $\mathcal{U}$ are the constrained input and state regions defined by the following linear inequalities:

$$\mathcal{X} = \{x : G^x x \leq g^x \in \mathbb{R}^n\} \quad \mathcal{U} = \{u : G^u u \leq g^u \in \mathbb{R}^m\} \tag{2.24}$$

with g^x and g^u are strictly positive constant vectors, so that the origin is contained in the resulting sets. Here we need the double indices, because the horizon is finite and we have constraints on the state and input variables, that leads to a state dependent solution of the QP

$$\Phi_k(x_k, \pi_k) = \sum_{j=0}^{N^P-1} x_{j|k}^T Q x_{j|k} + \sum_{j=0}^{N^C-1} u_{k|j}^T R u_{j|k} \quad Q \geq 0 \; R > 0 \tag{2.25}$$
$$x_{j+1|k} = Ax_{j|k} + Bu_{j|k} \tag{2.26}$$

We set $u_{j|k} = 0,\ j \geq (k + N^C)$ after the end of the control horizon. Many of the industrial used MPC algorithms like QDMC [15] and IDCOM [55], even if they do not use the state space description, owe their success to the fact that they could efficiently handle constraints. It is important to realize that these scheme do not guarantee stability for every choice of Q, R or N. Most practitioners are willing to check with simulations if their choice of parameters leads to a "stable" solution. However this seems to be an unnecessary burden and leads to a search for more satisfying algorithms. Bitmead et al[5] gives a more complete insight in the arising problems.

2.2.3 Finite Horizon End-Point Constraint MPC

There have been several methods proposed to overcome the problems discussed in the previous section. The first possibility would be to return to the infinite horizon under the additional consideration of the constraints.This however results in most cases in an unsolvable optimization problem if attacked in the wrong way. We show later that a general solution to this constrained LQR is possible in finite time, ([9],[57]). Another way would be to add a large enough penalty term at the end of the horizon

$$x_{k+N|k}^T \tilde{Q} x_{k+N|k} \tag{2.27}$$

which would force the state to be near the origin and would stabilize the system [5].

One of the first contributions to solve this problem was to constrain the states to return to zero after the prediction horizon ([28],[30]). This "stabilizing" constraint gained special importance in the case of nonlinear MPC and seemed to be for a long time the only way to guarantee stability for this class of systems ([38],[26]). The arising MPC formulation is similar to (2.23) with the additional end-point constraint

$$x_{k+N^P|k} = 0 \tag{2.28}$$

The closed-loop stability of this controller can be proven in a similar way as in the infinite horizon case (2.19), provided that a feasible solution exists. We outline the proof in section 2.3.1 for the more general nonlinear case. Details can be found in the paper of Keerthi and Gilbert [26] or in the review article of Rawlings, Meadows and Muske [53]. The requirement that *every* state has to return to 0 after the end of the prediction horizon can be very restrictive and can lead to "aggressive" behavior or undesirable properties of the controller. Even in the unconstrained case this requires that the system is controllable. This is very restrictive compared to the infinite horizon unconstrained case, in which the system has to be stabilizable only. As a result, the control and prediction horizon have to be much longer to allow feasibility. It seems important to mention that the resulting optimal cost gives an upper bound for the infinite horizon cost. x and u are zero at the end of the control and prediction horizon and because of this the rest of the infinite sum vanishes. Since all the constraints are enforced over the prediction horizon, this automatically leads to a feasible solution for the infinite horizon problem. We can interpret this as an "emulation" of an infinite cost and constraint horizon. However the resulting stability domain is in general smaller than in the infinite case since the end-constraint enforces, as mentioned earlier, a much more aggressive behavior. This can be cumbersome for small control horizons N^C or hard constraint inputs.

2.2.4 Linear MPC with Infinite Prediction Horizon

Rawlings and Muske [54] introduced a different approach which guarantees stability while removing some of the encountered problems. They use a finite control horizon N^C but in contrast to the previous approaches they keep the infinite prediction horizon N^P. The input parameterization stays the same as in section 2.2.3, $u_{j|k} = 0,\ j \geq (k + N^C)$. To guarantee convergence of the remaining infinite part of the cost function:

$$\Phi_{N^C/\infty}(x_k, \pi_k) = \sum_{j=0}^{N^C-1} u_{k|j}^T R u_{j|k} + x_{j|k}^T Q x_{j|k} + \sum_{j=N^C}^{\infty} x_{j|k}^T Q x_{j|k} \tag{2.29}$$

Rawlings and Muske concluded that the unstable modes of the system must be zero at the end of the control horizon. The remaining part of the sum would therefor be restricted to the free evolution of the stable modes and would converge. Furthermore this sum

$$\sum_{j=N^C}^{\infty} x_{j|k}^T Q x_{j|k} = x_{k+N^C|k}^T \tilde{Q} x_{k+N^C|k} \tag{2.30}$$

can be expressed as an quadratic weight on the state at the end of the control horizon. $\tilde{Q}$ is the positive definite solution of the Lyapunov equation $\tilde{Q} = Q + A_s^T \tilde{Q} A_s$. Here A_s is the "stable" part of A resulting from a Schur decomposition of A [41]:

$$\begin{aligned} U^T A U &= T && (2.31)\\ A &= \begin{bmatrix} U_s & U_u \end{bmatrix} \begin{bmatrix} T_{11} & T_{12} \\ 0 & T_{22} \end{bmatrix} \begin{bmatrix} U_s^T \\ U_u^T \end{bmatrix} && (2.32)\\ & && (2.33) \end{aligned}$$

A_s gets

$$A_s = U_s T_{11} U_s^T \tag{2.34}$$

T_{11} contains the stable eigenvalues of T whereas T_{12} contains the unstable. With the help of this decomposition we can express the constraint that the unstable modes of x must be 0 at the end of the control horizon by $U_u^T x_{k+N^C|k} = 0$. $U_u^T x_{k+N^C|k}$ represents a projection of $x_{k+N^C|k}$ on the unstable subspace. The resulting controller can be summarized as follows:

$$\min_{\pi_k} \sum_{j=0}^{N^C-1} u_{k|j}^T R u_{j|K} + x_{j|k}^T Q x_{j|k} + x_{k+N^C|k}^T \tilde{Q} x_{k+N^C|k} \tag{2.35}$$

s.t

$$U_u^T x_{k+N^C|k} = 0 \tag{2.36}$$

$$u_{j|k} \in \mathcal{U} \quad j = k, \ldots, k + N^C - 1 \tag{2.37}$$

$$x_{j|k} \in \mathcal{X} \quad j = k + 1, \ldots, \infty \tag{2.38}$$

This controller is not implementable in the given form.

First the resulting optimization problem is not solvable in the proposed form. We have a finite number of decision variables u but a infinite number of constraints once state constraints are present. Rawlings and Muske [54] showed that there exists a finite constraint horizon so that all active state constraints are contained. Additionally they provide an upper bound for this horizon for the case that A has distinct eigenvalues. This bound is generally much larger than the minimum horizon necessary and it is given in terms of the state $x_{k+N|k}$, where N is the constraint horizon. The result of this is that we cannot calculate the horizon a priori. Instead we have to solve a sequence of QP's with increasing constraint horizon (but fixed control horizon=decision variables) until we can verify that our chosen horizon is larger or equal to this upper bound. Another approach which makes an off-line calculation possible, but does not work for every case is Gilbert and Tan's theory of maximal output admissible sets [17]. They show that in special cases the infinite set of constraints for linear systems can be reduced to a finite set. They provide an off-line algorithm that determines a finite horizon which is equivalent to the infinite one.

The second problem is the feasibility of the state constraints. If we have no constraints on the inputs and outputs, the resulting optimization problem is feasible. The addition of state constraints can make this problem infeasible, so that a relaxation or softening is necessary. There are different ways to achieve this. We could remove the state constraints one by one starting with $j = 1$ until the problem becomes feasible [54]. Another approach would be to soften these infeasible constraints over the entire horizon. A good overview of the resulting problems and possible solutions can be found in [59].

Under the assumption of initial feasibility, Muske [44] has shown that the resulting controller is Lipschitz continuous. This is especially important if we considers the stability of a combination of this controller and a state estimator (see section 2.2.6).

2.2.5 The Constrained Linear Quadratic Regulator

A logical expansion of the linear MPC controller with infinite prediction horizon was proposed by Sznaier and Damborg [66], Chmielewski and Manousiouthakis [9] and Scokaert and Rawlings [57].

The controller of section 2.2.4 guarantees stability once a feasible solution is found. However the limitation of a finite control horizon leads to a mismatch between open and closed loop behavior. The controller will be suboptimal compared to the "ideal" LQR with constraints. The solution of the matrix Lyapunov equation makes it possible to calculate the cost of the uncontrolled system after the control horizon, but it is necessary to force the unstable modes to zero in finite time and freeze the inputs hereafter. This leads to a discrepancy in the performance compared to the LQR.

A key step to achieving a solution to the infinite constrained LQR is the following modification to problem (2.35)-(2.38). We do not fix the input after the control horizon "N^C", instead we apply a linear control law $u_{j|k} = -Kx_{j|k},\ j \geq N^C$ where K is the solution of the unconstrained LQR (2.22). The remainder of the infinite sum takes now the form:

$$\sum_{j=N^C}^{\infty} x_{j|k}^T Q x_{j|k} + x_{j|k}^T K^T R K x_{j|k} = x_{k+N^C|k}^T \tilde{Q} x_{k+N^C|k} \tag{2.39}$$

$$\tag{2.40}$$

$\tilde{Q}$ is given by the solution to the following Lyapunov equation

$$\tilde{Q} = Q + (A - BK)^T \tilde{Q}(A - BK) + K^T R K \tag{2.41}$$

The new controller is similar to (2.35)-(2.38) except that the constraints on the unstable modes $U_u^T x_{k+N^C|k} = 0$ have been removed.

$$\min_{\pi_k} \sum_{j=0}^{N-1} u_{k|j}^T R u_{j|K} + x_{j|k}^T Q x_{j|k} + x_{k+N|k}^T \tilde{Q} x_{k+N|k} \tag{2.42}$$

s.t

$$u_{j|k} \in \mathcal{U} \quad j = k, \dots, k+N-1 \tag{2.43}$$

$$x_{j|k} \in \mathcal{X} \quad j = k+1, \dots, N \tag{2.44}$$

It has been shown that there exists a horizon N so that the solution to this problem and the constrained LQR

$$\min_{\substack{\pi_k \\ u_{j|k}\in\mathcal{U} \\ x_{j|k}\in\mathcal{X}}} \Phi(x_k, \pi_k) \tag{2.45}$$

$$\Phi_\infty(x_k, \pi_k) = \sum_{j=0}^{\infty} x_{j|k}^T Q x_{j|k} + u_{k|j}^T R u_{j|k} \tag{2.46}$$

coincide ([9],[57]).
Scokaert and Rawlings [58] propose the following algorithm to identify the optimal control profile in finite time:

Constrained LQR Algorithm

STEP 0:

•Choose a finite horizon N

STEP 1:

•Solve Problem (2.42)-(2.44)
IF $x_{k+N|k} \in \mathbb{X}_K$
GOTO STEP 2
ENDIF
•Increase N
GOTO STEP 1

STEP 2:

•Terminate: $\pi_k^* = \pi_k$

Here $\mathbb{X}_K$ is the positive invariant set for which the linear system with the LQR law $u_k = -Kx_k$ satisfies all the state and input constraints over an infinite horizon.

$$\begin{aligned} \mathbb{X}_K = \{x_k \in \mathbb{R}^n \quad : \quad & x_{j+1} = (A - BK)x_j,\ u_j = -Kx_j, \\ & G^u u_j \leq g^u,\ G^x x_j \leq g^x,\ \forall j \geq k\} \end{aligned} \tag{2.47}$$

It is important to realize, that the proposed algorithm solves the infinite horizon constrained LQR problem in finite time while leading to a solution with optimal performance.

2.2.6 Linear MPC, Concluding Remarks

In the preceding sections we have outline the major results for linear MPC. We have seen that the constrained linear quadratic Regulator removes the difference between

closed and open loop performance for the nominal case. The solution involves a series of QP's with increasing constraint horizon. This often leads to a drastic increase in the computational time. The usage of interior point methods for the resulting QP under consideration of the special structure of the optimization problem could lead to efficient solution methods ([69],[52]).
We only considered the case that the complete state information is available at every control step. Scokaert et al [60] have shown that the in (2.2.4, 2.2.5) presented controllers in connection with an stable state estimator (usually a Kalman Filter) give a stable system. These results were derived by considering the estimated states as a decaying perturbation for the controlled system.
We have not included a presentation of the important issue of robustness with respect to model uncertainties and disturbances since the research in this area is still in early stages ([31],[73],[72]). A recently presented concept [4] seems to be very promising, especially since it can be expanded to the nonlinear case.

2.3 Nonlinear MPC, Existing Concepts

During the last section we only used discrete time models since most of the existing work was done in discrete time. One reason for this might be that a exact solution for the continuous time linear state equation is available, so that a change from continuous to discrete time without loss of generality is possible (at least for the case with no state constraints). Under the assumption of a fixed sampling time the resulting equations are still linear and time invariant. In the progress of the following sections we are forced to use both time representations, since parts of the existing literature consider continuous time, whereas other parts use discrete time or continuous time with a time discrete input.

We divide the existing control concepts in the following four groups, depending on how closed loop stability is achieved:

Controllers for which the Cost Function is a Lyapunov Function
The first group contains algorithms for which the stability can be shown via the stated Lyapunov theorem (1.1). For all algorithms in this group the Lyapunov function becomes the cost or objective function of the optimization. The major differences in this group are the length of the horizon and the consideration of an optimal or suboptimal solution:

1. Infinite Horizon MPC

2. Finite Horizon MPC without End Constraint

3. Finite Horizon MPC with End Constraint

4. Suboptimal MPC with End Constraint

Stability by Switching to a Stabilizing Controller
The algorithms of this group achieve stability by switching to a user supplied stabilizing controller once they have reached a pre specified region in the state space. We present the following two approaches:

1. Dual Mode Control

2. Suboptimal Dual Mode Control

Enforced Stability by Contraction
Contractive MPC algorithms guarantee stability by requiring that the state has to lay in a time contracting region after one control period. The phrase control period stands for the horizon length T or N.

Stability based on Results of Nonlinear Geometrical Control Theory
These control concepts derive their stability from results in geometric control theory. ([21], [46]).We only present the following two concepts without going into details:

1. Combined Feedback Linearization and Linear MPC

2. Shortest Prediction Horizon Nonlinear MPC and Nonlinear Internal Model Control (NIMC)

2.3.1 The Cost Function as Lyapunov Function

In this section we consider nonlinear MPC strategies which base their stability on theorem 1.1 or its modifications. The cost function becomes the Lyapunov function. The results are taken from papers by Keerthi and Gilbert [26], Mayne and Michalska [38], Mayne [37], Rawlings et al. [53] and Scokaert et al. [56]. We can achieve stability via an infinite horizon feedback law or via a finite horizon feedback law with endpoint constraint. However, since the system equations and the cost function are nonlinear, additional conditions on the cost function and the constraints are required to achieve stability.

We use the discrete time MPC formulations (2.10)-(2.16) to keep the resulting derivations short and clean. However some of the major contributions in this area [38] consider continuous time systems with a variable horizon length. Most of the results are directly convertible, however the stability proof for continuous time is more cumbersome, because we have to consider not only the states at the sampling times. We return to this formulation in section 2.3.2.

In the following the prediction and control horizon coincide. Our objective function is given by:

$$\Phi_N(x_k;\pi_k) := \sum_{j=k}^{k+N-1} L(x_{j|k}, u_{j|k}) \tag{2.48}$$

We explicitly allow the horizon to be infinite in the limiting case, $N \to \infty$. The NLP takes the following form:

$$\min_{\pi_k} \Phi_N(x_k;\pi_k) \tag{2.49}$$

s.t.

$$\begin{aligned}
x_{j+1|k} &= f(x_{j|k}, u_{j|k}) && (2.50)\\
x_{k|k} &:= x_k && (2.51)\\
x_{k+N|k} &= 0 && (2.52)\\
u_{j|k} \in\ & \mathcal{U} \qquad k = k, \dots, k+N-1 && (2.53)\\
x_{j|k} \in\ & \mathcal{X} \qquad k = k+1, \dots, k+N-1 && (2.54)
\end{aligned}$$

The zero end-constraint (2.52) is included to enforce stability for the finite horizon case. We have no final state penalty since at the end of the horizon x and u are zero. Keerthi and Gilbert [25] derive conditions for x_k, f, L, $\mathcal{U}$, $\mathcal{X}$ so that a solution to problem (2.49) exists. We have to limit the choice of L and demand that the sets $\mathcal{U}$ and $\mathcal{X}$ are closed and contain the origin. The following result is taken from [53]. A more general result can be found in [25].

Theorem 2.1 (Existence of a optimal solution) *If L is continuous, $L(0,u) \to \infty$ as $\|u\| \to \infty$, and $\mathcal{U}$ and $\mathcal{X}$ are closed, then the existence of a feasible hypothetical control sequence $\pi_k = \{u_{k|k}, u_{k+1|k}, \dots, u_{k+N-1|k}\}$ that yields a bounded open-loop objective function $\Phi(x_k;\pi_k)$ for a bounded initial condition x_k implies the existence of an bounded optimal control sequence $\pi_k^\star$ for this initial condition. This results holds for either finite or infinite horizon problems.*

The proof of stability for finite and infinite horizon with end-point constraint follows directly from the Lyapunov theorem. The application of this theorem, requires the following additional assumptions on the stage cost L.

- A1 $L(0,0) = 0$, and L continuous at the origin.
- A2 $\exists\, K^+$-function $\gamma[0,\infty) \to [0,\infty)$ with $0 < \gamma(\|(x,u)\|) \le L(x,u) \quad \forall (x,u) \neq (0,0)$
- A3 $\Phi_N{}^\star(x)$ is continuous at the origin x=0

Remarks: Assumptions A1 and A2 guarantee the following properties of $\Phi^\star$:

1. $\Phi^\star_N(x) \geq 0$
2. $\Phi^\star_N = 0 \Leftrightarrow x = 0$
3. $\gamma(\|x\|) \leq \Phi^\star_N(x)$
4. $\Phi^\star_{N+k}(x) \leq \Phi^\star_N(x) \; \forall k \geq 0$

We need these to satisfy requirement 3 and 1 in Theorem 1.1, which will guarantee the convergence of the resulting optimal cost function to zero. Assumption 3 is the key for stability in the finite and infinite horizon case. It allows us to formulate a continuous upper bound for the optimal cost near the origin, as required in Theorem 1.1 part 1. $\triangle$

Under these assumptions Meadows [39] shows that the following holds:

Theorem 2.2 *If property A1-A3 are satisfied, then the origin is an asymptotically stable equilibrium point for the dynamic system $x_k = f(x_k, u^\star_k(x_k))$ with a region of attraction consisting of those points in $\mathbb{R}^n$ for which a solution to the nonlinear program (2.49) exists.*

Proof: Meadows divides the proof in two parts. First he shows, under the usage of A1 and A2 and the resulting property on $\Phi^\star$, that the value of the costfunction converges to zero. This implies that the sequence of $(x_k, u_k) = (x_k, u^\star_k(x_k))$ converges to zero. However since convergence does not imply stability he proceeds in the second part of his proof with an Lyapunov like argument that uses A3 as a key argument to show stability. Stability combined with convergence implies asymptotic stability.

Instead of using this result we apply Theorem 1.1 with $\Phi^\star$ as our Lyapunov function. The requirement that $\Phi^\star$ has to be upper bounded by a K^+ function in a region at the origin is automatically satisfied, since we assumed continuity of $\Phi^\star$ at the origin. It is easy to see that $\Phi^\star$ is lower bounded by $\gamma(\|x_k\|)$. What remains is that $\Phi^\star$ has to satisfy requirement 3, namely that the forward difference $\Phi^\star(x_k) - \Phi^\star(x_{k+1})$ is greater or equal than zero and that the cost function decreases for all $M \in \mathbb{N}^+$ (M finite) steps by a value greater than 0. To show this we proceed similar to step one in Meadows [39] proof.

Infinite horizon. This part bases on the fact that the remaining part of the optimal input sequence $\pi^\star_\infty$ after applying the first step is feasible at the next step for the nominal case. We get:

$$\begin{aligned}
\Phi_\infty^\star(x_k) &= L(x_k, u_{k|k}^\star) + L(x_{k+1|k}^\star, u_{k+1|k}^\star) + \ldots \\
&= L(x_k, u_k^\star) + \Phi_\infty(f(x_k, u_k^\star), \tilde{\pi}) \\
&= L(x_k, u_k^\star) + \Phi_\infty(x_{k+1}, \tilde{\pi}) \\
&\geq L(x_k, u_k^\star) + \Phi_\infty^\star(x_{k+1})
\end{aligned}$$

The last step follows from the principle of optimality and the fact that $\tilde{\pi} = \{u_{k+1|k}^\star, u_{k+2|k}^\star, \ldots\}$. This shows that $\Phi_\infty^\star$ decreases by at least $0 < \gamma(\|x_k\|) \leq L(x_k, u_k^\star)$ for every step which satisfies requirement 3:

$$\Phi_\infty^\star(x_k) - \Phi_\infty^\star(x_{k+1}) \leq \gamma(\|x_k\|) \tag{2.55}$$

Finite horizon. For the finite horizon we first have to find a new feasible sequence $\tilde{\pi}$. The fact that we required $x_{k+N|k} = 0$ helps us herewith. With $\tilde{\pi} = \{u_{k+1|k}^\star, , \ldots\ u_{k+N-1|k}, 0\}$ we get:

$$\begin{aligned}
\Phi_N^\star(x_k) &= L(x_k, u_{k|k}^\star) + \ldots + L(x_{k+N-1}^\star, u_{k+N-1|k}^\star) \\
&= L(x_k, u_k^\star) + \ldots + L(x_{k+N-1}^\star, u_{k+N-1|k}^\star) + L(0,0) \\
&= L(x_k, u_k^\star) + \tilde{\Phi}_N(x_{k+1}) \\
&\geq L(x_k, u_k^\star) + \Phi_N^\star(x_{k+1})
\end{aligned}$$

From this and Theorem 1.1 we can deduce Theorem 2.2. □

The resulting closed and open loop controller is illustrated in Figure 2.3.

Keerthi and Gilbert [25] use a controllability assumption instead of the given continuity assumption (3) on the optimal cost. However their assumption seems to be even more difficult to check. Meadows et. al [40] show that every local feedback linearizable system satisfies A3 if we choose the horizon length N greater or equal than the maximum controllability indices of the system. Hence they conclude, that every system with this property can be locally asymptotically stabilizable by applying the stated controller (provided that a feasible input sequence exists). Furthermore they show that there exist controllable nonlinear discrete-time systems which cannot be stabilized with continuous feedback laws, but for which receding horizon control can provide a asymptotically stabilizing feedback. This shows additionally that receding horizon control can lead to discontinuous feedback laws.

Mayne and Michalska [38] consider the continuous time counterpart of the given algorithm. To ensure stability they need strong assumptions, which they relaxed in a later paper [35]. Mayne and Michalska ([43],[37]) also proposed a variation of this concept, using a flexible horizon T_i. By permitting the horizon to be an additional

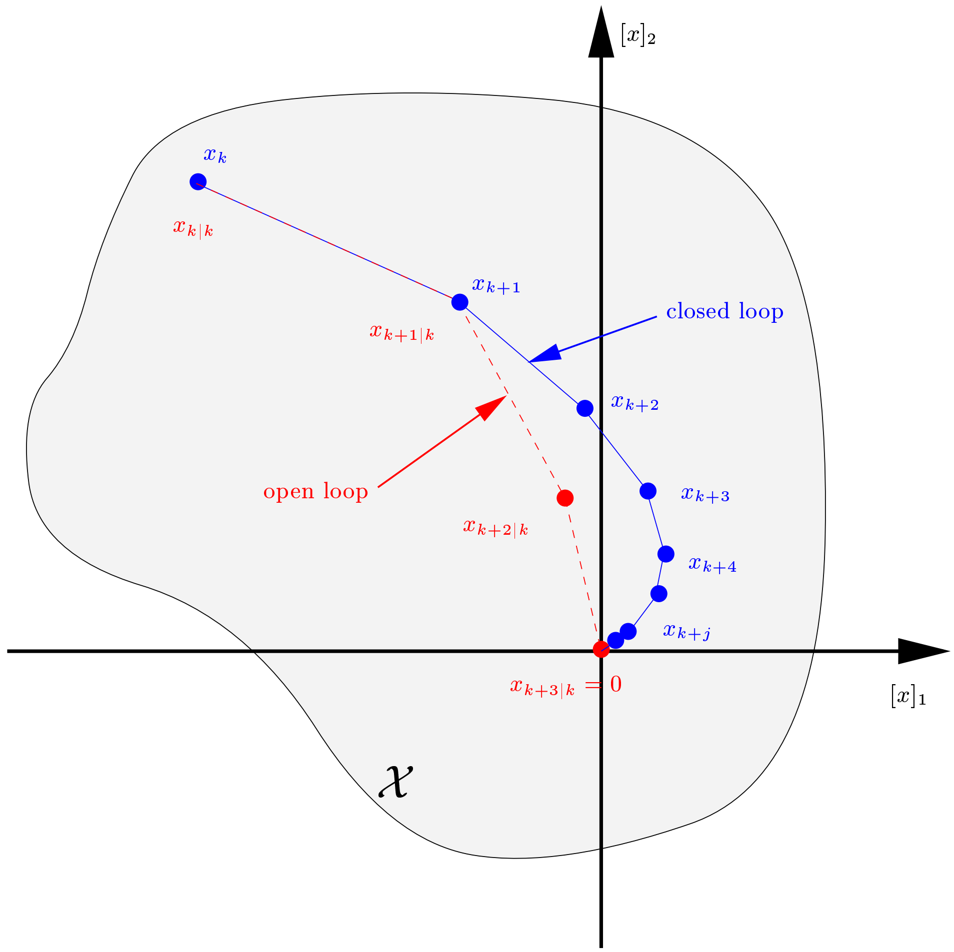

Figure 2.3: Closed and open loop performance of the finite horizon controller with final state constraint, $N = 3$

flexible decision variable in the optimization many difficulties can be overcome. For example an increase or decrease of the horizon length could allow faster calculations or could make feasibility possible. A simple formulation of this variable horizon MPC for continuous time with $\bar{\Phi}$ as the time invariant version of (2.7) is:

$$\min_{\substack{\bar{u}(\tau|t)\ \tau\in[t,t+T] \\ T\in[0,T_{\max}]}} \bar{\Phi}(\bar{x}(t); \bar{u}(\tau|t))$$

s.t.

$$\begin{aligned}
\dot{\bar{x}}(\tau|t) &= \bar{f}(\bar{x}(\tau|t), \bar{u}(\tau|t)) \quad \tau \in [t, t+T] \\
\bar{x}(t|t) &:= \bar{x}(t) \\
\bar{x}(t+T|t) &= 0 \\
\bar{u}(\tau|t) &\in \mathcal{U} \subset \mathbb{R}^m \\
\bar{x}(\tau|t) &\in \mathcal{X} \subset \mathbb{R}^n
\end{aligned}$$

The above stated MPC algorithm requires the exact global solution of a non-convex nonlinear program. Implementing this would mean to provide the solution on-line at every sample time. The stability relies on the global minima and the enforcement of an additional stability constraint, which is an equality constraint.
Summarized we have the following problems:

- Exact satisfaction of equality constraints is in general not possible in finite computational time. However, if we would interrupt the optimization before we converge we could not ensure that this constraint is satisfied. A solution for this problem is the application of a Dual-Mode or Hybrid controller, which we will present in section 2.3.2.
  Remark: The state equations do not lead to the same problem, since we can substitute them in the cost function and other constraints. It is only the final state equality constraint which leads to this problem.

- Finding the global solution of a non convex, nonlinear program is, in general computationally prohibitive.

This makes the presented concepts inapplicable. Scokaert et. al. [56] derive a sub-optimal MPC strategy which overcomes the second problem. They replace the given algorithm by an algorithm that does not require a global optimization.

Suboptimal MPC with Terminal Constraint Algorithm

STEP 0: $k = 0$

- find a open loop control sequence, $\pi_N = \{u_{0,0}, \dots, u_{N-1|0}\}$ which satisfies (2.50)- (2.54).
- set $u_0 = u_{0|0}$

STEP 1: $k > 0$, state x_k

- find a control sequence π_k which satisfies (2.50)-(2.54) and $\Phi_N(x_k, \pi_k) \leq \Phi_N(x_{k-1}, \pi_{k-1}) - \mu L(x_{k-1}, u_{k-1})$, using $\tilde{\pi} = \{u_{k|k-1}, \dots, u_{k+N-2|k}, 0\}$ as an initial guess.
- set $u_k = u_{k|k}$.

GOTO STEP 1

Remarks: The factor $\mu \in (0, 1]$ was introduced for the disturbance case. For simplicity we first consider the undisturbed case $\mu = 1$. Under these conditions $\tilde{\pi}$ delivers a feasible input sequence, with a decreasing cost/Lyapunov function, as required in theorem 1.1. To guarantee the continuous upper bound of the cost function at the origin, they assume that every realization of π_k with x_k in a (possibly small) ball containing the origin is upper bounded by a (continuous) K-function of the state.

$$\|\pi_k\| \leq \sigma(\|x_k\|) \quad \forall x_k \in B_r,\ r > 0 \tag{2.56}$$

They derive a special Lyapunov theorem to prove stability of the controlled system under the given conditions. However we can also use the given Lyapunov theorem , since (2.56) and the continuity of L and f lead to continuous upper bounded cost-function Φ_N at the origin.
Note that even a $\mu < 1$ leads to a decreasing cost function which implies convergence of (x_k, u_k). For the model mismatch case the x_{k+1} resulting from u_k might be different than the precalculated $x_{x+1|k}$. Then we have to calculate a new π_k to reduce the cost and can take $\tilde{\pi}$ only as an initial guess. It is clear that smaller values of μ make this easier to achieve, by this simplifying the search for a new feasible sequence π. $\triangle$

Finally we should remark that Alamir and Bonard [1] considered the case where prediction horizon length and control horizon length are different and no final state constraint $x_{k+N|k}$ is present. They were able to show that under additional conditions, a *finite* prediction horizon N^P for fixed control horizon N^C exists, so that the resulting optimal controller is asymptotically stable. A algorithm to calculate this prediction horizon is in the moment not available.

2.3.2 Stability due to Switching to a Stabilizing Controller

In the previous section we have seen that an infinite horizon with continuous optimal cost function at the origin or a finite horizon plus zero end constraint can lead to a stabilizing controller. However the infinite horizon controller leads to an unsolvable optimization problem. In the finite horizon case the optimization problem is solvable, but the requirement to find a global minima and to satisfy the zero or terminal state constraint leads to an in practice infeasible problem. The presented suboptimal controller dismisses the globality of the solution, however the problem of finding a feasible solution while satisfying the terminal state constraint remains. Mayne and Michalska proposed in [43] a possible solution to this problem. They considered continuous time models and removed the terminal state constraint by a terminal state inequality. They allow the last state in the horizon to lay in an region $\mathcal{B}$ instead of the fixed "value" 0. They achieve stability by switching to a user supplied stabilizing control law which keeps the system in $\mathcal{B}$ once the state has entered $\mathcal{B}$. This strategy gives the resulting controller the name dual mode or hybrid controller [43]. This is a new way to enforce stability, since we no longer use the cost function as an Lyapunov function.They show that the MPC part of the controller is able to force the controlled system in finite time into $\mathcal{B}$. To employ the resulting concept the following must hold (to see a complete list of the necessary assumptions and requirements, see [43]):

A1 $\exists$ a local stabilizing controller $\bar{h}(\cdot) : \mathcal{B} \to \mathcal{U}$ with the set $\mathcal{B} \subset \mathcal{X}$ so, that $\mathcal{B}$ is a positive invariant set for the controlled system $\dot{\bar{x}} = \bar{f}(\bar{x}, \bar{h}(\bar{x}))$.

To state the controller we need the following equations:

$$\dot{\bar{x}}(\tau|t) = \bar{f}(\bar{x}(\tau|t), \bar{u}(\tau|t)) \quad \tau \in [t, t+T] \tag{2.57}$$

$$\bar{x}(t|t) := \bar{x}(t) \tag{2.58}$$

$$\bar{x}(t+T|t) \in \mathcal{B} \tag{2.59}$$

$$\bar{u}(\tau|t) \in \mathcal{U} \subset \mathbb{R}^m \tag{2.60}$$

$$\bar{x}(\tau|t) \in \mathcal{X} \subset \mathbb{R}^n \tag{2.61}$$

Mayne and Michalska also consider the prediction horizon T as "optimization" variable.

$$\bar{\Phi}(\bar{x}(t); \bar{u}(\tau|t), T) := \int_{\tau=t}^{t+T} \bar{L}(x(\tau|t), u(\tau|t))d\tau \tag{2.62}$$

We present a simplified version as presented in [36], the original version can be found in [43]. ΔT means the sampling time, t_i stands for the time at the sampling point i, $t_i := t_0 + i\Delta T$. T_i means the horizon at step i, $T_{\max}$ will be the maximal horizon length and $\mu \in (0,1)$ will be a fixed factor.

Dual Mode Control Algorithm

STEP 0: Initialization, time t_0,

IF $\bar{x}(t_0) \in \mathcal{B}$

GOTO STEP 2

ELSE

- compute a feasible control horizon pair $(\bar{u}(\tau|t_0), T_0)$ with $T_0 \in [0, T_{\max}]$ for (2.57)-(2.61).
- apply $\bar{u}(\tau|t_0)$ $\tau \in [t_0, t_0 + \Delta T]$

STEP 1: time $t_i > t_0$

IF $\bar{x}(t_i) \in \mathcal{B}$

GOTO STEP 2

ELSE

- obtain a admissible control horizon pair $(\tilde{\bar{u}}(\tau|t_i), \tilde{T}_i)$ with $\tilde{T}_i \in [0, T_{i-1} - \mu\Delta T]$ for (2.57)-(2.61).
- determine a feasible control horizon pair $(\bar{u}(\tau|t_0), T_0)$ with $T_i \in [0, \tilde{T}_i]$ for (2.57)-(2.61) and additional $\bar{\Phi}(\bar{x}(t_i); \bar{u}(\tau|t_i), T_i) \leq \bar{\Phi}(\bar{x}(t_i); \tilde{\bar{u}}(\tau|t_i), \tilde{T}_i)$
- apply $\bar{u}(\tau|t_i)$ $\tau \in [t_i, t_i + \Delta T]$

GOTO STEP 1

STEP 2: time $t_i > t_0$, $\bar{x}(t_i) \in \mathcal{B}$

- implement $\bar{u}(\bar{x}) = h(\bar{x})$

Remarks: STEP 1 calculates twice an admissible control sequence. This is only necessary in the disturbance case. The first calculation provides a feasible input $u(\tau|t_i)$ and horizon T_i which is then improved. This means that the resulting controller does not have to optimize over the cost function space. A decrease in the cost function like in the suboptimal case of section 2.3.1 is sufficient for stability. For the nominal case and $\mu = 1$, $\tilde{u}(\tau|t_i)$ could be the remainder of the previous input $u(\tau|t_{i-1})$. Equation (2.59) warrants that the last state in the prediction is in $\mathcal{B}$ and ensures with $\bar{\Phi}(\bar{x}(t_i); \bar{u}(\tau|t_i), T_i) \leq \bar{\Phi}(\bar{x}(t_i); \tilde{\bar{u}}(\tau|t_i), \tilde{T}_i)$ that the state reaches $\mathcal{B}$ after a maximal time of T_0/μ. The horizon shrinks from control step to control step by minimal $\mu\Delta T$. $\triangle$

For a graphical interpretation of this concept see Figure 2.4. Michalska and Mayne

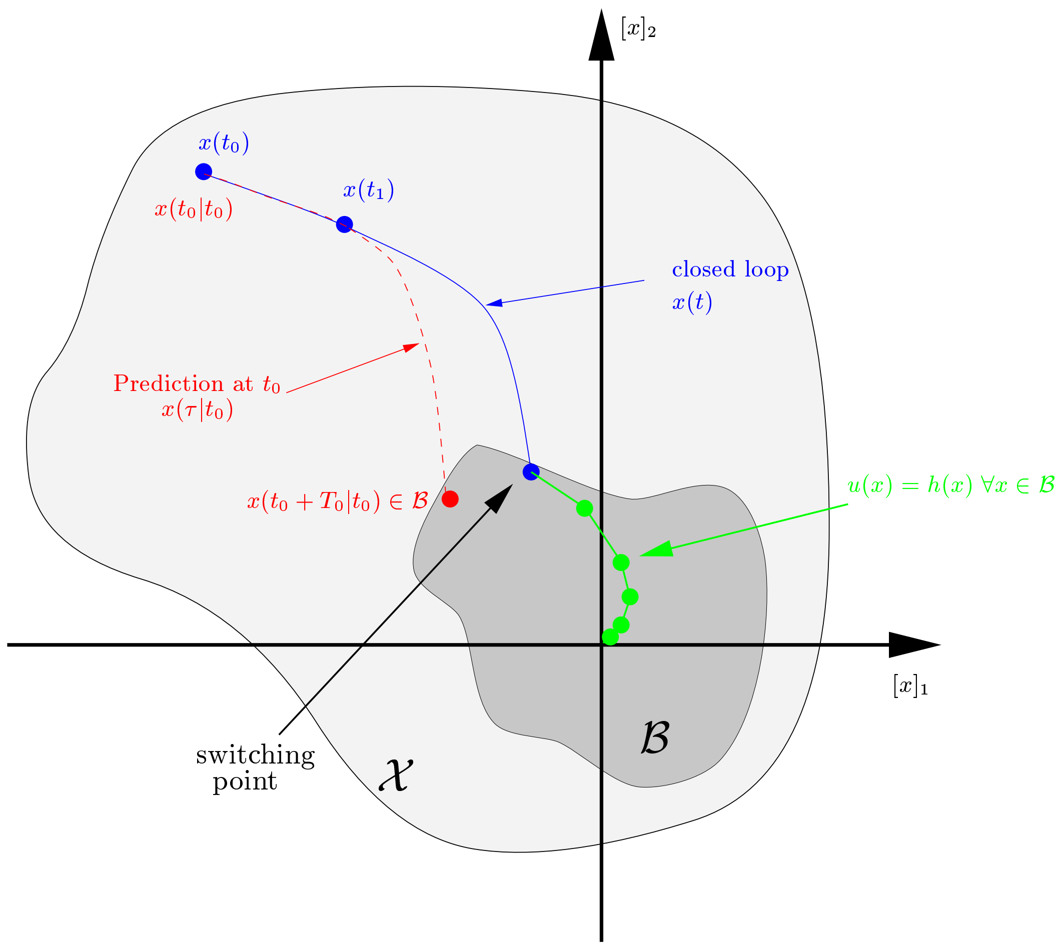

Figure 2.4: Closed and open loop behavior of dual mode control

[43] state a robust version of the given controller. To guarantee robustness they use a modified terminal set. This set is a subset of $\mathcal{B}$ in which the given controller h can

stabilize the system. The "new" terminal inequality constraint becomes:

$$\bar{x}(t+T|t) \in \mathcal{B}' \subset \mathcal{B}$$

The presence of modelling errors leads to a discrepancy between the calculated hypothetical final state $\bar{x}(t_i + T_i|t_i)$ and the final state of the plant. If the discrepancy between model and plant is not too big, then the state $\bar{x}^P(t_i + T_i|t_i)$ is in $\mathcal{B}$ and we can guarantee stability. This situation is shown in Figure 2.5. Scokaert et al. [56]

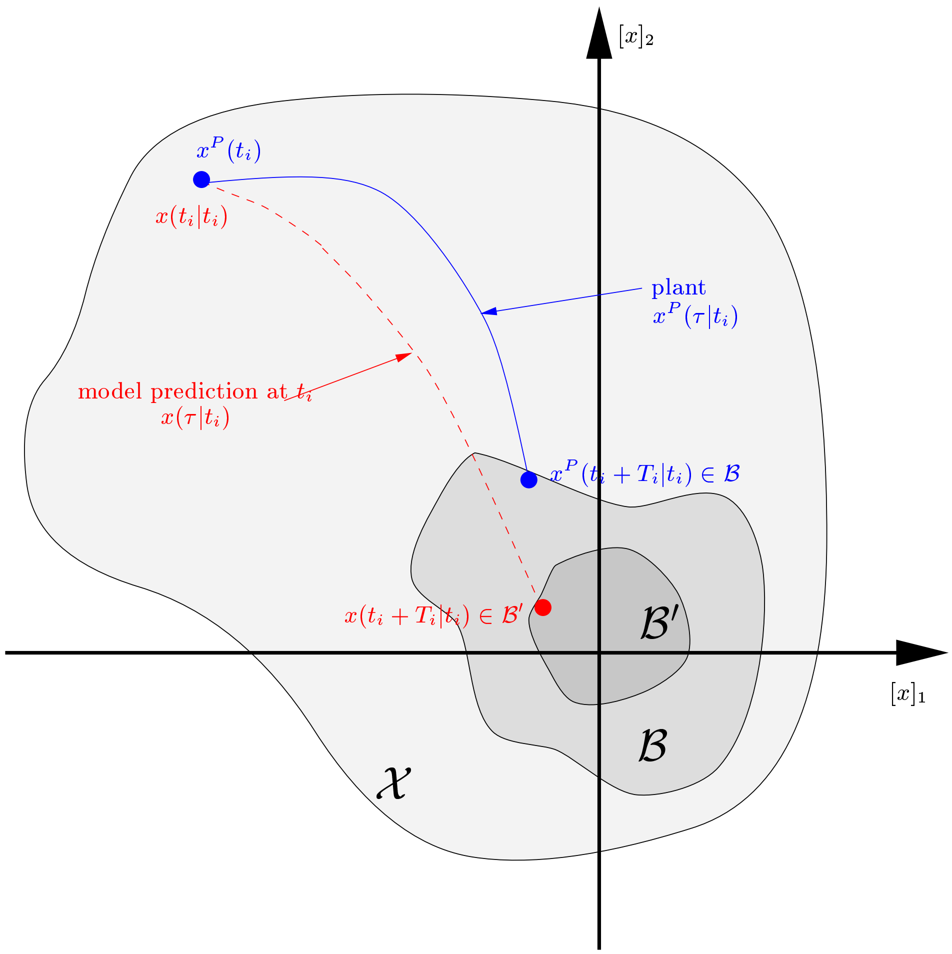

Figure 2.5: Robust version of dual mode control

present an optimal and suboptimal discrete time version of the given concept. Since they do not consider a variable horizon, they have to modify the discrete stage cost function:

$$L(x_k, h(x_k)) = 0 \; \forall x_k \in \mathcal{B}$$

Here $h(\cdot)$ is, similar to the continuous time version, a local stabilizing controller, so that the set $\mathcal{B}$ is positive invariant for the controlled system $x_k = f(x_k, h(x_k))$. They derive the following algorithm with $\mu \in (0, 1]$ as an constant factor which improves the robustness against model errors and disturbances similar to the suboptimal MPC with terminal end constraint.

Suboptimal Dual Mode Control Algorithm

STEP 0: Initialization, time $k = 0$, x_0

IF $x_0 \in \mathcal{B}$

GOTO STEP 2

ELSE

- compute a feasible input sequence $\pi_0 = \{u_{0|0}, \dots, u_{N-1|0}\}$ which satisfies (2.50), (2.51), (2.53), (2.54) and $x_{N|0} \in \mathcal{B}$
- apply $u_0 = u_{0|0}$

STEP 1: time $k > 0$

IF $x_k \in \mathcal{B}$

GOTO STEP 2

ELSE

- compute a feasible input sequence π_k which satisfies (2.50), (2.51), (2.53), (2.54), $x_{k+N|k} \in \mathcal{B}$ and $\Phi(x_k, \pi_k) \leq \Phi(x_{k-1}, \pi_{k-1}) - \mu L(x_{k-1}, u_{k-1})$ use $\tilde{\pi}_k = \{u_{k|k-1}, \dots, u_{k+N-2|k-1}, h(x_{k+N-1|k-1})\}$ as an initial guess
- apply $u_k = u_{k|k}$

GOTO STEP 1

STEP 2: time $k > 0$, $x_k \in \mathcal{B}$

- implement $u_k(x_k) = h(x_k)$

Remarks: For the optimal case $\Phi(x_k, \pi_k) \leq \Phi(x_{k-1}, \pi_{k-1}) - \mu L(x_{k-1}, u_{k-1})$ is replaced by $\min_{\pi_k} \Phi_N(x_k; \pi_k)$ △

The major disadvantage of the Dual Mode concept is the necessity to switch to a stabilizing controller once the system has reached $\mathcal{B}$. In most cases it is problematic to provide h and give a good estimate of the region $\mathcal{B}$. In addition the switching might lead to a dramatic change in the “behavior” of the controller once x_k has reached B, see Figure 2.4.

2.3.3 Enforced Stability via Contraction

A different MPC approach was proposed by Yang and Polak [71] resting upon an earlier work of Polak and Mayne [50]. Oliveira et al. [13] also proposed a similar concept. There stability does not depend on the cost function as Lyapunov function or switching to an stabilizing controller. We present the algorithm given in [13]. The original version considers a continuous time system with time discrete inputs $\bar{u}_k$. We restrict ourself for simplicity to discrete time. In the following $\mu \in [0, 1)$ is the contraction factor, N is the control horizon and k^C is over the interval T fixed position of the contraction constraint. P is be a positive definite matrix and x^C denotes a previous state x

Discrete Time Contractive MPC Algorithm

STEP 0: Initialization, time $k = 0$

- set $k^c = N$, $x^C = x_0$
- solve the optimization problem (2.49)-(2.54) where (2.52) is replaced with the contractive constraint $\|x_{k^C|0}\|_P \leq \mu \|x^C\|_P$
  this leads to the feasible sequence $\pi_0 = \{u_{0|0}, \dots , u_{N-1|0}\}$
- apply $u_0 = u_{0|0}$

STEP 1: time $k > 0$

IF $k = k^C$

- set $k^C = k + N$, $x^C = x_k$

END IF

- solve the optimization problem (2.49)-(2.54) where (2.52) is replaced with the contractive constraint $\|x_{k^C|k}\|_P \leq \mu \|x^C\|_P$
  this leads to the feasible sequence $\pi_k = \{u_{k|k}, \dots , u_{k+N-1|k}\}$
- apply $u_k = u_{k|k}$ GOTO STEP 1

Remarks: The position of the contractive constraint $\|x_{k+k^C|k}\|_P \leq \mu \|x^C\|_P$ and the value of x^C is only updated all N steps. This leads to the fact, that every Nth step is in the ball $\mu \|x^C\|_P$. This fact is shown in Figure 2.6 and Figure 2.7 $\triangle$

For a complete derivation for continuous time systems and the necessary assumptions see [13].

2.3.4 Stability based on Results of the Geometric Control Theory

The last class of MPC algorithms derives their stability from geometrical control theory. We can identify two mainstreams.

- Combined feedback linearization and “linear” MPC
- MPC algorithms with no terminal state constraint and a control horizon of one, shortest prediction horizon and NIMC

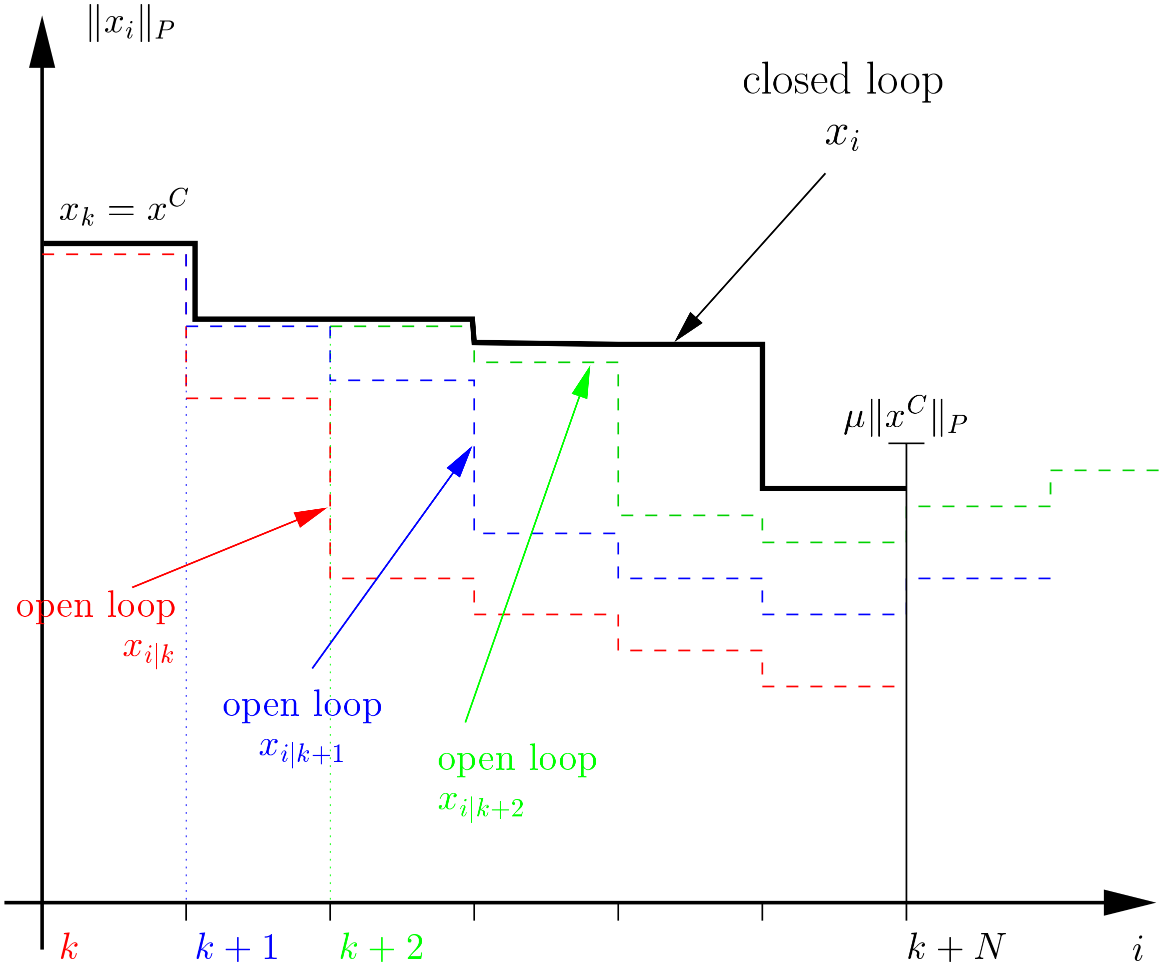

Figure 2.6: Contractive MPC, hypothetical trajectories and position of contraction constraint

Combined Feedback Linearization and "Linear" MPC

Oliveira et al. [12] propose a combination of two control concepts, input/output feedback linearization and "standard linear" MPC. The feedback linearization controller [21],[46] linearizes the input output behavior of the system. Using this as an inner loop we can apply the standard linear MPC. The major problem in this approach is that "the constrained MPC in the external loop enforces the existing input constraints in an implicit way" [12]. This follows from the fact, that the exact state feedback linearization law depends on the state and is nonlinear, this transfers the normal linear constraints on the inputs into nonlinear state dependent ones. They also present an algorithm which addresses these now nonlinear state dependent constraints. The resulting concept is pictured in Figure 2.8.

Shortest Prediction Horizon Nonlinear MPC and NIMC

Different authors have shown that for specific values of the control and prediction

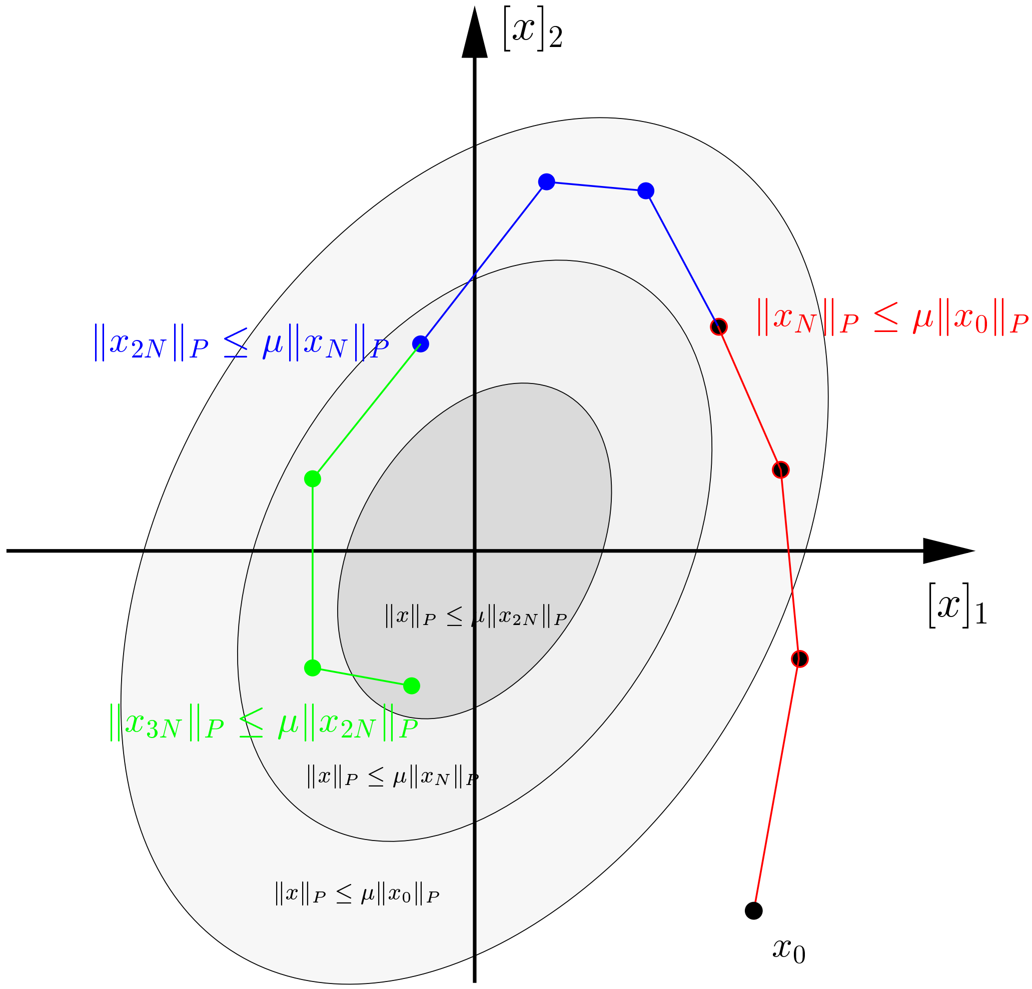

Figure 2.7: Contractive MPC, state space picture of resulting trajectories

horizon NMPC coincides with well known controller concepts of the geometrical control theory .

Soroush and Nikravesh [64] for example consider the so called shortest prediction horizon nonlinear MPC for continuous time. They derive a nonlinear MPC concept for continuous time by approximating the process model using a Volterra series. They show that for a short enough "prediction" horizon and no constraints, the resulting controller coincides with the feedback linearization controller.

Henson and Seborg [20] showed that nonlinear internal model control [19] for discrete time systems can be obtained as a special case of unconstrained NMPC.

A further presentation of these results would exceed the scope of this work.

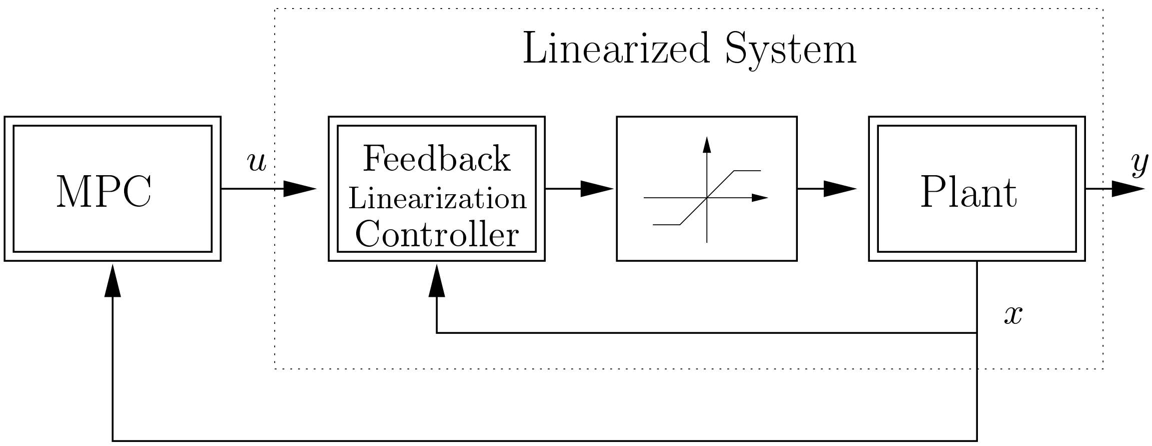

Figure 2.8: Combined feedback linearization and “linear” MPC

Chapter 3

Proposed Algorithm Convergence and Stability

During the presentation of the existing nonlinear MPC algorithms in the previous chapter we have seen that the following issues are important for an applicable NMPC controller:

- *Usage of a suboptimal solution instead of a global solution.* The traditional MPC approach requires the global solution of a non-convex nonlinear optimization problem to guarantee stability. This is in practice not achievable in finite computing time. Suboptimal concepts have been proposed, however they are based on requirements of the cost function and input sequence that are difficult to check in advance.

- *Avoiding the terminal state constraint.* The satisfaction of the stability and feasibility results in introducing a terminal state constraint, leads to a "hard" optimization problem. A strategy that would avoid or replace this constraint can lead to a significant simplification of the optimization problem.

Suboptimal dual mode controllers offer a solution to both problems. However the controller switches to another linear controller once the state has reached a specific region in the state space. This can lead to dramatical changes in the system behavior which can negatively influence the overall performance.

In this chapter we derive a new controller concept for discrete time systems. This controller utilizes the sub-optimality and avoids the use of a terminal state constraint. It can be seen as a combination of the constrained LQR (section 2.2.5) and the suboptimal dual mode concept as presented in section 2.3.2. Similar to the dual mode controller we constrain the final prediction state to a pre specified region. We avoid switching to another stabilizing controller by adding a final state penalty, which contains an upper bound on the optimal cost to go from the last predicted state to the origin. This bound is calculated by applying a hypothetical linear controller that stabilizes the nonlinear system in the terminal region. This is similar to the constrained LQR for which we calculated the exact value of the remaining infinite sum using a Lyapunov equation for the unconstrained LQR controlled system. The major difference between our concept and the constrained LQR is that the constrained LQR calculates an exact solution to the infinite horizon controller problem. This is only possible since the system equations are linear and the cost function is quadratic. Our controller provides, in contrary only a feasible, suboptimal and stabilizing solution. A similar concept for continuous time systems using quadratic cost function was recently proposed by Chen and Allgöwer ([7],[8].

In section 3.1 we derive a basic version of our controller, starting from the infinite horizon MPC problem. Section 3.2 contains the stability and convergence proofs, for the optimal case. Section3.3 shows that even a suboptimal version leads to stability and convergence. The last section of this chapter summarizes our concept in algorithmic form.

3.1 Basic Idea

We consider the control of the following nonlinear discrete time system

$$x_{k+1} = f(x_k, u_k) \tag{3.1}$$

with the given initial state x_0 and the input and state constraints

$$u_k \in \mathcal{U} \subseteq \mathbb{R}^m \qquad x_k \in \mathcal{X} \subseteq \mathbb{R}^n \tag{3.2}$$

We assume through the whole presentation that the following assumptions hold:

A1 $f \in \mathbf{C}^1,\ f(0,0) = 0$

A2 $\mathcal{U} \times \mathcal{X}$ compact and $(0,0) \in \mathcal{U} \times \mathcal{X}$

Probably the most natural formulation of nonlinear MPC is to consider an infinite control horizon. For the nominal case we would have to solve this problem only once, because the principle of optimality guarantees that the remaining part would also be optimal after we employ the first input. We will use the following discrete time formulation for infinite horizon MPC:

$$\Phi_\infty(x_k, \pi_\infty) = \sum_{j=0}^{\infty} L(x_{k+j|k}, u_{k+j|k}) \tag{3.3}$$

$$\min_{\pi_\infty} \Phi_\infty(x_k, \pi_\infty) \tag{3.4}$$

s.t.

$$x_{j+1|k} = f(x_{j|k}, u_{j|k}) \quad j = k, \dots \tag{3.5}$$
$$x_{k|k} = x_k \tag{3.6}$$
$$u_{j|k} \in \mathcal{U} \subseteq \mathbb{R}^m \tag{3.7}$$
$$x_{j|k} \in \mathcal{X} \subseteq \mathbb{R}^n \tag{3.8}$$

We implement the first u of the optimal sequence $\pi^\star_\infty(x_k) = \{u^\star_{k|k}, u^\star_{k+1|k}, \dots\}$ at time k

$$u^\star_k = u^\star_{k|k} \tag{3.9}$$

To guarantee feasibility we have to add the following:

A3 $\forall x_k \in \mathcal{X}$ $\exists$ a infinite input sequence $\pi_\infty(x_k) = \{u_k, u_{k+1}, \dots\}$ with $u_j \in \mathcal{U}$ so that $x_{k+1} = f(x_k, u_k)$ is asymptotically stable and satisfies (3.2) and (3.8) (feasibility of the ∞ horizon control problem).

We know from Theorem 2.2 that under the additional assumptions:

A4 $L(0,0) = 0$, L continuous at the origin.

A5 $\exists$ $\mathcal{K}^+$-function $\gamma[0,\infty) \to [0,\infty)$, so that $0 < \gamma(\|(x,u)\|) \leq L(x,u) \quad \forall (x,u) \neq (0,0)$

A6 $\Phi_N{}^\star(x)$ is continuous at the origin x=0

The origin of the controlled system is an asymptotically stable equilibrium point. The resulting control concept is a infinite dimensional optimization problem. To overcome this we limit the optimization variables to $N \cdot m$ by applying a stabilizing control law after N steps in the prediction. To make a “calculation” of the remaining infinite sum possible, we choose a static linear control law. This leads to the requirement that the linear controller has to stabilize the system in a region $\mathcal{B}$ around the origin, satisfying the state and input constraints and keeping the system in $\mathcal{B}$. So $\mathcal{B}$ is an positive invariant for (3.1) under $u = Kx$. We will show in Appendix A.1 that under the following assumptions such a region $\mathcal{B}$ exists:

A7 The linearization of (3.1) $x_{k+1} = Ax_k + Bu_k$ around the origin is stabilizable

$$A = \left.\frac{\partial f}{\partial x_k}\right|_{x_k, u_k = 0} \qquad B = \left.\frac{\partial f}{\partial u_k}\right|_{x_k, u_k = 0}$$

Since we need the linear controller at the end of the optimization horizon, we have to add $x_{k+N|k} \in \mathcal{B}$ as a constraint to the optimization problem. The resulting optimization is finite dimensional. Assumption A3 only assures feasibility for an infinite input sequence. To assure a feasible input sequence that brings the system in finite time in the interior of $\mathcal{B}$, we have to add:

A8 $\exists M$ finite, so that there is a input sequence $\pi_M = \{u_k, u_{k+1}, \dots, u_{k+M-1}\}$ which satisfies (3.2) and $x_{k+M} \in \mathcal{B}, \forall x_k \in \mathcal{X}$

In the following $N \geq M$.
After reformulating our optimization problem we get

$$\Phi_{\infty/N}(x_k, \pi_{\infty/N}) = \sum_{j=0}^{N-1} L(x_{k+j|k}, u_{k+j|k}) + \sum_{j=N}^{\infty} L(x_{k+j|k}, Kx_{k+j|k}) \tag{3.10}$$

$$\min_{\substack{\pi_\infty \\ \text{s.t.}(3.1)-(3.2) \\ x_{k+N|k} \in \mathcal{B}}} \Phi_{\infty/N}(x_k, \pi_{\infty/M})$$

$$\pi_{\infty/N} = \{u_{k|k}, u_{k+1|k}, \dots, u_{k+N-1|k}\}$$

Remarks: The ∞/N index represents the two parts of the sum, the finite up to $N-1$ and the infinite part. △

The assumptions made so far do not guarantee a solution with finite cost. We do not know if the last part of the sum $\sum_{j=N}^{\infty} L(x_{k+j|k}, Kx_{k+j|k})$ converges. In Appendix A.1 we show that we can find an quadratic upper bound on the infinite sum if we can bound $L(x_{k+j|k}, Kx_{k+j|k})$ in a region $\mathcal{W}, 0 \in \mathcal{W}$:

A9 $\exists\, Q > 0, R > 0$ so that:

$$L(x, Kx) \leq x^T Q x + x^T K^T R K x \quad \forall x \in \mathcal{W}$$

This leads to (A.1):

$$\sum_{j=k+N}^{\infty} L(x_j, Kx_j) \leq x_{k+N}^T \tilde{Q} x_{k+N}, \quad \forall x_{k+N} \in \hat{\mathcal{B}}, \hat{\mathcal{B}} \subset \mathcal{B} \wedge \hat{\mathcal{B}} \subset \mathcal{W}, \quad \tilde{Q} > 0 \tag{3.11}$$

The cost function becomes a sum over the finite horizon N plus an final state penalty term:

$$\Phi(x_k, \pi_k) = \sum_{j=0}^{N-1} L(x_{k+j|k}, u_{k+j|k}) + x_{k+N|k}^T \tilde{Q} x_{k+N|k} \tag{3.12}$$

The final optimal control strategy becomes:

$$\min_{\pi_k} \Phi(x_k, \pi_k) \tag{3.13}$$

$$u_k^\star = u_{k|k}^\star \tag{3.14}$$

s.t.

$$x_{j+1|k} = f(x_{j|k}, u_{k+j|k}) \quad j = k, \ldots \tag{3.15}$$

$$x_{k|k} = x_k \tag{3.16}$$

$$x_{k+N|k} \in \hat{\mathcal{B}} \tag{3.17}$$

$$u_{j|k} \in \mathcal{U} \subseteq \mathbb{R}^m \tag{3.18}$$

$$x_{j|k} \in \mathcal{X} \subseteq \mathbb{R}^n \tag{3.19}$$

3.2 Convergence and Stability, Optimal case

During this part we show that the above state optimal control algorithm results in an asymptotically stabilizing controller. The proof is based on Theorem 1.1. In the next section we modify the concept, so that even a suboptimal solution of (3.13) results in an stabilizing controller. The fact that we use a linear controller at the end of the horizon leads in the optimal case to a continuous control law near the origin. This avoids an additional, often difficult to check, assumption on the optimal cost function or input sequence as in the traditional NMPC approach.
The following theorem summarizes the above given fact.

Theorem 3.1 *If assumptions A1-A5, A7-A9 are satisfied, then the origin is an asymptotically stable equilibrium point for (3.1) if the moving horizon controller given by (3.13)-(3.19) is used.*

Proof: We use Theorem 1.1 to show the stability and convergence of the closed loop system. The Lyapunov function becomes, similar to the traditional NMPC, our optimal cost function $\Phi^\star$.
Lower bound on the cost function, requirement 2 of Theorem 1.1
It is easy to see that the cost function is bounded below by a $\mathcal{K}^+$ function, since we know from A5 that L is lower bounded by the $\mathcal{K}^+$ function γ

$$0 < \gamma(\|x_k\|) \leq \gamma(\|(x_{k|k}, u_{k|k})\|) \leq \sum_{j=0}^{N-1} L(x_{k+j|k}, u^\star_{k+j|k}) \leq \Phi^\star(x_k) \tag{3.20}$$

Decrease of cost function, requirement 3 of Theorem 1.1
We show that

$$\Phi^\star(x_k) - \Phi^\star(x_{k+1}) \geq L(x_k, u^\star_k) \tag{3.21}$$

so that part 3 of Theorem 1.1 is satisfied. This also implies that $(x_k, u_k) \to 0$ as $k \to \infty$.

Let $\Phi(x_k, \pi_k)$ represent the cost function at step k:

$$\Phi(x_k, \pi_k) = \sum_{j=0}^{N-1} L(x_{k+j|k}, u_{k+j|k}) + x_{k+N|k}^T \tilde{Q} x_{k+N|k} \tag{3.22}$$

with

$$\pi_k = \{u_k, u_{k+1}, \dots , u_{k+N-1}\} \tag{3.23}$$

Since we know that $u = Kx$ stabilizes the system in $\hat{\mathcal{B}}$ so that $\hat{\mathcal{B}}$ is positive invariant we can easily find an admissible sequence of new inputs $\tilde{\pi}_{k+1}$ for step $k + 1$.

$$\tilde{\pi}_{k+1} = \{u_{k+1|k}, u_{k+2|k}, \dots , u_{k+N-1|k}, Kx_{k+N|k}\} \tag{3.24}$$

with

$$\begin{aligned}\tilde{\Phi}(x_{k+1}, \tilde{\pi}_{k+1}) =& \sum_{j=1}^{N-1} L(x_{k+j|k}, u_{k+j|k}) + L(x_{k+N|k}, Kx_{k+N|k}) \\ &+ f(x_{k+N|k}, Kx_{k+N|k})^T \tilde{Q} f(x_{k+N|k}, Kx_{k+N|k})\end{aligned} \tag{3.25}$$

This leads to

$$\begin{aligned}\tilde{\Phi}(x_{k+1}, \tilde{\pi}_{k+1}) = \Phi(x_k, \pi_k) - L(x_{k|k}, u_{k|k}) + L(x_{k+N|k}, Kx_{k+N|k}) \\ + f(x_{k+N|k}, Kx_{k+N|k})^T \tilde{Q} f(x_{k+N|k}, Kx_{k+N|k}) - x_{k+N|k}^T \tilde{Q} x_{k+N|k}\end{aligned} \tag{3.26}$$

During the derivations in Appendix A.1 we will see that the following holds

$$x_{j+1}^T \tilde{Q} x_{j+1} - x_j^T \tilde{Q} x_j \le -x_j^T (Q + K^T RK) x_j \qquad \forall x_j \in \hat{\mathcal{B}} \tag{3.27}$$

After applying this to (3.26) we get

$$\begin{aligned}\tilde{\Phi}(x_{k+1}, \tilde{\pi}_{k+1}) \le& \Phi(x_k, \pi_k) - L(x_{k|k}, u_{k|k}) \\ &+ L(x_{k+N|k}, Kx_{k+N|k}) - x_{k+N|k}^T (Q + K^T RK) x_{k+N|k}\end{aligned} \tag{3.28}$$

Since $x_{k+N|k} \in \hat{\mathcal{B}}$ we can use A9 which guarantees that the sum of the last 2 terms is ≤ 0

$$\Phi(x_k, \pi_k) - \tilde{\Phi}(x_{k+1}, \tilde{\pi}_{k+1}) \ge L(x_k, u_k) \tag{3.29}$$

For the optimal control law we get that $\Phi(x_k, \pi_k) = \Phi^\star(x_k, \pi_k^\star)$ and $\Phi^\star(x_{k+1}, \pi_{k+1}^\star) \le \tilde{\Phi}(x_{k+1}, \tilde{\pi}_{k+1})$ which leads to

$$\Phi^\star(x_k, \pi_k) - \Phi^\star(x_{k+1}, \pi_{k+1}) \ge L(x_k, u_k) \tag{3.30}$$

This does satisfy the requirements of Theorem 3.

Upper bound of cost function, continuity requirement 1 of Theorem 1.1

We know that in $\hat{\mathcal{B}}$ a feasible solution exist: $\pi_k^K = \{Kx_k, Kx_{k+1}^K, \dots, Kx_{k+N-1}^K\}$. This sequence leads to an upper bound of the optimal cost function:

$$\begin{aligned}
\Phi^K(x_k) &= L(x_k, Kx_k) + \ldots + L(x_{k+N-1}^K, Kx_{k+N-1}^K) + (x_{k+N}^K)^T \tilde{Q} x_{k+N}^K && (3.31)\\
&\underset{x_j \in \hat{\mathcal{B}}}{\leq} \sum_{j=k}^{N-1} (x_j^K)^T (Q + K^T R K) x_j + (x_{k+N}^K)^T \tilde{Q} x_{k+N}^K && (3.32)
\end{aligned}$$

which leads with (3.27) to

$$0 < \Phi^K(x_k) \leq x_k^T \tilde{Q} x_k \tag{3.33}$$

This implies that $\Phi_k^\star$ is upper bounded by a $\mathcal{K}^+$ function:

$$0 < \Phi^\star(x_k) \leq \Phi^K(x_k) \leq x_k^T \tilde{Q} x_k \tag{3.34}$$

Remarks: Furthermore this shows that $\Phi^\star$ is continuous around the origin since $\Phi^\star$ is also bounded below by 0. △

This completes the necessary assumptions for Theorem 1.1 and with this we can conclude our proof. □

3.3 A Suboptimal Version

The proposed algorithm replaces the terminal end constraint with a terminal end region or inequality. This relaxes the resulting optimization problem significantly. The remaining question is, if it is possible to replace the optimization by a search for a control sequence that decreases the cost function?

The key steps in stability proof are, the decrease of the cost function and the upper bound on the cost function near the origin. The first point establishes the convergence from every point in the feasible set. The upper bound plus the decrease allows us to guarantee stability near the origin, and both combined establish asymptotic stability of the controlled system. The convergence does not depend on the optimality of the cost function. However for the continuity (or upper bound) of Φ for a suboptimal feasible solution we have to change the proposed approach slightly.

Decreasing of Φ for the suboptimal case

We can conclude from equation (3.29) that for a decrease in the cost function it is not necessary to have an optimal solution. We only have to find a feasible solution Φ which can "beat" the hypothetical input $\tilde{\Phi}$. This means we can replace the optimization

(3.13) with the following:
Find a feasible sequence $\pi(x_k)$ so that

$$\Phi(x_k, \pi_k) \leq \tilde{\Phi}(x_k, \tilde{\pi}_k) \tag{3.35}$$

$$\text{s.t.}(3.15) - (3.19) \tag{3.36}$$

with:

$$\tilde{\pi}_k = \{u_{k|k-1}, u_{k+1|k-1}, \ldots, u_{k+N-2|k-1}, Kx_{k+N-1|k-1}\} \tag{3.37}$$

Remarks: Similar to the Suboptimal MPC concept proposed by Scokaert et al. ([56]) (see section 2.3.1) would it be also sufficient to use the following instead of (3.35).

$$\Phi(x_k, \pi_k) \leq \Phi(x_{k-1}, \pi_{k-1}) - \mu L(x_{k-1}, u_{k-1}) \quad \mu \in (0, 1] \tag{3.38}$$

using (3.37) as an initial guess. This still enforces the necessary decrease of the cost function and at the same time provides a more robust algorithm for the disturbance case [56]. At the same time we should also use a subset $\hat{\mathcal{B}}' \subset \hat{\mathcal{B}}$ to ensure that under disturbances or model-mismatch $x^P_{k+N|k}$ is still in the region where the linear controller can stabilize, see for comparison Robust Dual Mode control (section 2.3.2). △

Upper bounding of Φ near the origin
To guarantee that $\Phi^K(x_k)$ in $\hat{\mathcal{B}}$ is still an upper bound for $\Phi(x_k)$ once the state has entered $\hat{\mathcal{B}}$ we have to add to the control law:

$$\Phi(x_k) \leq \Phi^K(x_k) \qquad x_k \in \hat{\mathcal{B}} \tag{3.39}$$

If our "search" algorithm can not find a feasible solution which satisfies this in finite time we can use π_k^K as the new input sequence.
Remarks: The difference between our algorithm and the Dual Mode control is that we only switch to the pre-given control law $u_k = Kx_K$ if we can not find a feasible solution. If we can find a feasible solution the resulting cost is lower than that of the linear controller. △

With this we can proceed as in the optimal version and prove that the following holds.

Theorem 3.2 *If assumptions A1-A5, A7-A8 are satisfied, then the origin is an asymptotically stable equilibrium point for (3.1) under {(3.35)-(3.36),(3.39)} or {(3.38),(3.36),(3.39)} given moving horizon controller.*

3.4 Proposed Algorithm

We can summarize our proposed control concept as follows:

"Robust" Suboptimal Zero Endpoint Constraint free Control Algorithm

Off-line: calculate K, Q, $\tilde{Q}$, R, $\hat{\mathcal{B}}' \subset \hat{\mathcal{B}}$, choose $\mu \in (0, 1]$

Online:

STEP 0: Initialization, time $k = 0$, x_0

- compute a feasible input sequence $\pi_0 = \{u_{0|0}, \dots, u_{N-1|0}\}$ which satisfies (3.15)-(3.19)
- apply $u_0 = u_{0|0}$

STEP 1: time $k > 0$

IF $x_k \in \hat{\mathcal{B}}'$

- $\tilde{\Phi} = \min(\Phi^K(x_k), \Phi(x_{k-1}, \pi_{k-1}) - \mu L(x_{k-1}, u_{k-1}))$

ELSE

- $\tilde{\Phi} = \Phi(x_{k-1}, \pi_{k-1}) - \mu L(x_{k-1}, u_{k-1})$

ENDIF

- compute a feasible input sequence π_k which satisfies (3.15)-(3.19) and $\Phi(x_k, \pi_k) \leq \tilde{\Phi}$
  using $\tilde{\pi}_k = \{u_{k|k-1}, u_{k+1|k-1}, \dots, u_{k+N-2|k-1}, Kx_{k+N-1|k-1})$ as a initial guess

IF this is not possible in feasible time and $x_k \in \hat{\mathcal{B}}'$

- set $\pi_k = \pi_k^K = \{Kx_k, Kx_{k+1}^K, \dots, Kx_{k+N-1}^K\}$

ENDIF

- apply $u_k = u_{k|k}$

GOTO STEP 1

Remarks:
We can replace $\Phi(x_{k-1}, \pi_{k-1}) - \mu L(x_{k-1}, u_{k-1})$ by $\tilde{\Phi}(x_k, \tilde{\pi}_k)$ with $\tilde{\pi}_k = \{u_{k|k-1}, u_{k+1|k-1}, \dots, u_{k+N-2|k-1}, Kx_{k+N-1|k-1})$. This leads to a less robust controller with respect to modelling errors and disturbances. $\triangle$

Suboptimal Zero Endpoint Constraint free Control Algorithm

Off-line: calculate K, Q, $\tilde{Q}$, R, $\hat{\mathcal{B}}$

Online:

STEP 0: Initialization, time $k = 0$, x_0

- compute a feasible input sequence $\pi_0 = \{u_{0|0}, \dots, u_{N-1|0}\}$ which satisfies (3.15)-(3.19)
- apply $u_0 = u_{0|0}$

STEP 1: time $k > 0$

IF $x_k \in \hat{\mathcal{B}}$

- $\tilde{\Phi} = \min(\Phi^K(x_k), \tilde{\Phi}(x_k, \tilde{\pi}_k))$
  with $\tilde{\pi}_k = \{u_{k|k-1}, u_{k+1|k-1}, \dots, u_{k+N-2|k-1}, Kx_{k+N-1|k-1})$

ELSE

- $\tilde{\Phi} = \tilde{\Phi}(x_k, \tilde{\pi}_k$
  with $\tilde{\pi}_k = \{u_{k|k-1}, u_{k+1|k-1}, \dots, u_{k+N-2|k-1}, Kx_{k+N-1|k-1})$

ENDIF

- compute a feasible input sequence π_k which satisfies (3.15)-(3.19) and $\Phi(x_k, \pi_k) \leq \tilde{\Phi}$
  using $\tilde{\pi}_k = \{u_{k|k-1}, u_{k+1|k-1}, \dots, u_{k+N-2|k-1}, Kx_{k+N-1|k-1})$ as a initial guess

IF this is not possible in practicable time and $x_k \in \hat{\mathcal{B}}$

- set $\pi_k = \pi_k^K = \{Kx_k, Kx_{k+1}^K, \dots, Kx_{k+N-1}^K\}$

ENDIF

- apply $u_k = u_{k|k}$

GOTO STEP 1

The resulting controller is illustrated in Figure 3.1.

Remarks: (**Comparison proposed Algorithm/Method of Chen and Allgöwer** [**8**]) A similar concept was recently proposed by Chen and Allgöwer for the continuous time case. Our work is not based on these results. However, we think it is important to show the major differences. Chen and Allgöwer consider the continuous time case, without state constraints and case with a quadratic cost function whereas, we consider the discrete time case with state constraints and a general nonlinear cost function for the . They derive an suboptimal version and show convergence. During their work they propose an algorithm which can calculate a potentially *large* region $\hat{\mathcal{B}}$. The method presented in appendix A.1 does not, in general, lead to as large a terminal region. The major difference is the introduction of constraint (3.39) that we think is necessary for stability in the suboptimal case. We also derive a "robust" version, based on works of Michalska and Mayne [43] and Scokaert et al. [56]. Chen and Allgöwer are mostly interested in the computational and feasibility improvements due to the removal of the zero end constraint and on finding an upper bound on the optimal infinite horizon cost function. We are mostly interested in suboptimal stabilizing solution that is computationally feasible. $\triangle$

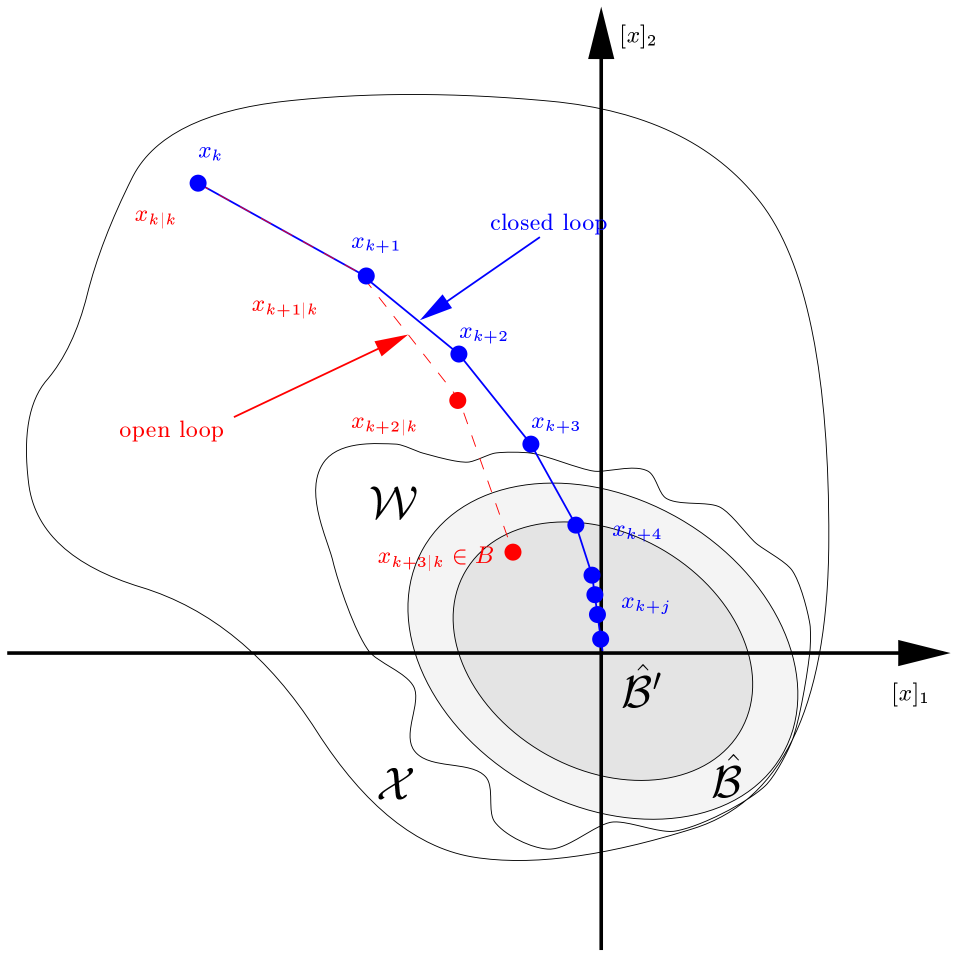

Figure 3.1: Closed and open loop behavior of proposed algorithm

Chapter 4

Implementation and Examples

As outlined in the previous chapters the optimization is the critical point in the MPC implementation. A transfer of the theoretical properties for linear systems derived to the nonlinear case is in most cases possible. It is the non-convexity of the cost and constraint functions which leads to a significant increase in complexity. Whereas the solution of quadratic optimization problems subject to linear constraints is well investigated, the solution of an general non-convex optimization is still more an “art than a science” [34]. The success or failure of the optimization routine is often connected to the experience of the user. We consider only discrete time systems. This avoids some of the problems encountered during the solution of continuous time MPC formulations. Especially the integration of the system equations in combination with the optimization can lead to major problems like numerical instability or ill conditioning ([33],[61]).
There are two major points to be addressed:

- *Usage of the inherent structure in the optimization problem* If we consider both the states and inputs as optimization variables, the resulting optimization problem takes a special structure. For example the Hessian of the Lagrangian becomes a sparse banded matrix. A well suited algorithm for the optimization problem should take advantage of this.

- *Feasibility and decrease of the cost function during optimization sub steps.* We have seen that some of the proposed algorithm, like Suboptimal MPC, Suboptimal Dual Mode MPC and our own proposed algorithm, do not necessarily depend on a global or even local minima to guarantee stability. It would be sufficient if the “optimizer” could provide a feasible decedent solution, instead of finding a global or local minima.

Mayne [36] gives a good review of possible algorithm and problems.

4.1 Computational Implementation, Special Algorithms

We can address optimization problems arising from MPC in different ways:

- *Elimination of the x_k:* We can include the state equation into the cost function. This leads to the following optimization problem:

$$\begin{aligned} \min_{\{u_{k|k}, u_{k+1|k}, \cdots, u_{k+N-1|k}\}} \Phi_k(x_k, \pi_k) &= L(x_{k|k}, u_{k|k}) + \\ & L(f(x_{k|k}, u_{k|k}), u_{k+1|k}) + \\ & L(f(f(x_{k|k}, u_{k|k}), u_{k+1|k}), u_{k+2|k}) + \dots \\ \text{s.t.} \quad u_{j|k} &\in \mathcal{U} \\ x_{j|k} &= \gamma(u_{k|k}, u_{k+1|j}, u_{k+j-1|k}) \in \mathcal{X} \end{aligned}$$

  The satisfaction of the state equations can be automatically satisfied. If we have no additional equality constraints we get an nonlinear, non convex optimization problem subject to inequality constrains. The optimization problem has pN optimization variables. The disadvantage is that we get a dense Hessian of the Lagrangian of Φ_k subject to the constraints. Since most optimizers have to invert the Hessian during iteration sub steps, the problem gets more and more computational infeasible for larger horizons N.

- *Optimize over x_k and u_k:* Instead of including the state equations in the objective function, we can treat the x_k as additional optimization variables and optimize with respect to the state equality constraints. The resulting optimization problem has $n(N-1) + pN$ optimization variables $\hat{x} = (x_1, x_2, \dots, x_{N-1}, u_0, \dots, u_{N-1})$. This seems to increase the dimensionality of the problem, but since we have a cost function which is a sum over stage costs the Hessian of the Lagrangian has a sparse structure. The inversion of this Hessian in the optimizer subproblem is not as expensive as in the case above. On the other hand we have no information how many iterations we need to satisfy the equality constraints, because we have these constraints not explicit included in the objective function.

The proposed algorithm in 3.4 does not require a solution to the optimization problem. Panier and Tits ([47],[49], [48]) presented in a series of papers on an optimization algorithm which can guarantee feasibility and descent at every substep under the absence of nonlinear equality constraints. The removal of the nonlinear state equality constraints from the optimization problem can be achieved by considering only the inputs as optimization variables. We decided to implement our algorithm under the usage of this optimizer since we can utilize the fact that we only need a suboptimal feasible solution. However we think that a combination of both ways might be possible, so that we could satisfy the wish after sub-optimality while still consider the special structure. The following two sections contain a discussion of both solution ways.

4.1.1 x_k and u_k as Decision Variables

The optimization algorithms considered here are based on Sequential Quadratic Programming. Since its popularization in the late 1970s, sequential quadratic programming (SQP) has arguable become the most successful method for solving nonlinear constrained optimization problems. SQP is not a single algorithm, but rather a conceptual method from which numerous specific algorithms have evolved. Boggs [6] gives a good review of the general idea and the different realizations. The considered nonlinear programming is

$$\begin{aligned} &\min_x f(x) \\ \text{s.t.:}\quad &h(x) = 0 \\ &g(x) \leq 0 \end{aligned}$$

with $f : \mathbb{R}^n \to \mathbb{R}$, $h : \mathbb{R}^n \to \mathbb{R}^m$, and $g : \mathbb{R}^n \to \mathbb{R}^p$. The basic idea of SQP is to approximate the nonlinear equations at a given sub step x^k, by a quadratic programming subproblem, and then use the solution to this subproblem to construct a better approximation x^{k+1}. This process is repeated and it is hoped that the sequence converges to the minima.

Dunn and Bertsekas [14] showed that for the classical discrete-time-optimal control problem with Bolza objectives (like in our case) the Hessian of the optimization is dense if we only consider the inputs $\{u_{k|k}, \ldots, u_{k+N-1|k}\}$ as optimization variables. Since the solution of the quadratic subproblem often requires a inversion of this Hessian, we would need $O(N^3m^3)$ operations. They derived a special algorithm which could take advantage of the, the problem inherent structure, which requires only $O(N(m^3 + n^3))$ operations.

This becomes clear if we use the states and inputs as optimization variables $\{u_{k|k}, \ldots, u_{k+N-1|k}, x_{k|k}, \ldots, x_{k+N-1|k}\}$ and view the state equations as nonlinear equality constraints. The Jacobian of the constraints and the Hessian of the objective are both sparse, block-banded matrices. The result is a drastic performance increase, if we use a SQP algorithm which exploits the sparse banded structure of the problem. Wright [70] outlines a concept which employs a interior point method for the solution of the quadratic subproblem which makes use of the special structure and results in an highly efficient algorithm. Other authors have also explored the sparsity. For example Steinbach [65] or Arnold et al. ([3],[2]) proposed Riccati like recursive solution algorithm to factorize the inverse in the underlying QP. A similar approach was recently proposed by Rao et. al [52] for the efficient solution of the from linear MPC resulting structured QP problem.

It is clear that the presented solution method is highly efficient and explores the underlying structure of the optimization problem. However it is not possible to expect that a feasible descent solution is found before the algorithm has converged. This property makes it at this time impossible to employ this kind of optimization algorithm to find a suboptimal feasible descent solution. This is especially cumbersome since a on-line solution of the optimal NMPC problem even with this effective algorithm is for computational reasons not possible.

4.1.2 Feasible Sequential Quadratic Programming, FSQP

The usage of an optimization algorithm in 2.3.1, 2.3.2 which guarantees feasibility and a decrease of the cost function during sub steps is necessary. In fact in these cases the optimization degrades to a search for a new feasible solution with a lower cost function. Standard SQP algorithm cannot guarantee the feasibility, neither the decrease at every sub step. They mostly decrease the value of an augmented Lagrangian resulting from the KKT conditions, which does only guarantee feasibility and decrease of the cost function once the algorithm has converged. We use the proposed algorithm by Panier and Tits ([47], [49],[48]), that guarantees feasibility and monotonically decreasing cost functions in the absence of nonlinear equality constraints.
The standard FSQP algorithm considers the following optimization problem

$$\begin{array}{rc} & \min_x \max_{i \in I}\{f_i(x)\} \\ \text{s.t.:} & h_j(x) = 0 \\ & g_j(x) \leq 0 \end{array}$$

FSQP generates iterates that satisfy all inequality constraints and linear equality constraints. If nonlinear equality constraints are present, the inequality constraints and linear equality constraints are satisfied at every sub step. However it is not possible to guarantee feasibility of the nonlinear equality constraints until the algorithm has converged. The calculation of a descent feasible direction can be divided in two phases [48]:

PHASE I: generate iterate satisfying all linear constraints and nonlinear inequality constraints.

PHASE II: minimize maximum of objectives. Iterates satisfy all constraints except nonlinear equality constraints

The resulting algorithm has two-step superlinear convergence. First the "standard" quadratic SQP subproblem is solved. The resulting direction d^0 is descent, but not necessary feasible. To remove this difficulty, d^0 is replaced by a convex combination of d^0 and a second, arbitrary feasible descent direction d^1. The resulting direction is "feasible" [10], and guarantees a quasi Newton like behavior of the algorithm near the solution point (which does then guarantee the 2 step superlinear convergence). In the last step a search over an arc described by $x^k + td + t^2 d$ where $t \in (0, 1]$ is performed to guarantee the feasibility and descent requirement [48]. From this description we can suspect the necessary numerical cost. At every iteration the algorithm has to solve 3 convex quadratic programs and perform one line-search (arch-search).

If we consider in our optimization problem only the inputs u_k as optimization variables we avoid all equality constraints and can directly employ the FSQP algorithm on the resulting problem. A interface to access CFSQP from Octave[1] basing on an earlier implementation by Peter Watkins was developed by the author and Peter Findeisen.

4.2 Examples

During the following section we show some features of the developed algorithm. We use two more instructional than sophisticated examples. The simulations where produced by Octave using the described interface to CFSQP.

4.2.1 A Small Explanatory Example

The system equation for the first example is given by:

$$x_{k+1} = 0.5x_k^2 + u_k \tag{4.1}$$

For this example, we can easily find a region $\hat{\mathcal{B}}$, a stabilizing liner gain K and the final state penalty $\tilde{Q}$. We use the non-robust algorithm as given in 3.4. We consider the following additional constraints

$$|u_k| \leq 2, \quad |\Delta u_k| \leq 0.4, \quad |x_k| \leq 2.5 \tag{4.2}$$

The horizon length be $N = 5$, the stage cost is given by

$$L(x_k, u_k) = x_k^2 + u_k^2 \tag{4.3}$$

We first notice that the linearization of the given system (4.1 at the origin is stabilizable. The chosen linear control law is $K = 0$ which stabilizes the system in the region

[1]A Matlab like interpreter for numerical calculations which was developed by John W. Eaton. The program freely available using anonymous FTP from bevo.che.wisc.edu.

$\mathcal{B} = \{x : |x| \leq \sqrt{2}\}$. This set is also positive invariant under this control law. What remains to show is, that we can find a matrix $\tilde{Q}$ so that equation (A.2) and (A.3) in the appendix are satisfied. First we have to find an upper bound on the infinite sum which in this case, will also provide $\tilde{Q}$.

$$\sum_{j=k}^{\infty} L(x_j, Kx_j) = \sum_{j=k}^{\infty} x_j^2 + 0x_j^2 \tag{4.4}$$

$$= \sum_{j=k}^{\infty} x_j^2 \tag{4.5}$$

From the system equations we get

$$x_{k+1} = 0.5x_k^2 \rightarrow x_{k+j} = (\frac{x_k}{\sqrt{2}})^{(2)^{j+1}} \tag{4.6}$$

so that we have to bound the following sum from above

$$\sum_{j=k}^{\infty} L(x_j, Kx_j) = \sum_{j=0}^{\infty} (\frac{x_k}{\sqrt{2}})^{(2)^{j+1}} \tag{4.7}$$

$$= (\frac{x_k}{\sqrt{2}})^2 \sum_{j=0}^{\infty} (\frac{x_k}{\sqrt{2}})^{(2)^j} \tag{4.8}$$

$$\leq (\frac{x_k}{\sqrt{2}})^2 \frac{1}{1 - \frac{x_k}{\sqrt{2}}} \quad \forall\, |x| < \sqrt{2} \tag{4.9}$$

If we choose $x_k \leq \frac{1}{\sqrt{2}}$ it follows,

$$\sum_{j=k}^{\infty} L(x_j, Kx_j) \leq x_k^2 \tag{4.10}$$

which leads to $\tilde{Q} = 1$. We decided to choose $\hat{\mathcal{B}}$ as

$$\hat{\mathcal{B}} = \{x_k : |x_k| \leq 0.5\} \tag{4.11}$$

which satisfies together with $\tilde{Q} = 1$ (A.2). However to satisfy (A.3) we have to set $\tilde{Q} \geq 2$. We have chosen $\tilde{Q} = 2.5$

Remarks: The rate of change constraint does not lead to problems, since it is inside of $\hat{\mathcal{B}}$ with $u = 0$ always satisfied, so that we do not have to augment the system to form (2.17). $\triangle$

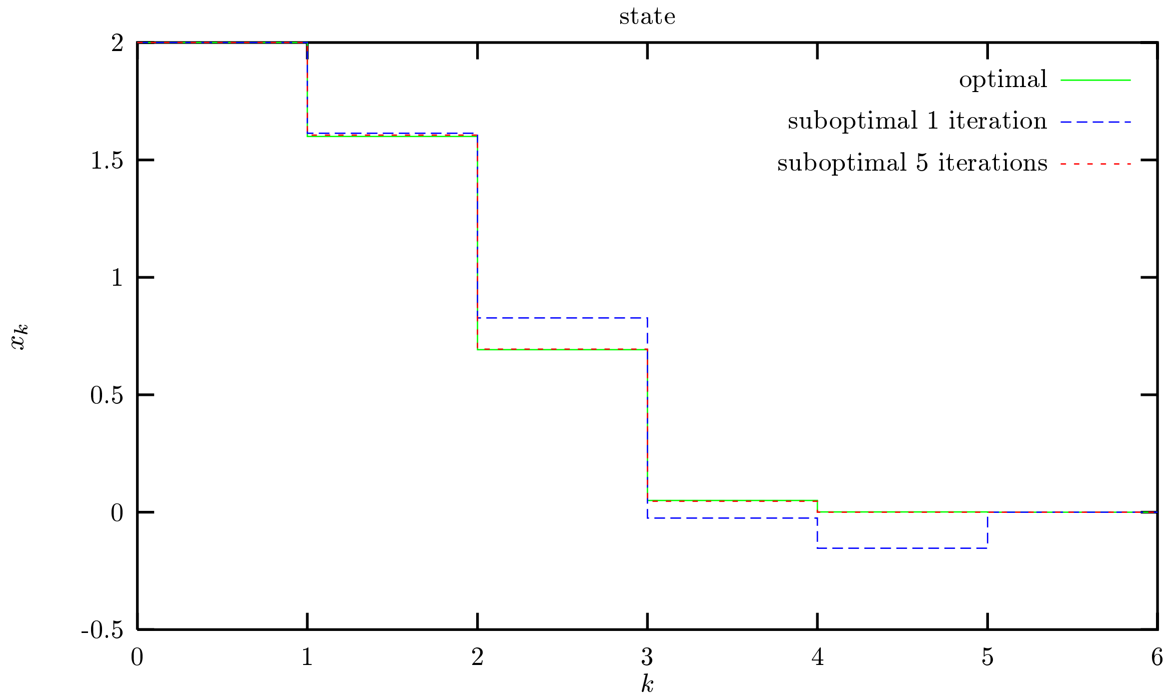

Figure 4.1: State for different iteration depths, horizon $N = 5$

4.2.2 Discussion of the Simulation Results

We tried the following simulations for different maximal values of iterations: 1 iteration, 5 iterations and as many iterations as necessary for finding the minima (the resulting optimization problem is convex, so that the calculated minima is global. The FSQP algorithm was used to ensure that suboptimal solutions are feasible and have a decreasing cost function. As we see from Figure 4.1 and Figure 4.2 the performance decrease between the results for 5 iterations and the optimal solution is small. This is especially interesting, since the CPU time used for 5 iterations is less than half compared to the optimal solution, see Table 4.1.

The solution for 1 iteration shows an noticeable performance decrease, but still provides stability and satisfactory performance. Figure 4.3 shows the value of the cost function Φ over time. It is important to notice that the cost function of the suboptimal solutions, as given by Figure 4.3, could fall at times after $k = 1$ below the "optimal" value, since the states are different after the first input is applied.

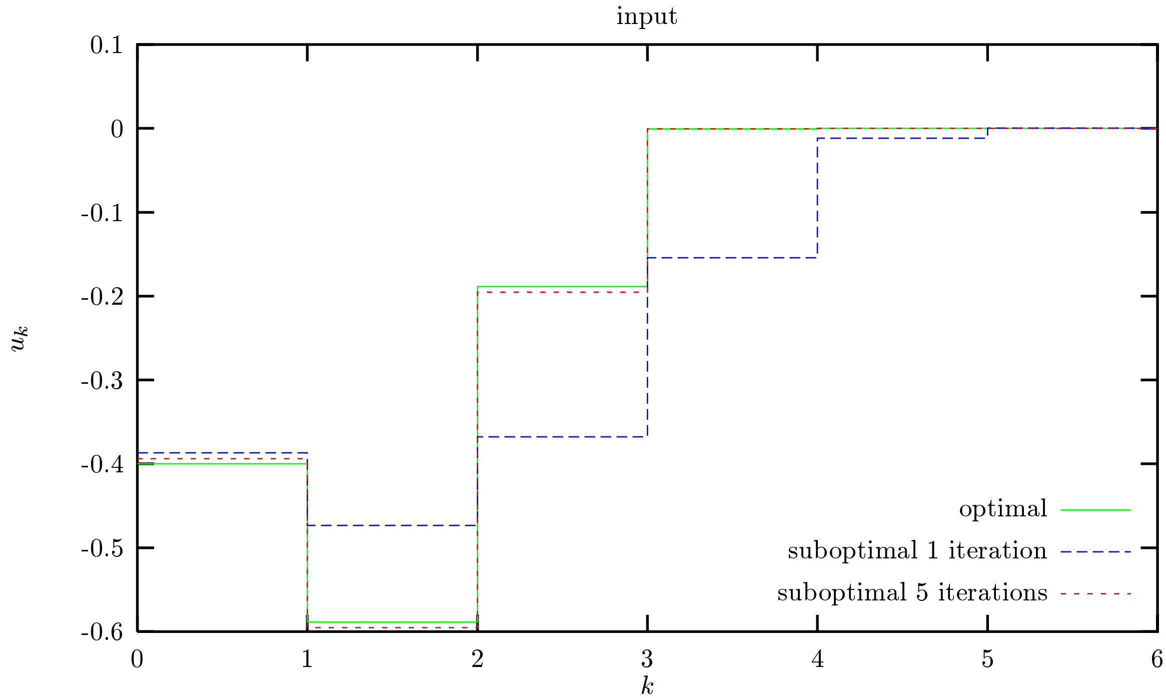

Figure 4.2: Input u for different iteration depths, horizon $N = 5$

4.2.3 The Van de Vusse Reactor

We consider the following reaction scheme, taking place in a isotherm CSTR:

$$A \xrightarrow{k_1} B \qquad B \xrightarrow{k_2} C \qquad 2A \xrightarrow{k_3} D \tag{4.12}$$

The reactor is assumed to be volume constant, isotherm and perfectly mixed. The continuous time model is given by:

$$\begin{aligned} \dot{x}^1 &= -k_1 x^1 - k_3 (x^1)^2 + (x_F - x^1) u \\ \dot{x}^2 &= k_1 x^1 - k_2 x^2 - x^2 u \end{aligned} \tag{4.13}$$

	CPU time $[s]$
optimal solution	312.15
suboptimal 5 FSQP iterations	152.67
suboptimal 1 FSQP iteration	51.34

Table 4.1: Computational cost for different iteration depths

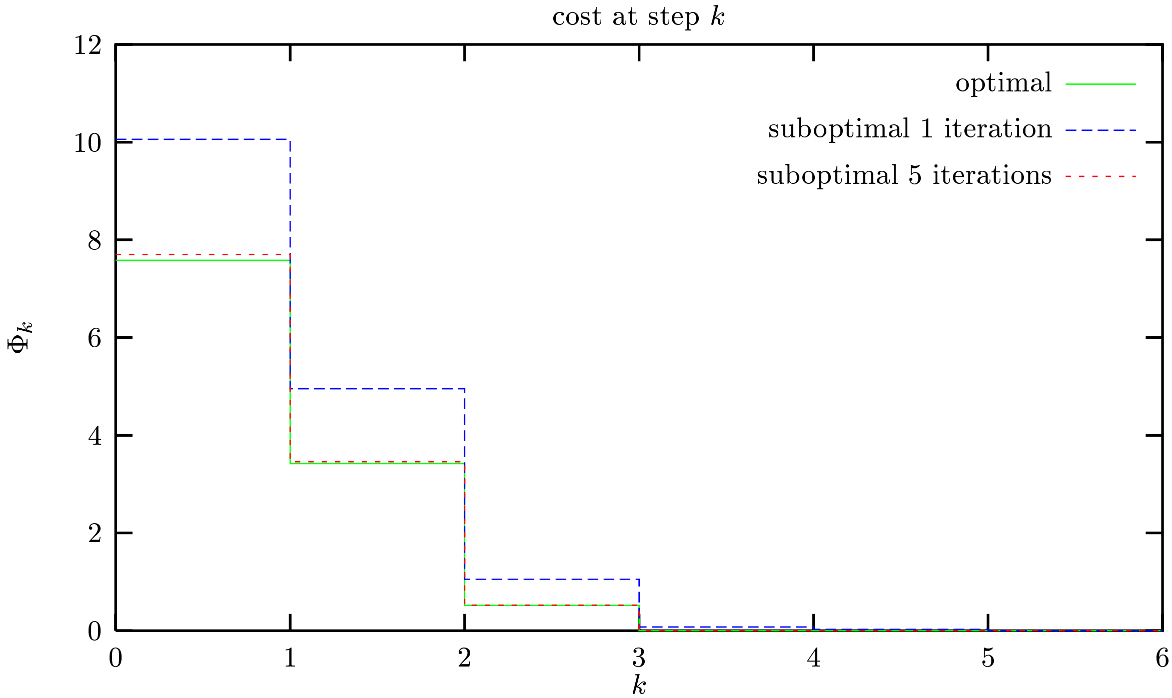

Figure 4.3: Cost Φ, for different iteration depths, horizon $N = 5$

With $x^1 = c_A, x^2 = c_B$ as the states. The manipulated variable is the dilution rate u. $x_f = c_{Ain}$ is the inlet stream concentration of A.

After discretization using the Euler approximation method, and the output measurement equation $y_k = c_{Bk}$ we get:

$$\begin{aligned}
x^1_{k+1} &= (1 - \Delta T k_1)x^1_k - \Delta T k_3 (x^1_k)^2 + \Delta T (x^F_k - x^1_k)u_k \qquad (4.14)\\
x^2_{k+1} &= (1 - \Delta T k_2)x^2_k + \Delta T k_1 x^1_k - \Delta T x^2_k u_k\\
y_k &= x^2_k = c_{Bk}
\end{aligned}$$

With the fixed parameter values (see [42]):

$$k_1 = 50\frac{1}{\mathrm{hr}},\ k_2 = 100\frac{1}{\mathrm{hr}},\ k_3 = 10\frac{\mathrm{lit}}{\mathrm{mol\ hr}},\ x_f = 10\frac{\mathrm{mol}}{\mathrm{lit}},\ \Delta T = 0.0005\mathrm{hr}$$

Using this values we can calculate two possible steady states for which we can achieve the desired output $y_s = 1\frac{mol}{liter}$.

$$x^1_s = 2.5\frac{\mathrm{mol}}{\mathrm{lit}}, x^2_s = 1.0\frac{\mathrm{mol}}{\mathrm{lit}}, u_s = 25\frac{\mathrm{mol}}{\mathrm{lit}} \qquad (4.15)$$

$$x^1_s = 6.6667\frac{\mathrm{mol}}{\mathrm{lit}}, x^2_s = 1.0\frac{\mathrm{mol}}{\mathrm{lit}}, u_s = 233.33\frac{\mathrm{mol}}{\mathrm{lit}} \qquad (4.16)$$

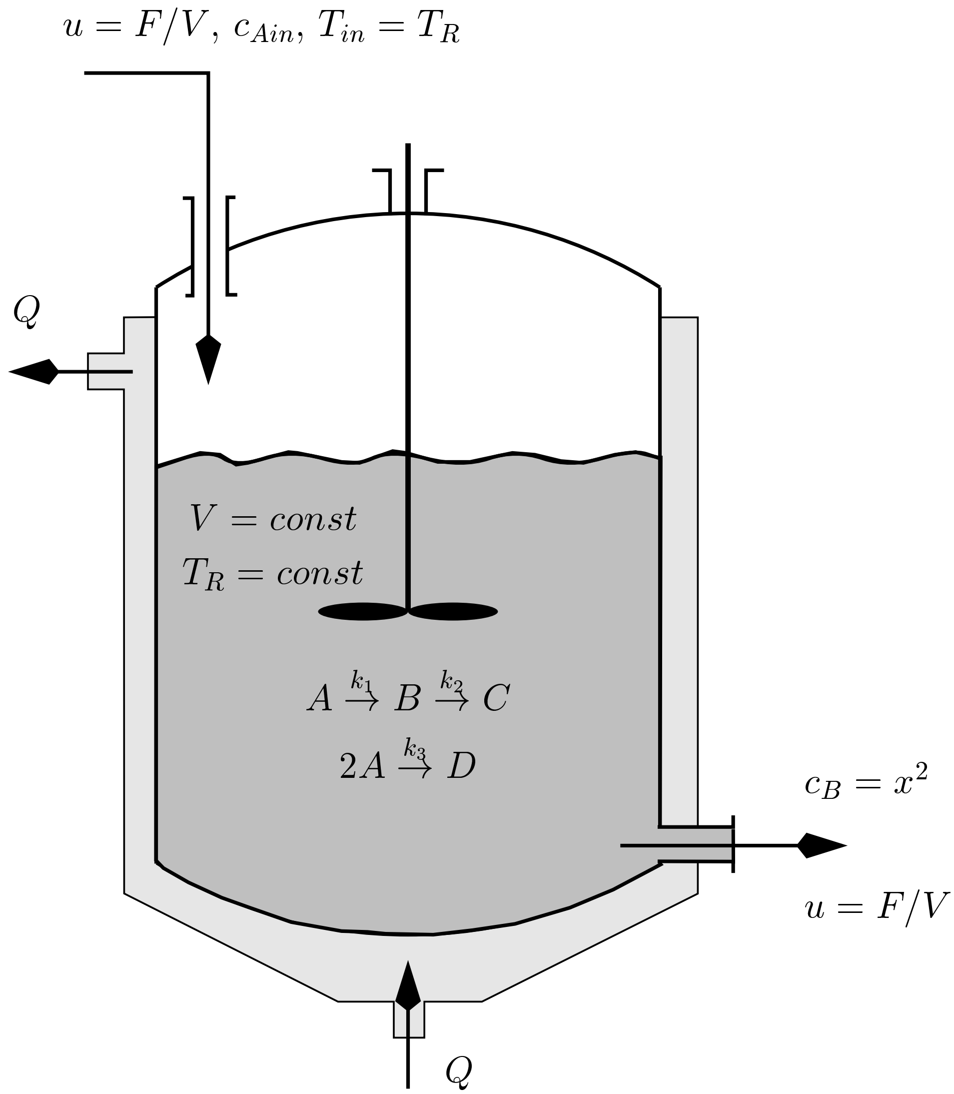

Figure 4.4: Schematic representation of the van de Vusse reactor

According to [42] the first steady state vector $\{2.5, 1.0, 25\}$ is preferable, because lower feed rates lead to higher conversion rates. After linearizing the equations we checked that the first steady state is controllable and stable:

$$A = \begin{bmatrix} 0.9375 & 0.0 \\ 0.025 & 0.9375 \end{bmatrix} \quad B = \begin{bmatrix} 0.00375 \\ -0.0005 \end{bmatrix} \tag{4.17}$$

so that we can find a K, $\hat{\mathcal{B}}$, $\tilde{Q}$. We chose as K the solution of the LQR problem with the in (4.19) given Q and R matrices:

$$K = \begin{bmatrix} 18.41 & 1.06 \end{bmatrix} \tag{4.18}$$

Simulations

The considered costfunction for this reactor is:

$$L(x_{j|k}, u_{j|k}) = 1000[x^1_{j|k} - 2.5]^2 + 1000[x^2_{j|k} - 1.0]^2 + [u_{j|k} - 25]^2 \tag{4.19}$$

This means that we are penalizing differences between the desired steady state value and the actual state. This is equal to a coordinate transformation which moves the stabilizable steady state to the origin.

To compare the behavior of the suboptimal and optimal controllers, we simulated the system with the following maximal iteration numbers: 1, 3 and unlimited=optimal solution. We used an horizon length of $N = 11$ without any state or input constraints.

4.2.4 Discussion of the Simulation Results

We can see from the state and input trajectories in Figures 4.5–4.7 that the three simulations show significant different behavior. Especially the 1 step iteration solution is degraded. In comparison the 3 step solution shows good performance compared to the optimal one. This is obvious in Figure 4.8, which shows the summed stage costs $\sum_{j=0}^{k} L(x_j, u_j)$. We see a drastical performance difference between 1 iteration and 3 iterations or the optimal solution. This is similar to the first example. The comparison of the computational time is given in Table 4.2. As in the first example, the execution for 1 or 3 iterations is significantly lower than for the optimal solution. The used CPU time seems to be small compared to example 1, especially since the horizon length is 11. This is mostly due to the fact that we have no constraints on the states; these would lead to a drastical increase in the necessary computation time.

	CPU time $[s]$
optimal solution	544.35
suboptimal 3 FSQP iterations	246.82
suboptimal 1 FSQP iteration	134.61

Table 4.2: Computational cost for different iteration depths

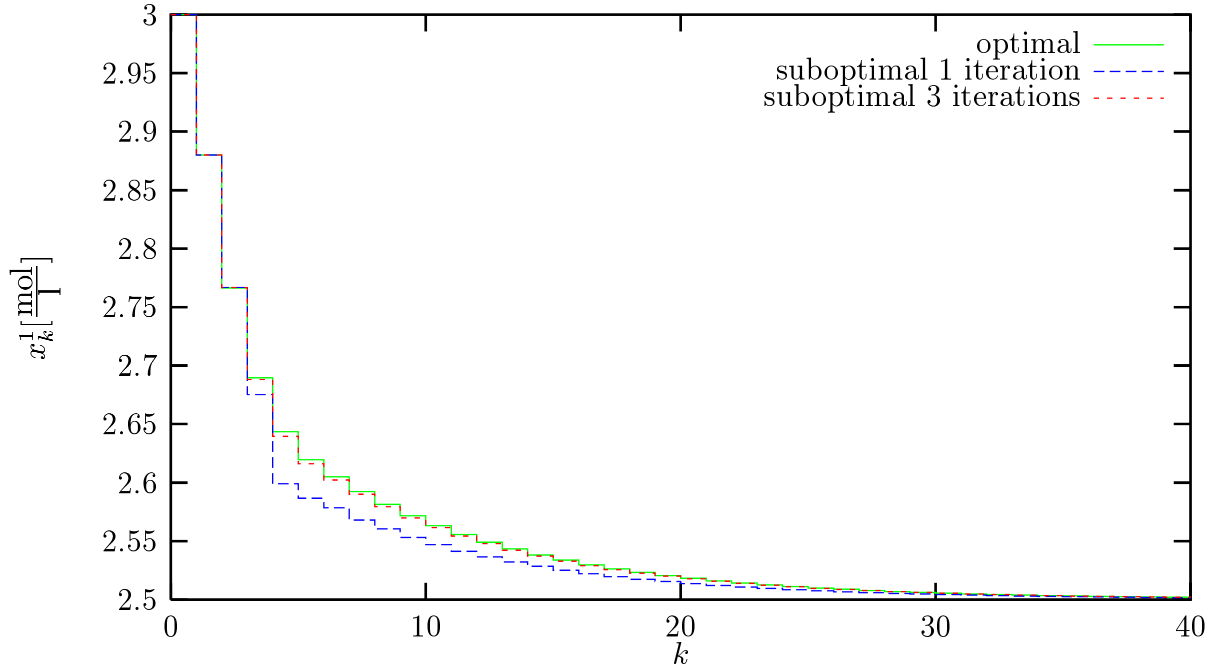

Figure 4.5: Comparison of the controlled system state $x^1 = c_A$, $N = 11$

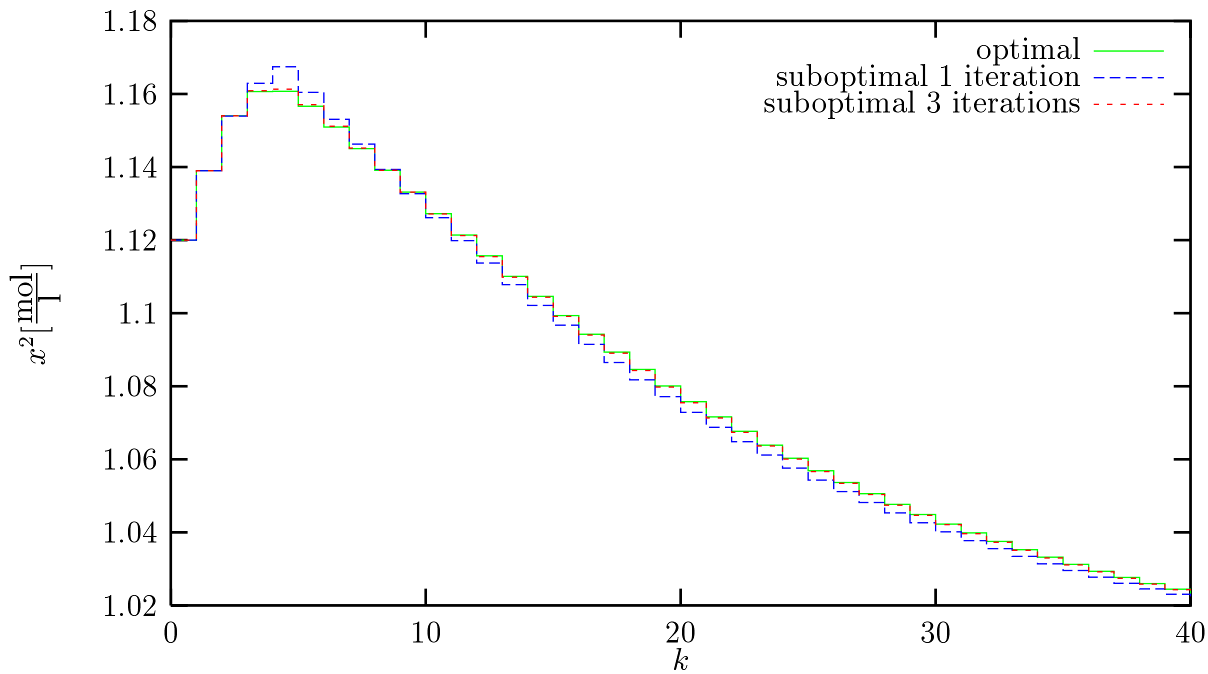

Figure 4.6: Comparison of the controlled system state $x^2 = c_B$, $N = 11$

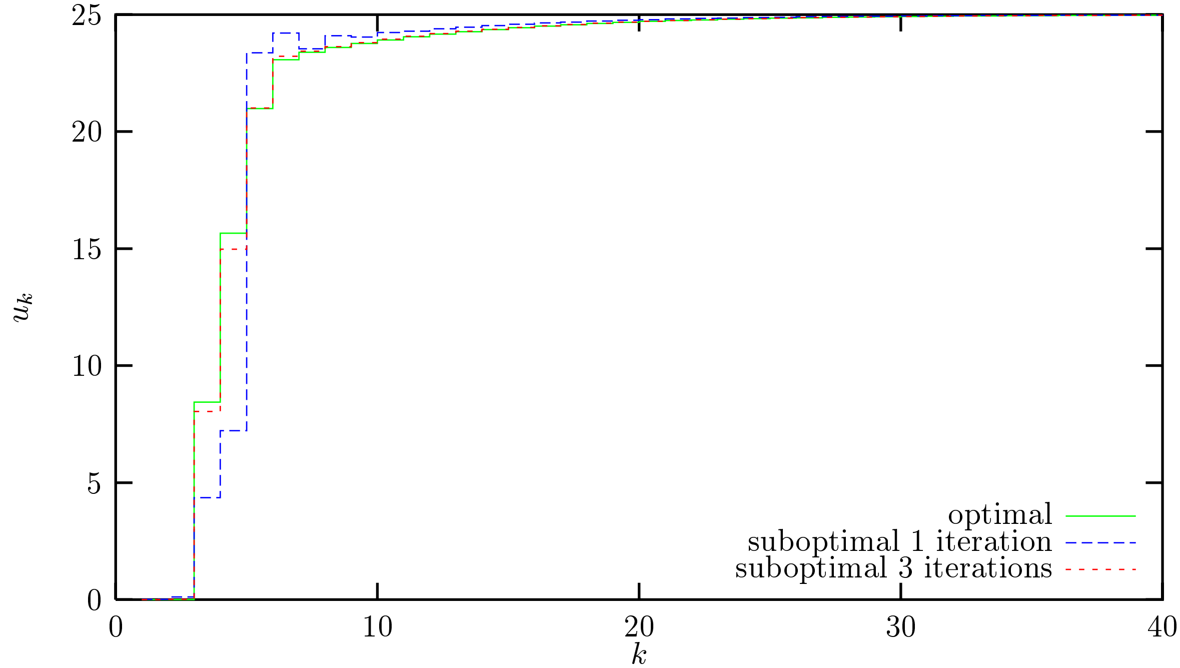

Figure 4.7: Comparison of the value of manipulated variable u, $N = 11$

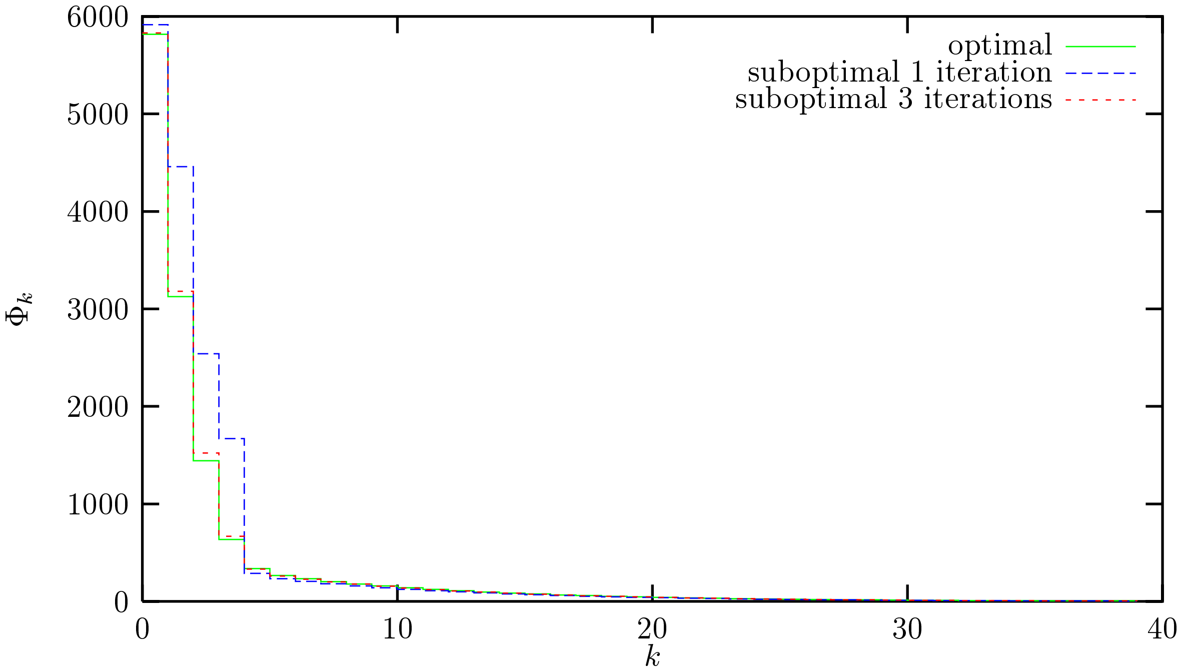

Figure 4.8: The changes of the cost function with time

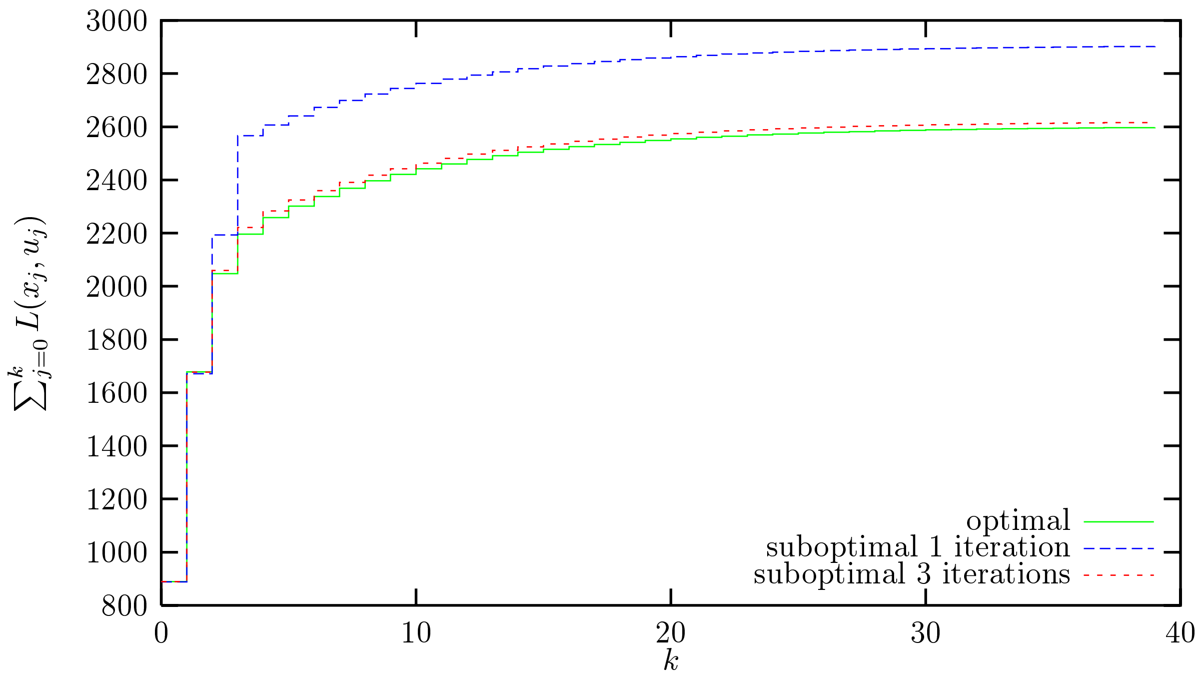

Figure 4.9: Sum of stage cost $L(x_k, u_k)$ up to time k

Chapter 5

Conclusions

This work has focused on two main goals: providing a review of the state of the art MPC concepts for linear and nonlinear systems including their main advantages or disadvantages; and deriving a new nonlinear MPC algorithm which overcomes some of the encountered problems.

During the review section it was shown that the mathematical transfer of ideas between linear and nonlinear MPC is surprisingly easy. We have pointed out that the during the solution of MPC for nonlinear plants encountered optimization is the biggest hurdle which has to be overcome before NMPC algorithms can be applied in practice. We have shown that there have been already some algorithm proposed, which avoid the resulting global optimization problem and relax the terminal state equality constraint to an terminal inequality constraint. The now encountered nonlinear programs or feasibility problems are easier to solve. One of the main disadvantages of these proposed algorithms is that they require a switching to a user supplied stabilizing controller. This can lead, to slow behavior of the system or an decrease of the performance since the applied controller does not have to be "optimal." Here from we concluded that an algorithm which does not require a global or local optima and a terminal state equality constraint and which does not switch to a different controller might lead to better performance.

We derived such a controller during the second part of the work, based on the "ideal" MPC controller with infinite prediction and control horizon. We considered a general nonlinear cost function. The infinite dimensional optimization problem was reduced to a solvable finite dimensional one by applying after the control horizon length N in a region the, the nonlinear system stabilizing linear control law. It was shown, basing on an earlier work of Mayne and Michalska [43], that such a controller for a specific class of system exists. The contained systems in this class must have at the origin a stabilizable linearization. For this class of systems it is possible to find an upper bound on the remaining cost function and guarantee feasibility of the next step under nominal conditions. We outlined an algorithm which can provide the region in which this is possible and showed how to calculate the now necessary quadratic terminal state penalty. To make an application of the linear controller at the end possible, we had to include in the optimization problem that the final prediction state has to lay in the region in which the linear controller guarantees stability. A proof of stability and convergence was given and the proposed concept was later on expanded, so that even a suboptimal but feasible sequence of inputs leads to an stabilizing controller. It was shown that for this it was necessary to modify the algorithm to guarantee continuity of the resulting cost function at the origin. We also proposed a version of the controller with respect to model mismatch and disturbances. This approach was motivated by the robust dual mode concept of Mayne and Michalska [43] and the suboptimal dual mode MPC concept by Scokaert et al. [56]. We concluded the work with a presentation of possible efficient solution methods for the encountered feasibility/optimization problems. We showed that the FSQP optimizer of Tits et al. ([47],[49],[48]) can provide feasible descent sub steps which can be used in the Suboptimal MPC concepts. An interface between this optimizer and Octave was programmed. The resulting code was tested on two instructional rather than sophisticated example systems. We have seen that the proposed algorithm can lead to significant computational savings while providing a stabilizing solution.

This works raises a number of theoretical and implementation issues that are unsolved or only partly solved.

- It is not clear what the best way is to calculate a terminal region $\hat{\mathcal{B}}$ with maximal size for the proposed approach. One starting point might be Hauser and Lai's [18] algorithm proposed for continuous time.
- It has been shown that the efficient implementation of the proposed algorithm is possible. The currently available implementation, however, should be seen more as conceptual than as a finished solution. Especially the numerically efficiency and robustness should be improved.
- The suggested controller should be employed on some more realistic example systems to confirm these results.
- The applied FSQP algorithm is very inefficient compared structured SQP algorithm. One step requires the solution of 3 QP's and 1 line search. It might be possible to combine both proposed optimization methods to achieve a fast, feasible descent algorithm.

Appendix A

A.1 Existence of $\hat{\mathcal{B}}$, $\tilde{Q}$

From section 3.2 it remains to show that if the linearization of

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

is stabilizable at the origin, that we can find an set $\hat{\mathcal{B}}$ and a static linear control law K so that $\hat{\mathcal{B}}$ is a positive invariant set for $x_{k+1} = f(x_k, Kx_k)$. Additional there must exist an $\tilde{Q}$ so that

$$\sum_{j=k}^{\infty} L(x_j, Kx_j) \leq x_k^T \tilde{Q} x_k \quad \forall x \in \hat{\mathcal{B}} \tag{A.2}$$

with $\tilde{Q}$ positive definite and

$$x_{j+1}^T \tilde{Q} x_{j+1} - x_j^T \tilde{Q} x_j \leq -x_j^T (Q + K^T R K) x_j \qquad \forall x_j \in \hat{\mathcal{B}} \tag{A.3}$$

We will follow a similar proof for continuous time presented by Michalska and Mayne [43]. A corresponding proof can be also found in the book by Sontag [63] pp. 171–172. The proof will be based on an Lyapunov argument. During the following we assume that $x \in \mathcal{W}$. The Lyapunov function will be the quadratic solution of the following sum for the linearized system.

$$V_k = \sum_{j=k}^{\infty} (x_j^L)^T Q (x_j^L) + (x_j^L)^T K^T R K (x_j^L) \tag{A.4}$$

K so that $\tilde{A} := A + BK$ is stable, Q, R given by A9 section 3.1. V_j for the linearized system $x_{j+1}^L = A x_j^L + B u_j^L$ under $u_j^L = K x_j^L$ can be calculated by solving the following discrete Lyapunov equation.

$$(A + BK)^T P (A + BK) - P = -(Q + K^T R K) \tag{A.5}$$

Here P is the positive definite solution and our Lyapunov function becomes

$$V_k = x_k^T P x_k \tag{A.6}$$

Similar to equation (2.11) in [43] we are interested in how the value of the Lyapunov function for the nonlinear, by K controlled, system is evolving for points near the origin.

$$V_{k+1} - V_k = x_{k+1}^T P x_{k+1} - x_k^T P x_k \tag{A.7}$$

For simplicity we introduce e as the difference between the linearized system x_j^L and the nonlinear system x_j under the linear control law $u = Kx$.

$$\begin{aligned} e_k &:= f(x_k, Kx_k) - (A + BK)x_k && \text{(A.8)} \\ &= f(x_k, Kx_k) - \tilde{A}x_k && \text{(A.9)} \end{aligned}$$

We will start with the assumption of the constraint free case. This leads to:

$$\begin{aligned} V_{k+1} - V_k &= x_{k+1}^T P x_{k+1} - x_k^T P x_k && \text{(A.10)} \\ &= e_k^T P e_k + e_k^T P \tilde{A} x_k + x_k^T \tilde{A}^T P e_k + x_k^T \tilde{A}^T P \tilde{A} x_k - x_k^T P x_k && \text{(A.11)} \\ &\overset{A.5}{=} e_k^T P e_k + 2 e_k^T P \tilde{A} x_k - x_k^T (Q + K^T R K) x_k && \text{(A.12)} \\ &\leq \underbrace{e_k^T P e_k}_{O(\|x\|)^4} + \underbrace{|2 e_k^T P \tilde{A} x_k|}_{O(\|x\|)^4} - \underbrace{x_k^T (Q + K^T R K) x_k}_{O(\|x\|)^2} && \text{(A.13)} \end{aligned}$$

We can conclude here from that it is possible to find a $\beta > 0$ so that the set $\hat{\mathcal{B}} := \{x : \|x\|_P \leq \beta\} \subset \mathcal{W}$ for a given constant $\alpha \in (0, 1)$ is not empty and the following holds:

$$x_{k+1}^T P x_{k+1} - x_k^T P x_k \leq -\alpha x_k^T (Q + K^T R K) x_k \tag{A.14}$$

Because P and $(Q + K^T RK)$ are positive definite we can conclude, that $\hat{\mathcal{B}}$ is positive invariant for $x_{j+1} = f(x_j, Kx_j)$. After division by α we get with $\tilde{Q} := \frac{P}{\alpha}$

$$x_{k+1}^T \tilde{Q} x_{k+1} - x_k^T \tilde{Q} x_k \leq -x_k^T (Q + K^T R K) x_k \tag{A.15}$$

this establishes (A.3). Since we assumed that $x_k \in \mathcal{W}$ we know from A9 section 3.1 that:

$$L(x, Kx) \leq x^T Q x + x^T K^T R K x \quad \forall x \in \mathcal{W} \tag{A.16}$$

This and (A.15) lead to (A.2).
In the constrained case we have to ensure that $\hat{\mathcal{B}} \subset \mathcal{X}$ and that $u = Kx \in \mathcal{U}\ \forall x \in \hat{\mathcal{B}}$. This is possible, since we can make β as small as necessary. Michalska and Mayne [43] offer for a similar case the following algorithm for the calculation of $\hat{\mathcal{B}}$.

STEP 0:

- given $\mathcal{W}$, Q, R: choose K so that $A + BK$ is stable; choose $\alpha \in (0, 1)$
- calculate P from $(A + BK)^T P (A + BK) - P = -(Q + K^T R K)$

STEP 1:

- find a feasible solution of
  $e^T Pe + |2e^T P\tilde{A}x| - x^T(Q + K^T RK)x^T + \alpha x^T(Q + K^T RK)x \leq 0$,
  $d(Kx,\mathcal{U}) \leq 0$ and $d(x,\mathcal{X}) \leq 0$
  for all $x \in \hat{\mathcal{B}} := \{x : \|x\|_P \leq \beta\}$ where $\beta \geq 0$.

$d(Kx,\mathcal{U})$ stands for the distance of a point $u = Kx$ to the set $\mathcal{U}$, similar $d(x,\mathcal{X})$ stands for the distance of a point x to the set $\mathcal{X}$. The resulting problem is a semi-infinite feasibility problem, since $\beta \geq 0$. The size of the produced set $\hat{\mathcal{B}}$ depends on the chosen K and α. It is not totally clear how this parameters should be "tuned" to achieve a possible large set $\hat{\mathcal{B}}$. Chen and Allgöwer [7] proposed as similar algorithm for continuous time systems, which additional optimizes over a parameter to make $\hat{\mathcal{B}}$ as large as possible .

Another interesting approach which could provide a maximal sized set $\hat{\mathcal{B}}$ by taking K in consideration could possibly be derived from an algorithm proposed by Hauser and Lai [18].

Bibliography

[1] Mazen Alamir and Guy Bornard. Stability of a truncated infinite constrained receding horizon scheme: The general discrete nonlinear case. *Automatica*, 31:1352–1356, 1995.

[2] E. Arnold, P. Tatjewski, and P. Wolochowicz. An SQP-type solution method for constrained discrete-time optimal control problems. In R. Burlirsch and D. Kraft, editors, *Computational Optimal Control*, pages 127–136. Birkhäuser, 1994.

[3] E. Arnold, P. Tatjewski, and P. Wolochowicz. Two methods for large-scale nonlinear optimization and their comparison on a case study of hydropower optimization. *Journal of Optimization Theory and Applications*, 81(2):221–248, 1994.

[4] Thomas A. Badgwell. Robust stability constraints for model predictive control: A unifying review and some new results. Presented at the 1996 AICHE meeting, Chicago, Illinois, Paper 1141, 1996.

[5] Robert R. Bitmead, Micheal Gevers, and Vincent Wertz. *Adaptive Optimal Control, The Thinking Man's GPC*. Prentice-Hall, Englewood Cliffs, New Jersey, 1990.

[6] Paul T. Boggs and Jon W. Tolle. Sequential Quadratic Programming. *Acta Numerica*, pages 1–52, 1995.

[7] H. Chen and F. Allgöwer. A quasi-infinite horizon nonlinear model predictive control scheme for constrained nonlinear systems. In *CCC, Qingdiao/China*, pages 309–316, 1996.

[8] H. Chen and F. Allgöwer. A quasi-infinite horizon nonlinear model predictive control scheme for constrained nonlinear systems. To be published in *Automatica*, 1997.

[9] D. Chmielewski and V. Manousiouthakis. On constrained infinite-time linear quadratic optimal control. Report ENG 95–150, UCLA School of Engineering and Applied Science (also submitted to *Systems and Control Letters*), October 1995.

[10] Jian L. Zhou Craig Lawrence and Andre L. Tits. User's guide for CFSQP version 2.4: A C code for solving (large scale) constrained nonlinear (minimax) optimization problems, generating iterates satisfying all inequality constraints. Technical report, Electrical Engineering Department and Institute for Systems Research, University of Maryland, College Park, MD 20742, 1996.

[11] C. R. Cutler and B. L. Ramaker. Dynamic matrix control—a computer control algorithm. In *Proceedings of the Joint Automatic Control Conference*, 1980.

[12] Simone L. de Oliveira, Vesna Nevistić, and Manfred Morari. Control of nonlinear systems subject to input constraints. In *IFAC Symposium on Nonlinear Control System Design, Tahoe City, California*, pages 15–20, 1995.

[13] Simone Loureiro de Oliveira. *Model predicitve control for unconstraint nonlinear systems*. PhD thesis, California Institue of Technology (Caltech), California, USA, 1996.

[14] J.C. Dunn and D.P. Bertsekas. Efficient dynamic programming impementations of newton's method for unconstrained optimal control problems. *Journal of Optimization Theory and Applications*, 63(1):23–38, 1989.

[15] Carlos E. García and A. M. Morshedi. Quadratic programming solution of dynamic matrix control (QDMC). *Chemical Engineering Communications*, 46:73–87, 1986.

[16] Carlos E. García, David M. Prett, and Manfred Morari. Model predictive control: Theory and practice—a survey. *Automatica*, 25(3):335–348, 1989.

[17] Elmer G. Gilbert and Kok Tin Tan. Linear systems with state and control constraints: The theory and application of maximal output admissible sets. *IEEE Transactions on Automatic Control*, 36(9):1008–1020, September 1991.

[18] J. Hauser and M. C. Lai. Estimating quadratic stability domains by nonsmooth optimization. In *Proceedings of American Control Conference, Chicago, IL*, pages 571–576, 1992.

[19] Michael A. Henson and Dale E. Seborg. An internal model control strategy for nonlinear systems. *AIChE Journal*, 37(7):1065–1081, 1991.

[20] Michael A. Henson and Dale E. Seborg. Theoretical analysis of unconstrained nonlinear model predictive control. *International Journal of Control*, 58(5):1053–1080, 1993.

[21] Alberto Isidori. *Nonlinear Control Systems.* Springer-Verlag, Berlin, 2nd edition, 1989.

[22] R. E. Kalman. Contributions to the theory of optimal control. *Bull. Soc. Math. Mex.*, 5:102–119, 1960.

[23] R. E. Kalman and J. E. Bertram. Control system analysis and design via the "Second method" of Lyapunov, Part II: Discrete–time systems. *ASME Journal of Basic Engineering.*, pages 394–400, June 1960.

[24] R. E. Kalman and J. E. Bertram. Control system analysis and design via the "Second method" of Lyapunov, Part I: Continuous–time systems. *ASME Journal of Basic Engineering.*, pages 371–393, June 1960.

[25] S. S. Keerthi and E. G. Gilbert. An existence theorem for discrete-time infinite-horizon optimal control problems. *IEEE Transactions on Automatic Control*, 30(9):907–909, September 1985.

[26] 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 Optimization Theory and Applications*, 57(2):265–293, May 1988.

[27] Sathiya S. Keerthi. *Optimal Feedback Control of Discrete-Time Systems with State-Control Constraints and General Cost Functions.* PhD thesis, University of Michigan, 1986.

[28] D. L. Kleinman. An easy way to stabilize a linear constant system. *IEEE Transactions on Automatic Control*, 15(12):692, December 1970.

[29] Huibert Kwakernaak and Raphael Sivan. *Linear Optimal Control Systems.* John Wiley and Sons, New York, 1972.

[30] W. H. Kwon and A. E. Pearson. On feedback stabilization of time-varying discrete linear systems. *IEEE Transactions on Automatic Control*, 23(3):479–481, June 1978.

[31] Wook Hyun Kwon. Advances in predictive control: Theory and applications. In *1996 Workshop on Model Predictiv Control in Seoul*, pages 1–43, 1996.

[32] Jay H. Lee. Recent advances in model predictive control and other related areas. Fifth International Conference on Chemical Process Control, Lake Tahoe, California, January 1996.

[33] Nuno M.C. de Oliveira Lino O. Santos and Lorenz T. Biegler. Reliable and efficient optimization strategies for nonlinear model predictive control. In *Proceedings of the 1995 IFAC DYCORD Symposium, Copenhagen, Denmark*, pages 33–38, 1995.

[34] Olivi L. Magnasarian. Nonlinear programming. Course notes, University of Wisconsin, Madison, Spring 1996.

[35] D. Q. Mayne and H. Michalska. Receding horizon control of non-linear systems without differentiability of the optimal value function. *Systems & Control Letters*, 16:123–130, 1991.

[36] David Q. Mayne. Optimization in model based control. In *Proceedings of the IFAC Symposium Dynamics and Control of Chemical Reactors, Distillation Columns and Batch Processes*, pages 229–242, Helsingor, Denmark, June 1995.

[37] David Q. Mayne. Nonlinear model predictive control: an assessment. Fifth International Conference on Chemical Process Control, Lake Tahoe, California, January 1996.

[38] David Q. Mayne and Hannah Michalska. Receding horizon control of nonlinear systems. *IEEE Transactions on Automatic Control*, 35(7):814–824, July 1990.

[39] Edward S. Meadows. *Stability and Continuity of Nonlinear Model Predictive Control*. PhD thesis, The University of Texas at Austin, 1994.

[40] Edward S. Meadows, Michael A. Henson, John W. Eaton, and James B. Rawlings. Receding horizon control and discontinuous state feedback stabilization. *International Journal of Control*, 62(5):1217–1229, 1995.

[41] Edward S. Meadows, Kenneth R. Muske, and James B. Rawlings. Implementable model predictive control in the state space. In *Proceedings of the 1995 American Control Conference*, pages 3699–3703, 1995.

[42] Edward S. Meadows and James B. Rawlings. Model predictive control. In Michael A. Henson and Dale E. Seborg, editors, *Nonlinear Process Control*. Prentice Hall, 1996.

[43] Hanna Michalska and David Q. Mayne. Robust receding horizon control of constrained nonlinear systems. *IEEE Transactions on Automatic Control*, 38(11):1623–1633, 1993.

[44] Kenneth R. Muske. *Linear Model Predictive Control of Chemical Processes*. PhD thesis, The University of Texas at Austin, 1995.

[45] Kenneth R. Muske and James B. Rawlings. Model predictive control with linear models. *AIChE Journal*, 39(2):262–287, 1993.

[46] Henk Nijmeijer and Arjan J. van der Schaft. *Nonlinear Dynamical Control Systems*. Springer-Verlag, New York, 1990.

[47] E. R. Panier and A. L. Tits. A superlinearly convergent feasible method for the solution of inequality constrained optimization problems. *SIAM Journal on Control and Optimization*, 25(4):934–950, 1987.

[48] E. R. Panier and A. L Tits. On combining feasibility, descent and superlinear convergence in inequality constrained optimization. *Mathematical Programming*, 59:261–276, 1993.

[49] E. R. Panier, A. L. Tits, and J. N. Herskovits. A qp-free, globally convergent, locally superlinearly convergent algorithm for inequality constrained problems. *SIAM Journal on Control and Optimization*, 26(4):788–811, 1988.

[50] E. Polak and D. Q. Mayne. Design of nonlinear feedback controllers. *IEEE Transactions on Automatic Control*, 26(3):1083–1112, 1981.

[51] S. Joe Qin and Thomas A. Badgwell. An overview of industrial model predictive control technology. Fifth International Conference on Chemical Process Control, Lake Tahoe, California, January 1996.

[52] Christopher V. Rao, Stephen J. Wright, and James B. Rawlings. On the application of interior point methods to model predictive control. To be submitted*Journal of Optimization Theory and Application*, February 1997.

[53] James B. Rawlings, E. Scott Meadows, and Kenneth R. Muske. Nonlinear model predictive control: A tutorial and survey. In *ADCHEM '94 Proceedings, Kyoto, Japan*, 1994.

[54] James B. Rawlings and Kenneth R. Muske. Stability of constrained receding horizon control. *IEEE Transactions on Automatic Control*, 38(10):1512–1516, October 1993.

[55] J. Richalet, A. Rault, J. L. Testud, and J. Papon. Model predictive heuristic control: Applications to industrial processes. *Automatica*, 14:413–428, 1978.

[56] Pierre O.M. Scokaert, David Q. Mayne, and James B. Rawlings. Suboptimal model predictive control. Submitted for publication in *IEEE Trans. Auto. Cont.*, April 1996.

[57] Pierre O.M. Scokaert and James B. Rawlings. Constrained linear quadratic regulation. Accepted for publication in *IEEE Transactions on Automatic Control*, December 1995.

[58] Pierre O.M Scokaert and James B. Rawlings. Infinite horizon linear quadratic control with constraints. In *Proceedings of the 13th IFAC World Congress*, San Francisco, California, July 1996.

[59] Pierre O.M. Scokaert and James B. Rawlings. On infeasibilities in model predictive control. In Jeffrey C. Kantor, editor, *Proceedings of Chemical Process Control – V*. CACHE, 1996.

[60] Pierre O.M. Scokaert, James B. Rawlings, and Edward S. Meadows. Discrete-time stability with perturbations: Application to model predictive control. Accepted for publication in *Automatica*, 1996.

[61] P. B. Sistu, R. S. Gopinath, and B. Wayne Bequette. Computational issues in nonlinear predictive control. *Computers & Chemical Engineering*, 17(4):361–366, 1993.

[62] Phani B. Sistu and B. Wayne Bequette. Nonlinear model-predictive control: Closed-loop stability analysis. *AIChE Journal*, 42(12):3388–3402, 1996.

[63] Eduardo D. Sontag. *Mathematical Control Theory*. Springer-Verlag, New York, 1990.

[64] Masoud Soroush and Costas Kravaris. A continuous–time formulation of nonlinear model predictive control. In *Proceedings of the 1992 American Control Conference*, pages 1561–1567, 1992.

[65] Marc C. Steinbach. A structured interior point SQP method for nonlinear optimal control problems. In R. Burlirsch and D. Kraft, editors, *Computational Optimal Control*, pages 213–222. Birkhäuser, 1994.

[66] Mario Sznaier and Mark J. Damborg. Suboptimal control of linear systems with state and control inequality constraints. In *Proceedings of the 26th Conference on Decision and Control*, pages 761–762, 1987.

[67] E.D. Gilles und U. Knöpp. Skriptum zur Vorlesung Regelungstechnik II, Analyse und Synthese von Regelungssystemen im Zustandsraum. chapter XVI, pages 145–190. Universität Stuttgart, Institut für Systemdynamik und Regelungstechnik, 1993.

[68] Mathukumalli Vidyasagar. *Nonlinear Systems Analysis.* Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 2nd edition, 1993.

[69] Stephen J. Wright. Applying new optimization algorithms to model predictive control. Fifth International Conference on Chemical Process Control, Lake Tahoe, California, January 1996.

[70] Stephen J. Wright. A path-following interior algorithm for linear and quadratic optimization problems. *Preprint MCS-P401-1293*, Mathematics and Computer Science Division, Argonne National Laboratory, Argonne, Ill. To appear in *Annals of Operations Research*, December, 1993.

[71] Z.Q. Yang and E. Polak. Moving horizon control of nonlinear systems with input saturation, disturbances and plant uncertainty. *International Journal of Control*, 58:875–903, 1993.

[72] A. Zheng and M. Morari. Control of linear unstable systems with constraints. In *American Control Conference, Seattle, Washington*, pages 3704–3708, 1995.

[73] Zhi Q. Zheng and Manfred Morari. Robust stability of constrained model predictive control. In *Proceedings of the 1993 American Control Conference*, pages 379–383, 1993.

Vita

Rolf Heinz Findeisen was born in Nürtingen, Germany on February 5, 1970, as the third son of Konrad Findeisen and Renate Findeisen.
He graduated from Max Eyth Technisches Gymnasium, Kirchheim unter Teck, Germany in 1990. After this he fulfilled his compulsory military service as an maintenance soldier in the army base Großengstingen . He entered the University of Stuttgart in the fall of 1991 and received his Vordiplom in Technische Kybernetik in the summer of 1993. He studied for four more semesters on his Hauptdiplom of Technische Kybernetik at the University of Stuttgart before he left for an integrated exchange program with the University of Wisconsin-Madison in Fall 1995.

Permanent Address: Rolf Findeisen
Siemensstraße 10
D-72622 Nürtingen
Germany

This dissertation was prepared with LaTeX 2_ε[1] by the author.

[1]This particular University of Wisconsin complient style was carved from The University of Texas at Austin styles as written by Dinesh Das (LaTeX 2_ε), Khe–Sing The (LaTeX), and John Eaton (LaTeX). Knives and chisels wielded by John Campbell and Rock Matthews.