Real-time Model Predictive Control

Diss. ETH No. 19524

Real-time Model Predictive Control

A dissertation submitted to
ETH ZURICH

for the degree of
Doctor of Sciences

presented by

MELANIE NICOLE ZEILINGER

Dipl.-Ing., University of Stuttgart
born 09.05.1982
citizen of Germany

accepted on the recommendation of

Prof. Dr. Manfred Morari, examiner
Prof. Dr. Jan Maciejowski, co-examiner

2011

Real-time Model Predictive Control

Melanie N. Zeilinger

Dissertation ETH Zurich No. 19524

...to my mother.

Acknowledgments

I am grateful to a number of outstanding people who I have had the opportunity to meet, learn from and work with during my PhD and whose support and encouragement have made this exciting and challenging journey possible.

My first and special thanks go to my advisor Manfred Morari for giving me the opportunity to do my PhD at the Automatic Control Laboratory and for his continuous support and trust during my PhD. Thanks for guiding me through this time, it has been and will be an honor and great pleasure working with you.

I owe my deepest gratitude to Colin Jones for all his help and support, which had a major influence on my research and a big part in my PhD and this thesis. I was very lucky to be one of the first PhD students that he co-supervised and the list of skills I have learned from him is too long to be included here. Therefore, simply thanks for everything, for being a great teacher, for great research support, advice and for correcting this thesis.

My sincere thanks also go to Jan Maciejowski for being my co-examiner and, most importantly, for reading this thesis and providing very valuable comments.

I want to thank the people that have worked with me during my PhD for inspiring research discussions and introducing me to new ideas and topics, I look forward to continuing these collaborations in the future. Special thanks go to Davide for introducing me to the area of input-to-state stability, which turned out extremely valuable in this thesis. In the last year of my PhD, I had the pleasure to work with the junior PhD students, Alexander Domahidi, Georg Schildbach and Christian Conte, thanks for proof reading parts of this thesis, I look forward to exciting new research projects in the future.

The Automatic Control Laboratory is one of the best places I could have hoped for doing a PhD, it is an exceptional lab and research environment with amazing and interesting people from all over the world and I am grateful to Manfred Morari and John Lygeros for all the opportunities this has provided. At this point, I also want to thank Frank Doyle, without whom I might have not applied to this lab, thanks for being a great mentor during my early research days.

During my PhD, I have met not only great colleagues but also very good friends.

In particular, I want to thank Frank Christophersen for his help in getting started in the lab and in research, Philipp Rostalski for his advice in research and teaching; Akin Sahin for being a great coffee and salsa buddy; Yannis Lymperopolous for making IfA a loud and happy place; Urban Mäder for great discussions and for his refreshing honesty; Stefan Richter for always having an open door and time to discuss research problems, for proof reading parts of this thesis and for an amazing research trip to California and Frauke Oldewurtel, my female support at IfA, for unforgettable conference and workshop travels. I want to thank Frank Christophersen, Mirko Fiacchini, Alessandra Parisio, Miroslav Baric, Stefan Almer and Daniel Axehill for being great office mates (K13.1 and K13.2 basically count as one office). Thanks to all present and past members for making this a special place, it was a great time and I wish you all the best for your future.

Finally, my special thanks go to my family and friends, without them I would not have made it here. Thanks for understanding that time was often scarce and for always encouraging me. I want to thank my mother for her love and endless support, thanks for always being there for me; Gabriel for simply everything, thanks for your understanding and support in all these crazy days and for reading this thesis (you probably know more about real-time MPC than you should ever need); Pamela and Mirjam for distracting me when it was necessary and for just being who you are (Pam, your survival package was crucial in finishing this thesis), the rest of my 'family' in Zurich and all other friends in the world.

After all, finishing the PhD is not the end of a journey, but the beginning of a much bigger one.

Melanie Zeilinger
Zürich, Summer 2011

Contents

Abstract

The main theme of this thesis is the development of real-time and soft constrained Model Predictive Control (MPC) methods for linear systems, providing the essential properties of closed-loop feasibility and stability. MPC is a successful modern control technique that is characterized by its ability to control constrained systems. In practical implementations of MPC, computational requirements on storage space or online computation time have to be considered in the controller design. As a result, the optimal MPC control law can often not be implemented and a suboptimal solution has to be provided that is tailored to the system and hardware under consideration and meets the computational requirements. Existing methods generally sacrifice guarantees on constraint satisfaction and/or closed-loop stability in such a real-time environment. In addition, enforcing hard state constraints in an MPC approach can be overly conservative or even infeasible in the presence of disturbances. A solution commonly applied in practice is to relax some of the constraints by means of so-called soft constraints. Current soft constrained approaches for finite horizon MPC, however, do not provide a guarantee for closed-loop stability.

This thesis addresses these limitations and aims at reducing the gap between theory and practice in MPC by making three main contributions: A real-time MPC method based on a combination of explicit approximation and online optimization that offers new tradeoff possibilities in order to satisfy limits on the storage space and the available computation time and provides hard real-time feasibility and stability guarantees; a real-time MPC approach that is based on online optimization and provides these properties for any time constraint while allowing for extremely fast computation; a soft constrained method based on a finite horizon MPC approach that guarantees closed-loop stability even for unstable systems.

First, two methods are presented that consider the application of MPC to high-speed systems imposing a hard real-time constraint on the computation of the MPC control law. There are generally two main paradigms for the solution of an MPC problem: In online MPC the control action is obtained by executing an optimization online, while in explicit MPC the control action is pre-computed and stored offline. Limits on the storage space or the computation time have therefore restricted the applicability of MPC in many practical problems. This thesis introduces a new approach, combining the two paradigms of explicit and online MPC in order to overcome their individual limitations. The use of an offline approximation together with warm-start techniques from online optimization allows for a tradeoff between the warm-start and online computational effort. This offers new possibilities in satisfying system requirements on storage space and online computation time. A preprocessing analysis is introduced that provides hard real-time execution, stability and performance guarantees for the proposed controller and can be utilized to identify the best solution method for a considered application and set of requirements.

By using explicit approximations, the first real-time approach is best suited for small or medium size problems. In contrast, the second real-time MPC approach presented in this thesis is solely based on online optimization and can be practically implemented and efficiently solved for large-scale dynamic systems. A hard real-time constraint generally prevents the computation of the optimal solution to the MPC problem, which can lead to constraint violation, and more importantly, instability when using a general optimization solver. The proposed method is based on a robust MPC scheme and recovers guarantees on feasibility and stability in the presence of additive disturbances for any given time constraint. The approach can be extended from regulation to tracking of piecewise constant references, which is required in many applications. All computational details needed for an implementation of a fast MPC method are provided and it is shown how the structure of the resulting optimization problem can be exploited in order to achieve computation times equal to, or faster than those reported for methods without guarantees.

One of the main difficulties in real-time MPC methods is the initialization with a feasible solution. This motivates the investigation of soft constrained MPC schemes and their robust stability properties in the final part of this thesis. The relaxation of state and/or output constraints in a standard soft constrained approach generally leads to a loss of the stability guarantee in MPC, relying on the use of a terminal state constraint. In this thesis a new soft constrained MPC method is presented that provides closed-loop stability even for unstable systems. The proposed approach significantly enlarges the region of attraction and preserves the optimal behavior with respect to the hard constrained MPC control law whenever all constraints can be enforced. By relaxing state constraints, robust stability in the presence of additive disturbances can be provided with an enlarged region of attraction compared to a robust MPC approach considering the same disturbance size. In order to allow for a more flexible disturbance handling, the proposed soft constrained MPC scheme can be combined with a robust MPC framework and the theoretical results directly extend to the combined case.

Zusammenfassung

Das Thema dieser Dissertation ist die Entwicklung von modellprädiktiven Regelungsmethoden (engl.: Model Predictive Control, MPC) welche die Berechnung des Reglers in Echtzeit und mit relaxierten Beschränkungen ermöglichen und dabei die Lösbarkeit des MPC Problems und Stabilität des geschlossenen Regelkreises gewährleisten. MPC ist ein modernes Regelungsverfahren, das sich dadurch auszeichnet, dass Beschränkungen explizit in der Berechnung des Reglers berücksichtigt werden können. In praktischen Implementierungen von MPC müssen die Anforderungen der jeweiligen Anwendung an den Speicherplatz oder die Rechenzeit im Reglerdesign berücksichtigt werden. Es ist daher oft nicht möglich den optimalen MPC Regler zu implementieren und ein suboptimaler Regler muss stattdessen berechnet werden, der auf das betrachtete System und die Hardware zugeschnitten ist und die Rechenzeitanforderung erfüllt. Bislang verfügbare Methoden nehmen in diesem Fall meist eine Verletzung der Beschränkungen und/oder den Verlust der Stabilitätsgarantie in Kauf. Darüber hinaus kann das Erzwingen der Beschränkungen oft zu konservativen Lösungen, oder in Anwesenheit von Störungen sogar zu unlösbaren MPC Problemen führen. In der Praxis werden daher oft die Beschränkungen relaxiert. Bisherige Methoden für MPC mit endlichem Zeithorizont und relaxierten Beschränkungen bieten jedoch keine Garantie für Stabilität des geschlossenen Kreises.

Diese Dissertation behandelt diese Problemstellungen und macht drei wichtige Beiträge: Eine echtzeitfähige MPC Methode, die explizite Approximationen mit online Optimierung kombiniert und dadurch neue Möglichkeiten bietet, Speicherplatz- und Rechenzeitanfoderungen zu erfüllen, während Garantien für die Einhaltung der Beschränkungen und Stabilität gewährleistet werden; eine echtzeitfähige MPC Methode, die auf online Optimierung basiert und diese Garantien für jede beliebige Zeitbeschränkung gewährleistet und zudem schnelle online Rechenzeiten erreicht; eine MPC Methode für relaxierte Beschränkungen und endliche Zeithorizonte, die Stabilität des geschlossenen Kreises auch für instabile Systeme garantiert.

Zu Beginn werden zwei Methoden präsentiert, welche die Anwendung von MPC auf schnelle Systeme betrachten und damit eine Beschränkung der verfügbaren Rechenzeit zur Berechnung des MPC Reglers. Es gibt generell zwei Lösungsansätze in MPC: in online MPC wird der Regler durch Lösen eines Optimierungsproblems online berechnet, während in explizitem MPC der Regler offline vorberechnet und gespeichert wird. Begrenzungen des Speichers oder der Rechenzeit schränken daher die Anwendbarkeit von MPC in vielen praktischen Problemen ein. Diese Arbeit stellt eine neue Methode vor, in der die beiden Ansätze explizites und online MPC kombiniert werden, um ihre jeweiligen Nachteile zu vermeiden. Die Verwendung einer offline Approximation zusammen mit warm-start Techniken der online Optimierung erlaubt es zwischen dem Rechenzeitaufwand für warm-start und dem für die online Optimierung abzuwägen. Dies bietet neue Möglichkeiten, um Anforderungen an den Speicherplatz oder die Rechenzeit zu erfüllen. Eine Methode zur Analyse des resultierenden Reglers wird vorgestellt, die es erlaubt Garantien für die Berechnung, Stabilität und Regelgüte in Echtzeit zu geben und welche ebenfalls verwendet werden kann, um die beste Lösungsmethode für eine bestimmte Anwendung und ihre Anforderungen zu identifizieren.

Da die erste hier vorgestellte Methode auf einer expliziten Approximation basiert, ist sie am Besten für kleinere oder mittelgrosse Probleme geeignet. Die zweite Methode, die in dieser Dissertation präsentiert wird, verwendet ausschliesslich online Optimierung und kann daher für grosse Probleme praktisch implementiert und effizient gelöst werden. Eine Beschränkung der Rechenzeit erlaubt es im Allgemeinen nicht, die optimale Lösung des MPC Problems zu berechnen, was bei Verwendung eines allgemeinen Optimierungsverfahrens zu einer Verletzung der Beschränkungen oder Instabiliät führen kann. Der entwickelte Ansatz verwendet ein robustes MPC Schema und garantiert die Einhaltung der Beschränkungen sowie Stabilität bei additiven Störungen für eine beliebige Beschränkung der Rechenzeit. Die Methode kann vom Problem der Regulierung auf das der Folgeregelung von stückweise konstanten Referenzsignalen erweitert werden, welche in vielen Anwendungen relevant ist. Alle erforderlichen Details für die Implementierung einer schnellen MPC Methode werden beschrieben und es wird gezeigt, wie die Struktur des resultierenden Optimierungsproblems ausgenutzt werden kann, um Berechnungszeiten zu erreichen, die gleich oder schneller sind als diejenigen, die für Methoden ohne jegliche Garantien gezeigt wurden.

Eine der Herausforderungen in echtzeitfähigem MPC ist die Initialisierung mit einer gültigen Lösung des MPC Problems. Dies motiviert die Untersuchung von MPC Ansätzen mit relaxierten Beschränkungen und ihrer robusten Eigenschaften im letzten Teil der Dissertation. Die Relaxierung von Zustands- oder Ausgangsbeschränkungen führt im Allgemeinen dazu, dass die Stabilitätsgarantie in MPC verloren geht, welche meist auf einer Beschränkung des Endzustandes basiert. In dieser Arbeit wird eine neue MPC Methode für relaxierte Beschränkungen vorgestellt, die Stabilität des geschlossenen Regelkreises auch für instabile Systeme garantiert. Die Region, in der Stabilität garantiert werden kann, wird vergrössert und Optimalität in Bezug auf den MPC Regler mit harten Beschränkungen bleibt erhalten, wenn alle Beschränkungen erfüllt werden können.

Durch das Relaxieren der Zustandsbeschränkungen kann robuste Stabilität gegenüber additiven Störungen in einem grösseren Bereich gewährleistet werden, als mit einem robusten MPC Schema für dieselbe Grösse der Störung. Um mehr Flexibilität in der Handhabung der Störungen zu erlauben, kann die Methode mit einem robusten Ansatz kombiniert werden, wobei alle theoretischen Ergebnisse auf die kombinierte Methode erweitert werden können.

Notation

Sets and spaces

$\{\bullet, \dots, \bullet\}$	a set or a sequence
$\emptyset$	the empty set
$\mathcal{N}(S)$	neighborhood of the set S
$\mathbb{R}$	set of real numbers
$\mathbb{R}^n$	set of n-dimensional (column) vectors with real entries
$\mathbb{R}^{n \times m}$	set of n by m matrices with real entries
$\mathbb{N}$	set of natural numbers (non-negative integers)
$\mathbb{N}_r$	set of natural numbers from 0 to r, $\mathbb{N}_r = (0 \dots, r)$ for $r \in \mathbb{N}$

Set Operators

Let $\mathcal{S}_1$, $\mathcal{S}_2 \subseteq \mathbb{R}^n$ be sets and $\{\mathcal{S}_i\}_{i=1}^b$ be a collection of sets.

$\lvert\mathcal{S}_1\rvert$	cardinality of $\mathcal{S}_1$
$\mathcal{S}_1 \cap \mathcal{S}_2$	set intersection, $\mathcal{S}_1 \cap \mathcal{S}_2 = \{s \mid s \in \mathcal{S}_1 \text{ and } s \in \mathcal{S}_2\}$
$\mathcal{S}_1 \cup \mathcal{S}_2$	set union, $\mathcal{S}_1 \cup \mathcal{S}_2 = \{s \mid s \in \mathcal{S}_1 \text{ or } s \in \mathcal{S}_2\}$
$\bigcup_{i=1}^b \mathcal{S}_i$	union of b sets, $\bigcup_{i=1}^b \mathcal{S}_i = \{s \mid s \in \mathcal{S}_1 \text{ or } \dots \text{ or } s \in \mathcal{S}_b\}$
$\mathcal{S}_1 \setminus \mathcal{S}_2$	set difference, $\mathcal{S}_1 \setminus \mathcal{S}_2 = \{s \mid s \in \mathcal{S}_1 \text{ and } s \notin \mathcal{S}_2\}$
$\mathcal{S}_1 \times \mathcal{S}_2$	Cartesian product, $\mathcal{S}_1 \times \mathcal{S}_2 = \{(s_1, s_2) \mid s_1 \in \mathcal{S}_1, s_2 \in \mathcal{S}_2\}$
$\mathcal{S}_1 \oplus \mathcal{S}_2$	Minkowski sum, $\mathcal{S}_1 \oplus \mathcal{S}_2 \triangleq \{s_1 + s_2 \mid s_1 \in \mathcal{S}_1, s_2 \in \mathcal{S}_2\}$
$\bigoplus_{i=1}^b \mathcal{S}_i$	Minkowski sum over b sets, $\bigoplus_{i=1}^b \mathcal{S}_i \triangleq \mathcal{S}_1 \oplus \mathcal{S}_2 \oplus \dots \oplus \mathcal{S}_b$
$\mathcal{S}_1 \ominus \mathcal{S}_2$	Minkowski difference, $\mathcal{S}_1 \ominus \mathcal{S}_2 \triangleq \{s \mid s + s_2 \in \mathcal{S}_1, s_2 \in \mathcal{S}_2\}$

Algebraic Operators

Let $A \in \mathbb{R}^{m \times n}$ be a matrix, $x \in \mathbb{R}^n$ be a vector and $\mathcal{E} \subseteq \mathbb{N}_m$, $\mathcal{F} \subseteq \mathbb{N}_n$ be sets.

$\mathrm{null}(A)$	nullspace of A, $\mathrm{null}(A) \triangleq \{x \in \mathbb{R}^n \mid Ax = 0\}$
$A_{\mathcal{E}}$	matrix formed by the rows of A whose indices are in the set $\mathcal{E}$
$A_{\cdot,\mathcal{F}}$	matrix formed by the columns of A whose indices are in the set $\mathcal{F}$
$A \succeq 0$	positive semi-definite matrix
$A \succ 0$	positive definite matrix
$[x]_+$	positive magnitude of a vector, $[x]_+ = \max\{0, x\}$ taken elementwise
$\lvert\cdot\rvert$	absolute value
$\lVert\cdot\rVert$	any vector norm
$\lVert\cdot\rVert_1$	l_1-norm or vector 1-norm (sum of absolute values)
$\lVert\cdot\rVert_2$	l_2-norm or vector 2-norm (Euclidian norm)
$\lVert\cdot\rVert_\infty$	l_∞-norm or vector ∞-norm (largest absolute element)
$\lVert\cdot\rVert_Q$	weighted 2-norm, $\lVert x\rVert_Q \triangleq \lVert Q^{\frac{1}{2}}x\rVert_2$

Control

n_x	number of states, $n_x \in \mathbb{N}$
n_u	number of inputs, $n_u \in \mathbb{N}$
x	state vector, $x \in \mathbb{R}^{n_x}$
u	input vector, $u \in \mathbb{R}^{n_u}$
$\mathbb{X}$	pre-specified set of state constraints, $\mathbb{X} \subseteq \mathbb{R}^{n_x}$
$\mathbb{U}$	pre-specified set of input constraints, $\mathbb{U} \subseteq \mathbb{R}^{n_u}$
$\mathbf{u}$	sequence of input vectors, $\mathbf{u} = [u_0, \ldots, u_N]$
$\mathbf{x}$	sequence of state vectors, $\mathbf{x} = [x_0, \ldots, x_N]$
u_j	j'th element of $\mathbf{u}$
$\mathbf{u}(x)$	sequence that depends on a parameter x
$u_j(x)$	j'th element of $\mathbf{u}(x)$

Other

I	identity matrix of appropriate dimension
0	zero matrix of appropriate dimension
$\mathbf{1}$	vector of ones of appropriate dimension, $\mathbf{1} \triangleq [1, \ldots, 1]^T$
$\mathrm{diag}(s)$	diagonal matrix whose diagonal entries are given by the vector s

Acronyms

LP	linear program
QP	quadratic program
QCQP	quadratically constrained program
SOCP	second-order cone program
SDP	semidefinite program
KKT	Karush-Kuhn-Tucker optimality condition
pLP	parametric linear program
pQP	parametric quadratic program
IPM	interior-point method
ASM	active set method
MPC	model predictive control
PI	positively invariant
RPI	robust positively invariant
MPI	maximal positively invariant
mRPI	minimal robust positively invariant

1 Introduction

Model Predictive Control (MPC) is a modern control technique that enjoys great success and widespread practical application due to its ability to control constrained systems. It is based on the computation of the optimal control action by solving a constrained finite horizon optimal control problem for the current state of the plant at each sample time. The theory of optimal MPC is well established and the optimal MPC control law provides closed-loop stability and constraint satisfaction at all times under certain assumptions on the problem setup.

Although part of the success of MPC as a sophisticated control method is based on its solid theoretical foundation, guarantees on feasibility and/or stability are often sacrificed in practical implementations. This motivates the work presented in this thesis with the focus of developing real-time and soft constrained methods that provide these essential properties. In practice, computational requirements on storage space or online computation time have to be considered for the particular application of interest. High-speed applications, for example, impose a hard real-time constraint on the solution of the MPC problem, i.e. a limit on the computation time that is available to compute the control input. The goal in *real-time MPC* is to deliver an MPC control law that is tailored to the system and hardware under consideration and meets the computational requirements, in particular in terms of the real-time constraint. In addition, state constraints are often relaxed in practice by means of so-called soft constraints that provide feasibility of the optimization problem when hard state constraints cannot be enforced, e.g. in the presence of disturbances. The goal in *soft constrained MPC* is to provide a guarantee for closed-loop stability with a large region of attraction, such that disturbances can be tolerated.

Classically the MPC problem is solved by executing an optimization routine online, which has restricted the application of model predictive controllers to slow dynamic processes. In recent years, various methods have been developed with the goal of enabling MPC to be used for fast sampled systems, which will be denoted as *fast MPC* methods in this thesis. These fast MPC approaches can generally be classified into two main paradigms: *explicit MPC* and *online MPC* methods. Depending on the particular problem properties and implementation restrictions, the user then has to decide for one of the two approaches.

Explicit MPC methods exploit the fact that the optimal solution to the MPC problem under certain assumptions on the problem setup is a piecewise affine function (PWA) defined over a polyhedral partition of the feasible states (see e.g. [BBM02, BMDP02, Bor03]. This so-called explicit solution can then be used as a control look-up table online, providing very low online computation times. The main limitation is, however, that the number of state-space regions over which the control law is defined, the so-called complexity of the partition, grows in the worst case exponentially due to the combinatorial nature of the problem [BBM02]. Thus, limits on the storage space or the computation time restrict the applicability of model predictive controllers in many real problems that are either computationally intractable for explicit MPC or the complexity of the explicit solution exceeds the storage capacity, while an online MPC solution cannot meet the required online computation times. This has given rise to an increasing interest in the development of new methods to either improve online optimization [Han00, KCR02, ART03, FBD08, WB10, CLK08, PSS10] or to approximate explicit solutions (e.g. [RWR98, LR06, YW02, BF03, JG03, BF06, JBM07]). While approximate explicit MPC methods are still limited to small or medium problem sizes due to their inherent explicit character, online MPC methods can be applied even to large-scale systems.

Recent results show that the computation times for solving an MPC problem can be pushed into a range where an online optimization becomes a reasonable alternative for the control of high-speed systems [WB10, FBD08]. Significant reduction of the online computation time can be achieved by exploiting the particular structure and sparsity of the optimization problem given by the MPC problem using tailored solvers. Available methods for fast online MPC, however, do not give guarantees on either feasibility or stability of the applied control action in a real-time implementation. The real-time MPC methods proposed in this thesis address this issue and provide hard real-time feasibility and stability guarantees. A method based on a combination of explicit approximation and online optimization for smaller or medium size problems is developed, as well as a real-time approach that is entirely based on online optimization and can be practically implemented and efficiently solved for all problem dimensions.

In addition to the satisfaction of computational requirements, an approach that is highly relevant and frequently applied in practice is soft constrained MPC. Soft constraints allow for a temporary relaxation of constraints when enforcing hard constraints would be infeasible and thereby provide feasibility of the optimization problem at all times. Soft constrained MPC methods and their robust stability properties gain additional importance in the context of real-time MPC, where feasibility cannot always be recovered if no initial feasible solution is available due to the real-time constraint. Several methods for the development of controllers that enforce state constraints when they are feasible and allow for relaxation when they are not have been studied in the literature [RM93, ZM95, dOB94, SMR99, KM00, Pri07]. Existing soft constrained schemes for finite horizon MPC, however, do not provide guarantees on closed-loop stability. This issue is addressed by the soft constrained method developed in this thesis, which is based on a finite horizon MPC approach and guarantees closed-loop stability even for unstable systems.

1.1 Outline and Contribution

Part I introduces some background material that is relevant for the main results of this thesis. In Chapter 2 we repeat some basic mathematical definitions and Chapter 3 presents results in convex optimization that are employed as a main tool in this thesis. The considered system formulation as well as the associated stability concepts are introduced in Chapter 4. Chapter 5 discusses the concept of Model Predictive Control in the nominal case and the variants for robust MPC, in particular tube based robust MPC, and MPC for tracking of piecewise constant references.

Part II is concerned with the development of real-time MPC methods that allow for the computation of a control input within the real-time constraint while still providing closed-loop stability. First, a survey of fast MPC methods in the literature using explicit or online MPC is provided in Chapter 6. The main focus is on existing methods for fast online MPC, which in most cases are either based on active set or interior-point methods, and the comparison of the properties of these two optimization methods for a real-time approach.

Chapter 7 develops a real-time MPC scheme combining the two paradigms of explicit and online MPC to overcome their individual limitations. This provides new possibilities in the applicability of MPC to practical problems that often have limits on the storage space or the available computation time. The proposed strategy combines the idea of offline approximation with warm-start techniques from online optimization and thereby offers a tradeoff between the warm-start and online computational effort. A preprocessing method is introduced that provides hard real-time execution, stability and performance guarantees for the proposed controller. The goal is to choose a good tradeoff between the complexity of the PWA approximation and the number of active set iterations required in order to satisfy system constraints in terms of online computation, storage and performance. It is shown how the provided analysis can be utilized to compare different solution methods and identify the best combination of warm-start and online computational effort for a considered application and set of requirements.

By using explicit approximations, the approach presented in Chapter 7 is limited to small or medium problem dimensions. A real-time MPC approach is proposed in Chapter 8 that is based solely on online optimization and can be practically implemented and efficiently solved for large-scale dynamic systems. It is shown how feasibility and input-to-state stability can be guaranteed in the presence of additive disturbances for any given time constraint using robust MPC design and a stability enforcing constraint, while allowing for low computation times with the same complexity as MPC methods without guarantees. The a-priori stability guarantee then allows one to trade the performance of the suboptimal controller for lower online computation times. We show how the proposed scheme can be extended from regulation to tracking of piecewise constant references. All computational details required for a fast implementation based on a barrier interior-point method are provided and results in the robust MPC literature are consolidated into a step-by-step implementation for large-scale systems.

Part III treats the use of soft constraints in MPC. In Chapter 9, a soft constrained MPC approach is developed that provides closed-loop stability even for unstable systems. The presented method is based on a finite horizon MPC setup and uses a terminal weight as well as a terminal constraint. The use of two different types of soft constraints for the relaxation of state constraints along the horizon and the terminal constraint is proposed. We show that, in contrast to existing soft constrained MPC schemes, asymptotic stability of the nominal system in the absence of disturbances is guaranteed. The proposed method significantly enlarges the region of attraction and preserves the optimal behavior with respect to the hard constrained problem whenever all state constraints can be enforced. The robust stability properties are analyzed and input-to-state stability under additive disturbances is proven. In order to allow for a more flexible disturbance handling, the proposed soft constrained MPC scheme can be combined with a robust MPC framework and it is shown that the theoretical results directly extend to the combined case.

Concluding remarks are provided at the end of each chapter in Parts II and III. In Chapter 10 we then provide a brief summary of the results presented in this thesis and an outlook to possible directions for future research on these topics.

1.2 Publications

The work presented in this thesis was done in collaboration with colleagues and is largely based on previous publications, which are listed in the following.

Chapter 7 is entirely based on the following publication:

> *Real-time suboptimal Model Predictive Control using a combination of Explicit MPC and Online Optimization.* M.N. Zeilinger, C.N. Jones and

> M. Morari, to appear in IEEE Transactions on Automatic Control, July 2011. [ZJM11].

A conference version of this article appeared in

> *Real-time suboptimal Model Predictive Control using a combination of Explicit MPC and Online Optimization.* M.N. Zeilinger, C.N. Jones and M. Morari, Proceedings of the 47th IEEE Conference on Decision and Control, Cancun, Mexico, Dec. 2008, 4718-4723. [ZJM08].

Chapter 8 is based on the work in the following paper:

> *On Real-time Robust MPC.* M.N. Zeilinger, C.N. Jones, D.M. Raimondo and M. Morari, submitted to IEEE Transactions on Automatic Control, 2011. [ZJRM11].

The basic ideas of this work were presented in the conference paper

> *Real-time MPC – Stability through Robust MPC design.* M.N. Zeilinger, C.N. Jones, D.M. Raimondo and M. Morari, Proceedings of the 48th IEEE Conference on Decision and Control, Shanghai, China, Dec. 2009 3980-3986. [ZJRM09].

Part II and Chapter 9 is based on the paper

> *Robust stability properties of soft constrained MPC.* M.N. Zeilinger, C.N. Jones and M. Morari, Proceedings of the 49th IEEE Conference on Decision and Control, Atlanta, USA, Dec. 2010. [ZJM10].

Part I

Preliminaries

2 Mathematical Preliminaries

In this chapter we introduce some basic mathematical concepts employed in this thesis. All given definitions are standard and can be found in the literature on analysis and set theory; for a more detailed discussion please refer to one of the standard textbooks [Roc70, RW98, Kur61].

Set Terminology

Definition 2.1 (Convex set). A set $\mathcal{S} \subseteq \mathbb{R}^n$ is *convex* if

$$\alpha s_1 + (1-\alpha)s_2 \in \mathcal{S} \text{ for any } s_1, s_2 \in \mathcal{S} \text{ and } \alpha \in [0,1] \ .$$

Definition 2.2 (Convex hull). The *convex hull* of a set $\mathcal{S} \subseteq \mathbb{R}^n$ is the smallest convex set containing $\mathcal{S}$:

$$\mathrm{conv}(\mathcal{S}) \triangleq \bigcap \left\{ \tilde{\mathcal{S}} \subseteq \mathbb{R}^n \;\middle|\; \mathcal{S} \subseteq \tilde{\mathcal{S}}, \tilde{\mathcal{S}} \text{ is convex} \right\} \ .$$

Definition 2.3 ((Convex) Cone). A set $\mathcal{S} \subseteq \mathbb{R}^n$ is called a *cone* if for every $s \in \mathcal{S}$ and $t \geq 0$: $ts \in \mathcal{S}$. A set $\mathcal{S} \subseteq \mathbb{R}^n$ is called a *convex cone* if it is convex and a cone, i.e. for any $s_1, s_2 \in \mathcal{S}$ and $t_1, t_2 \geq 0$: $t_1 s_1 + t_2 s_2 \in \mathcal{S}$.

Definition 2.4 (Norm cone). Let $\|\cdot\|$ be any norm on $\mathbb{R}^n$. The *norm cone* associated with the norm $\|\cdot\|$ is the set $S = \{(s,r) \mid \|s\| \leq r\} \subseteq \mathbb{R}^{n+1}$, which is a convex cone.

The *second-order cone* is the norm cone associated with the Euclidian norm. It is also known as the *quadratic cone* or the *Lorentz cone*.

Definition 2.5 (Affine set). A set $\mathcal{S} \subseteq \mathbb{R}^n$ is *affine* if

$$\alpha s_1 + (1-\alpha)s_2 \in \mathcal{S} \text{ for any } s_1, s_2 \in \mathcal{S} \text{ and } \alpha \in \mathbb{R} \ .$$

Definition 2.6 (ϵ-ball). The open *ϵ-ball* in $\mathbb{R}^n$ around a given point $x_c \in \mathbb{R}^n$ is the set $\mathcal{B}_\epsilon(x_c) \triangleq \{x \in \mathbb{R}^n \mid \|x - x_c\| < \epsilon\}$, where the radius $\epsilon > 0$ and $\|\bullet\|$ denotes any vector norm, usually the Euclidian norm $\|\bullet\|_2$.

Definition 2.7 (Interior). The *interior* of a set $\mathcal{S} \subseteq \mathbb{R}^n$ is given as

$$\operatorname{int} \mathcal{S} \triangleq \{s \in \mathcal{S} \mid \exists\ \epsilon > 0, \mathcal{B}_\epsilon(s) \subseteq \mathcal{S}\} \ .$$

Definition 2.8 (Relative Interior). The *relative interior* of a set $\mathcal{S} \subseteq \mathbb{R}^n$ is defined as

$$\operatorname{relint} \mathcal{S} \triangleq \{s \in \mathcal{S} \mid \exists\ \epsilon > 0, \mathcal{B}_\epsilon(s) \cap \operatorname{aff} \mathcal{S} \subseteq \mathcal{S}\} \ ,$$

where

$$\operatorname{aff} \mathcal{S} \triangleq \bigcap \left\{ \tilde{\mathcal{S}} \subseteq \mathbb{R}^n \ \middle|\ \mathcal{S} \subseteq \tilde{\mathcal{S}}, \tilde{\mathcal{S}} \text{ is affine} \right\}$$

denotes the affine hull of a set $\mathcal{S}$.

Definition 2.9 (Open/Closed set). A set $\mathcal{S} \subseteq \mathbb{R}^n$ is called *open* if $\mathcal{S} = \operatorname{relint} \mathcal{S}$. A set is called closed if its complement $\mathcal{S}^c \triangleq \{s \mid s \notin \mathcal{S}\}$ is open.

Definition 2.10 (Bounded set). A set $\mathcal{S} \subseteq \mathbb{R}^n$ is *bounded* if it is contained inside some ball $\mathcal{B}_r(\bullet)$ of finite radius r, i.e. $\exists r < \infty, s \in \mathbb{R}^n$ such that $\mathcal{S} \subseteq \mathcal{B}_r(s)$.

Definition 2.11 (Compact set). A set $\mathcal{S} \subseteq \mathbb{R}^n$ is *compact* if it is closed and bounded.

Definition 2.12. A set $\mathcal{S} \subseteq \mathbb{R}^n$ is *full-dimensional* if $\exists \epsilon > 0, s \in \operatorname{int} \mathcal{S}$ such that $\mathcal{B}_\epsilon(s) \subseteq \mathcal{S}$.

Polyhedra

A brief summary of the basic definitions related to polyhedra used in this thesis is provided in the following, for more information see e.g. [Zie94, Grü03].

Definition 2.13 (Halfspace). A closed *halfspace* in $\mathbb{R}^n$ is a set of the form

$$\mathcal{P} = \{x \in \mathbb{R}^n \mid a^T x \leq b\}, \quad a \in \mathbb{R}^n, b \in \mathbb{R} \ .$$

Definition 2.14 (Polyhedron). A *polyhedron* is the intersection of a finite number of halfspaces.

Definition 2.15 (Polytope). A *polytope* is a bounded polyhedron.

Definition 2.16 (Polyhedral partition). The collection of sets $\mathcal{P}^N \triangleq \{P_j\}_{j=1}^N$ with $N \in \mathbb{N}$ is called a *polyhedral partition* of a set $\mathcal{P} \subseteq \mathbb{R}^n$ if all P_j are full-dimensional polyhedra, $\cup_{j=1}^N P_j = \mathcal{P}$ and $\operatorname{int} P_i \cap \operatorname{int} P_j = \emptyset\ \forall i \neq j$ and $i, j \in \{1, \ldots, N\}$.

Function Terminology

Definition 2.17 (Epigraph). The *epigraph* of a function $f : \mathcal{D} \to \mathbb{R}$ with the domain $\mathcal{D} \subseteq \mathbb{R}^n$ is given by the set

$$\operatorname{epi} f \triangleq \{(x,t) \mid (x,t) \in \mathcal{D} \times \mathbb{R}, f(x) \leq t\} \ .$$

Definition 2.18 ((Uniformly) Continuous function). A function $f : \mathcal{D} \to \mathbb{R}^{n_f}$ with the domain $\mathcal{D} \subseteq \mathbb{R}^n$ is called *continuous* at a point $\hat{x} \in \mathcal{D}$ if

$$\forall \epsilon \ \exists \delta : \|x - \hat{x}\| < \delta \Rightarrow \|f(x) - f(\hat{x})\| < \epsilon \ . \tag{2.1}$$

A function is called continuous if it is continuous at all $x \in \mathcal{D}$.
A function is called *uniformly continuous* at $\hat{x} \in \mathcal{D}$ if (2.1) holds with $\delta = \delta(\epsilon)$.

Definition 2.19 (Lipschitz continuity). A function $f : \mathcal{D} \to \mathbb{R}^{n_f}$ with the domain $\mathcal{D} \subseteq \mathbb{R}^n$ is called *Lipschitz continuous* if

$$\|f(x) - f(\hat{x})\| \leq L\|x - \hat{x}\| \ \forall \hat{x}, x \in \mathcal{D} \ ,$$

for some $L \in \mathbb{R}$ called a Lipschitz constant.

Definition 2.20 ((Polyhedral) PWA function). A function $f : \mathcal{P} \to \mathbb{R}^{n_f}$ with $\mathcal{P} \subseteq \mathbb{R}^n$ is *piecewise affine (PWA)* if $\mathcal{P}^N$ is a polyhedral partition of $\mathcal{P}$ and

$$f(x) \triangleq C^j x + D^j \text{ if } x \in P_j \ \forall \, P_j \in \mathcal{P}^N \ ,$$

where $C^j \in \mathbb{R}^{n_f \times n}$, $D^j \in \mathbb{R}^{n_f}$ and $j \in \{1, \ldots, N\}$.

Definition 2.21 (Convex/Concave function). A function $f : \mathcal{D} \to \mathbb{R}$ is *convex*, if its domain $\mathcal{D} \subseteq \mathbb{R}^n$ is a convex set and

$$f(\alpha x_1 + (1-\alpha)x_2) \leq \alpha f(x_1) + (1-\alpha) f(x_2) \tag{2.2}$$

for any $x_1, x_2 \in \mathcal{D}$ and $\alpha \in [0,1]$. $f(\bullet)$ is concave if $-f(\bullet)$ is convex.
A function f is *strictly convex* if strict inequality holds in (2.2) whenever $x_1 \neq x_2$ and $0 < \alpha < 1$.

Definition 2.22 (Support function). The *support function* $h_{\mathcal{S}} : \mathbb{R}^n \to \mathbb{R}$ of a convex set $\mathcal{S}$ evaluated at $x \in \mathbb{R}^n$ is defined as $h_{\mathcal{S}}(x) \triangleq \sup_{y \in \mathcal{S}} x^T y$.

3 Convex Optimization

Convex optimization is a subclass of mathematical optimization with favorable theoretical and practical properties, which will be used as an important tool for control throughout this thesis. One of the most important theoretical properties is the fact that if a local minimum exists, it is also a global minimum. In addition to a well-established theoretical foundation the main practical advantage of convex optimization is the existence of reliable and efficient solvers for finding the global optimal solution. It has been discovered that a large number of problems arising in applications in a wide range of fields like automatic control, estimation or communications, can be cast as convex problems making convex optimization a widely used tool in practice.

In this chapter we present an overview of a selection of topics in convex optimization that are relevant for this thesis. We first repeat some basic theoretical results and then provide an overview of two popular optimization methods, active set and interior-point methods, as well as some important properties of exact penalty functions. Finally, the problem of parametric programming, in particular for linear and quadratic problems, is briefly introduced. More information on these topics can be found in one of the numerous textbooks on optimization, for example [BGK$^+$82, GMW82, Fle87, Nes03, BTN01, BV04, NW06].

We consider a *convex optimization* problem of the following general form:

$$\begin{aligned} \text{minimize} \quad & f_0(z) && (3.1a)\\ \text{subject to} \quad & f_{\mathcal{I}}(z) \leq 0 \ , && (3.1b)\\ & f_{\mathcal{E}}(z) = 0 \ , && (3.1c) \end{aligned}$$

where $z \in \mathbb{R}^{n_z}$ is the optimization variable, the vector $f_{\mathcal{I}}$ is formed from the scalar functions $f_i(z) : \mathbb{R}^{n_z} \to \mathbb{R}, i \in \mathcal{I}$ and $f_{\mathcal{E}}$ from $i \in \mathcal{E}$, where the index sets $\mathcal{E} \subset \{1, \cdots, m\}$ and $\mathcal{I} = \{1, \cdots, m\} \setminus \mathcal{E}$ define the set of equality and inequality constraints. The functions $f_0, f_i \ \forall i \in \mathcal{I}$ are convex and the functions $f_i \ \forall i \in \mathcal{E}$ are affine.

Optimality Conditions

The primal problem (3.1) has an associated Lagrange *dual problem*, which serves as a certificate of optimality for a primal feasible solution:

$$\begin{aligned} &\text{maximize} \quad g(\lambda, \nu) && (3.2a)\\ &\text{subject to} \quad \lambda \geq 0 \ , && (3.2b) \end{aligned}$$

where $g(\lambda, \nu) = \min_z \mathcal{L}(z, \lambda, \nu)$, $\lambda \in \mathbb{R}^{|\mathcal{I}|}$ and $\nu \in \mathbb{R}^{|\mathcal{E}|}$ are the *Lagrange multipliers* associated with the inequality and equality constraints, respectively, and $\mathcal{L}$ is the *Lagrange function* given by

$$\mathcal{L}(z, \lambda, \nu) = f_0(z) + \lambda^T f_{\mathcal{I}}(z) + \nu^T f_{\mathcal{E}}(z) \ . \tag{3.3}$$

The difference between a primal feasible and a dual feasible solution is called the *duality gap.* Under certain constraint qualification conditions *strong duality* holds, i.e. the optimal value of the primal and dual problem will be equal. One commonly used constraint qualification is Slater's condition.

Definition 3.1 (Slater's condition). If the primal problem in (3.1) is strictly feasible, i.e. there exists a z with $f_{\mathcal{E}}(z) = 0$ and $f_{\mathcal{I}}(z) < 0$, then strong duality holds.

The *Karush-Kuhn-Tucker (KKT) optimality conditions* provide necessary and sufficient conditions for a solution to be globally optimal.

Theorem 3.2 (KKT optimality conditions) *Let z be a feasible point for a convex optimization problem of the form* (3.1) *satisfying some regularity conditions, e.g. Slater's condition, and let $f_i(z), i \in \mathcal{I} \cup \mathcal{E}$ be continuously differentiable. If z^* is a global optimum then there exists vectors λ^* and ν^* such that the following conditions are satisfied:*

$$\begin{aligned} \nabla_z \mathcal{L}(z^*, \lambda^*, \nu^*) &= 0 \ , && (3.4a)\\ f_{\mathcal{I}}(z^*) &\leq 0 \ , && (3.4b)\\ f_{\mathcal{E}}(z^*) &= 0 \ , && (3.4c)\\ \lambda^* &\geq 0 \ , && (3.4d)\\ \lambda_i^* f_i(z^*) &= 0 \quad \forall i \in \mathcal{I} \ . && (3.4e) \end{aligned}$$

The KKT conditions comprise primal and dual feasibility conditions (3.4b)-(3.4d), minimization of $\mathcal{L}(\bullet, \bullet, \bullet)$ (3.4a) and the complementarity conditions (3.4e). See e.g. [Nes03, BV04, NW06] for further details and proofs on convex optimization.

In the following we describe three widely used special subclasses of the general convex optimization problem in (3.1), listed in the order of increasing computational complexity: linear programs, quadratic programs and second-order cone programs.

Linear Program

If the cost and the constraints in (3.1) are all affine, the optimization problem is called a *linear program (LP)*. We consider the following form of an LP:

$$\begin{aligned} \text{minimize} \quad & c^T z && (3.5a)\\ \text{subject to} \quad & G_{\mathcal{I}} z \leq d_{\mathcal{I}} \ , && (3.5b)\\ & G_{\mathcal{E}} z = d_{\mathcal{E}} \ , && (3.5c) \end{aligned}$$

where $G \in \mathbb{R}^{m \times n_z}$ and $d \in \mathbb{R}^m$. The feasible set defined by the constraints in (3.5) is a polyhedron over which a linear function is minimized.

Linear programs can be efficiently solved using e.g. the simplex method [Dan49], an optimization method specifically designed for linear programming that can be considered as a specialized active set method, or interior-point methods [NN94, NW06]. These two approaches are fundamentally different and will be explained in more detail in Section 3.1. LPs appear in a vast number of different applications and LP solvers have reached a high level of sophistication. Commercially available software tools are, for example, CPLEX [CPL] or NAG [NAG], but also the freely available C++ implementations [CDD] and [CLP] provide efficient solvers.

Quadratic Program

The optimization problem in (3.1) is called a convex *quadratic program (QP)* if the constraints are affine and the cost function is (convex) quadratic:

$$\begin{aligned} \text{minimize} \quad & z^T H z + c^T z && (3.6a)\\ \text{subject to} \quad & G_{\mathcal{I}} z \leq d_{\mathcal{I}} \ , && (3.6b)\\ & G_{\mathcal{E}} z = d_{\mathcal{E}} \ , && (3.6c) \end{aligned}$$

where $H \in \mathbb{R}^{n_z \times n_z}$ with $H \succeq 0$ defines a convex quadratic objective function, which is minimized over a polyhedron. If $H \succ 0$, problem (3.6) is called a *strictly convex QP*. The difficulty of solving a convex QP is generally considered similar to an LP and most quadratic programs can be solved efficiently by means of, for example, active-set, gradient or interior-point methods [NW06]. Quadratic programs form an important problem class for many applications and also arise as subproblems in general constrained optimization methods, such as sequential programming or interior-point methods. Commercial solvers implementing quadratic programming algorithms are CPLEX [CPL] or NAG [NAG], a freely available solver implementation in C is provided by the QPC package [QPC].

Second-order Cone Program

Second-order Cone Programs (SOCP) are nonlinear convex problems that include linear and (convex) quadratic programs as special cases, but are less general than semi-definite programs (SDP). A good reference for SOCPs is [NN94], see e.g. [LVBL98] for a study on problems that can be cast as an SOCP.

We consider the following general form of an SOCP:

$$\begin{aligned} \text{minimize} \quad & c^T z && (3.7a)\\ \text{subject to} \quad & G_{\mathcal{E}} z = d_{\mathcal{E}} \ , && (3.7b)\\ & \|G^i z + f^i\|_2 \le {d^i}^T z + e^i, \ i \in \mathcal{I} \ , && (3.7c) \end{aligned}$$

where $G^i \in \mathbb{R}^{p_i \times n_z}$, $f^i \in \mathbb{R}^{p_i}$, $d^i \in \mathbb{R}^{n_z}$ and $e^i \in \mathbb{R}$. Constraints (3.7c) represent second-order cone constraints of dimension p_i. If $G^i = 0$ for all $i \in \mathcal{I}$ the SOCP reduces to a general LP. If $d^i = 0$ for all $i \in \mathcal{I}$, the SOCP reduces to a *quadratically constrained quadratic program (QCQP)*, i.e. a QP with quadratic inequality constraints, which is obtained by squaring the constraints in (3.7c). SOCPs are, however, more general than QCQPs. Several efficient primal-dual interior-point methods for SOCPs were developed in recent years, which can solve an SOCP more efficiently than solvers treating it as an SDP. The difference is significant if the dimensions of the second-order cone constraints are large. A powerful commercial software for SOCPs is MOSEK, which implements the algorithm in [ART03], but they can also be solved using SeDuMi [SED] or SDPT3 [TTT].

3.1 Optimization Methods

There are many different optimization methods for the solution of convex problems in the optimization literature. In this section we describe two of the most prominent approaches, active set and interior-point methods, that are widely used in practice and will be applied in this thesis. At the end of the section, the notion of exact penalty functions is briefly introduced, which is relevant for the methods presented in this work.

3.1.1 Active Set Methods

Active set methods aim at identifying the set of active constraints at the optimum (the optimal active set), which defines the optimal solution of an optimization problem, by guessing an active set and then iteratively improving this guess.

Definition 3.3 (Active Set). At a feasible point z, the inequality constraint $i \in \mathcal{I}$ is said to be *active* if $f_i(z) = 0$. The *active set* $\mathcal{A}(z)$ consists of the equality constraint indices $\mathcal{E}$ together with the indices of active inequality constraints:

$$\mathcal{A}(z) = \mathcal{E} \cup \{i \in \mathcal{I} \mid f_i(z) = 0\} \ . \tag{3.8}$$

There are three variants of active set methods: primal, dual and primal-dual methods. An early work on active set methods is presented in [Fle71], although the introduction of the simplex method for linear programming [Dan49, Dan63] can also be interpreted as a specialized active set method and the first QP algorithms were extensions of the simplex method [Wol59]. A detailed description of active set methods can be found in almost every standard textbook on optimization, for example [GMW82, Fle87, NW06]. The discussion in this section focuses on primal methods which generate primal feasible iterates while decreasing the objective function. The main steps of a primal active set method are outlined in the following for the QP (3.6).

Active set methods solve a quadratic subproblem at each iteration, where all equality constraints and some of the inequality constraints are enforced as equalities. This set, which is a subset of the active set, is called the *working set* $W \subseteq \mathcal{A}(z)$. Although cycling is often not an issue in practice, we assume that the active set is non-degenerate, i.e. the active constraints are linearly independent. At a given iterate z with working set W, the step direction Δz is computed by solving a constrained QP, where all constraints with index in W are considered as equality constraints and all other inequality constraints are discarded:

$$\begin{aligned} \text{minimize} \quad & \Delta z^T H \Delta z + c^T \Delta z && (3.9) \\ \text{subject to} \quad & G_W \Delta z = 0 \ . && (3.10) \end{aligned}$$

The solution to (3.9) can be computed by any standard method for equality constrained QPs, see e.g. [NW06] for an overview. A commonly applied approach is the direct or the iterative solution of the corresponding KKT system

$$\begin{bmatrix} 2H & G_W^T \\ G_W & 0 \end{bmatrix} \begin{bmatrix} \Delta z \\ \lambda \end{bmatrix} = \begin{bmatrix} -c \\ 0 \end{bmatrix} \ . \tag{3.11}$$

If the search direction Δz is non-zero, the step length is obtained in a line search procedure determining the maximum step size for which all constraints are satisfied:

$$\tau = \min_{i \in \mathcal{I} \setminus W} \left\{ \frac{d_i - G_i z}{G_i \Delta z} \,\middle|\, G_i \Delta z > 0 \right\} \ . \tag{3.12}$$

If $\tau < 1$ the step was blocked by some inequality constraint not in W and one of the blocking constraints is added to W. The iterate is then updated to

$$z^+ = z + \tau \Delta z \ . \tag{3.13}$$

The iterations are continued in this manner until $\Delta z = 0$, i.e. the minimum of the objective function for the current working set is found. If all the Lagrange multipliers λ corresponding to the inequality constraints are nonnegative, the optimal solution to the original problem (3.6) is found. If one or more of the multipliers is negative, one of these constraints is removed from the working set W allowing for a decrease of the objective function. A new search direction is computed and the iterations are continued until the optimal active set is found. There are many variants for adding or deleting constraints in the active set, a discussion of the different options and more details on active set methods can be found in the literature, e.g. in [GMW82, Fle87, NW06].

3.1.2 Interior-Point Methods

Classical interior-point methods start the optimization from an interior point, i.e. a point that strictly satisfies the inequality constraints in (3.14e), and then follow a so-called central path to the optimal solution by generating feasible iterates that satisfy the inequality constraints strictly, which is the origin of the name. Interior-point algorithms have been shown to work well in practice and efficiently solve convex optimization problems, in particular the special cases of convex programs discussed previously, such as LPs, QPs and SOCPs. The area was started by Karmarkar's work in 1984 [Kar84] with the proposition of the first interior-point method for LPs and today it represents a theoretically rich and practical field of convex optimization.

While there is a variety of different interior-point methods, we will restrict the brief outline in the following to path following methods, in particular *primal-dual* and *barrier* interior-point methods. While the focus is on primal feasible methods, we note that infeasible variants for both primal-dual as well as barrier interior-point methods are available, details can be found e.g. in [Wri97b, BV04, NW06]. While barrier methods are among the earliest developed methods, primal-dual approaches belong to the more modern continuation based implementations of interior-point methods. We will discuss both variants briefly for the general convex optimization problem in (3.1) in the following.

Primal-dual Interior-point Methods

Primal-dual methods compute primal-dual solutions of the convex problem in (3.1) by applying Newton's method to the KKT conditions in (3.4). As in most iterative algorithms there are two main ingredients: the choice of the step direction and the step size. The Newton procedure is modified to bias the search direction into the interior and keep (λ, z) from moving too close to the boundary of their constraints.

Consider the perturbed KKT conditions given by

$$\begin{aligned}
\nabla_z f_0(z) + \Delta G_{\mathcal{I}}(z)^T \lambda + \Delta G_{\mathcal{E}}(z)^T \nu = 0 \ , && (3.14a)\\
f_{\mathcal{I}}(z) + s = 0 \ , && (3.14b)\\
G_{\mathcal{E}}(z) = 0 \ , && (3.14c)\\
S\lambda - \kappa \mathbf{1} = 0 \ , && (3.14d)\\
\lambda \geq 0,\ s \geq 0 \ , && (3.14e)
\end{aligned}$$

where $S = \text{diag}(s)$ and $\Delta G(z)$ is the Jacobian matrix of the functions $f_i(z), i \in \mathcal{I} \cup \mathcal{E}$. The solution to (3.14) $(z(\kappa), s(\kappa), \lambda(\kappa), \nu(\kappa))$ is parameterized by the scalar $\kappa > 0$ and the trajectory formed by these points is called the primal-dual central path $\mathcal{C}$. The perturbed KKT conditions are solved for a sequence of parameters κ that converges to zero, while maintaining $\lambda, s \geq 0$. If $\mathcal{C}$ converges as $\kappa \to 0$, the algorithm converges to the optimal primal-dual solution of the convex program.

Most primal-dual solutions take Newton steps towards the central path. Let (z, s, λ, ν) be a strictly feasible point at iteration k. The primal-dual Newton step $(\Delta z, \Delta s, \Delta \lambda, \Delta \nu)$ is obtained from

$$\begin{bmatrix} \nabla^2_{zz}\mathcal{L}(z,\lambda,\nu) & 0 & \Delta G^T_{\mathcal{E}}(z) & \Delta G^T_{\mathcal{I}}(z) \\ 0 & \Sigma & 0 & I \\ \Delta G_{\mathcal{E}}(z) & 0 & 0 & 0 \\ \Delta G_{\mathcal{I}}(z) & I & 0 & 0 \end{bmatrix} \begin{bmatrix} \Delta z \\ \Delta s \\ \Delta \nu \\ \Delta \lambda \end{bmatrix} = - \begin{bmatrix} r_d \\ S\lambda - \kappa \mathbf{1} \\ 0 \\ 0 \end{bmatrix} , \qquad (3.15)$$

where $\Sigma = S^{-1}\Lambda$, $\Lambda = \text{diag}(\lambda)$ and r_d, the so-called dual residual, is given by

$$r_d = \nabla_z f_0(z) + \Delta G_{\mathcal{I}}(z)^T \lambda + \Delta G^T_{\mathcal{E}}(z)\nu \ . \qquad (3.16)$$

A line-search along this Newton direction is performed, such that the new iterate

$$(z^+, s^+, \lambda^+, \nu^+) = (z, s, \lambda, \nu) + \alpha(\Delta z, \Delta s, \Delta \lambda, \Delta \nu) \ , \qquad (3.17)$$

where $\alpha \in (0, 1]$ is the line search parameter, does not violate the inequality constraints and is not too close to the boundary. The particular algorithm parameters such as step size and decrease rate of κ depends on the method of choice. There exist numerous variants of primal-dual interior-point algorithms, such as path-following, potential reduction or trust region interior-point methods, which cannot all be discussed here and we refer the interested reader to the overviews in [NN94, Wri97b, NW06]. An interesting variant are so-called predictor-corrector methods, which use the factorization of the KKT system to compute two directions, a predictor and a corrector step. In particular Mehrotra's algorithm [Meh92] has proven very efficient and is implemented in many available solvers.

Barrier Interior-point Methods

In a barrier method, the inequality constraints of the convex program in (3.1) are replaced by a barrier penalty in the cost function resulting in the approximate problem formulation

$$\begin{aligned} \text{minimize} \quad & f_0(z) + \kappa\psi(\mathbf{s}) && (3.18a)\\ \text{subject to} \quad & f_{\mathcal{E}}(z) = 0 \ , && (3.18b)\\ & f_{\mathcal{I}}(z) + s = 0 \ , && (3.18c) \end{aligned}$$

where $\kappa > 0$ is the barrier parameter and $\psi(\bullet)$ is the log barrier function given by

$$\psi(z) = \sum_{i=1}^{|\mathcal{I}|} - \log(s_i) \ . \tag{3.19}$$

Note that the KKT conditions of (3.18) coincide with (3.14) after a small transformation. The barrier method starts from a strictly primal feasible (interior) point and then solves a sequence of linearly constrained minimization problems (3.18) for decreasing values of the barrier parameter κ starting from the previous iterate using Newton's method. At each iteration, the primal search direction Δz from the current iterate z is obtained by solving the following linear system:

$$\begin{bmatrix} \nabla^2_{zz}\mathcal{L}(z,\nu) + \kappa\Delta G_{\mathcal{I}}(z)^T S^{-2}\Delta G_{\mathcal{I}}(z) & \Delta G_{\mathcal{E}}^T(z) \\ \Delta G_{\mathcal{E}}(z) & 0 \end{bmatrix} \begin{bmatrix} \Delta z \\ \Delta\nu \end{bmatrix} = - \begin{bmatrix} r_d \\ 0 \end{bmatrix} \ , \tag{3.20}$$

where the dual residual r_d is again defined by (3.16). The Newton step used in the barrier method can be obtained directly from (3.15) by eliminating Δs and $\Delta\lambda$ from the system of equations and using the relationship $\Lambda = \kappa S^{-1}$.

A line-search along the primal direction is performed such that the new variables

$$z^+ = z + \alpha\Delta z \tag{3.21}$$

satisfy the primal inequality constraints. As $\kappa \to 0$ the solution converges in the limit to the optimal solution of (3.1). See for example [BV04, NW06] for a detailed description of the method and the choice of parameters involved in the procedure.

3.1.3 Exact Penalty Functions

Penalty or barrier function methods are often applied for the solution of nonlinear constrained optimization problems in order to reduce the original problem to a sequence of unconstrained subproblems. The barrier interior-point method described in Section 3.1.2 is one variant of a barrier function method based on the logarithmic barrier function, one of the most popular barrier functions for the solution of inequality constrained problems. If the penalty function is chosen such that it is minimized locally by the optimal solution to the original problem, it is called an *exact penalty function.* Exact penalty functions are often employed in control theory in order to allow for a relaxation of certain constraints while still guaranteeing that optimality and constraint satisfaction is maintained if possible.

This section states the main condition for a penalty function that is defined in terms of norms to be exact. For simplicity the results are given for a convex optimization problem with inequality constraints only, i.e. $\mathcal{E} = \emptyset$, but the results can be directly extended to the combination with equality constraints. The presentation follows closely that in [Fle87] but the results can similarly be found in [Lue84, GMW82, NW06].

Using a penalty function based on norms, the convex optimization problem in (3.1) with $\mathcal{E} = \emptyset$ can be written as the following unconstrained minimization problem:

$$\text{minimize} \quad f_0(z) + \rho \|[f_{\mathcal{I}}(z)]_+\| \ , \tag{3.22}$$

where ρ is a penalty parameter weighting the contribution of the penalty function to the cost. The definition of an exact penalty function uses the concept of a dual norm, defined as

$$\|u\|_D = \max_{\|v\| \leq 1} u^T v \ . \tag{3.23}$$

It has been shown that the dual of $\|\cdot\|_1$ is $\|\cdot\|_\infty$ and that $\|\cdot\|_2$ is self-dual.

The following result provides a condition on the parameter ρ ensuring the equivalence of the solutions to (3.1) (with $\mathcal{E} = \emptyset$) and (3.22).

Theorem 3.4 (Exact penalty function, Theorem 14.3.1 in [Fle87]) *Let z^* denote the optimizer of* (3.1) *(for $\mathcal{E} = \emptyset$) and λ^* the optimal Lagrange multipliers corresponding to the inequality constraints satisfying the KKT conditions in* (3.4)*. If the penalty weight $\rho > \|\lambda^*\|_D$ and $f_{\mathcal{I}}(z^*) \leq 0$, then z^* is also the optimizer of* (3.22) *and* (3.22) *is called an exact penalty function.*

The main disadvantage of the exact penalty function is that it is non-smooth and can therefore not be solved using the effective techniques for smooth optimization. Note that it is, however, exactly the non-smoothness which causes the penalty function to be exact, a quadratic penalty or a log barrier penalty, for example, does not provide an exact penalty function. The non-smoothness of the optimization problem can be avoided by reformulating problem (3.22) using so-called slack variables ϵ capturing the constraint violations:

$$\begin{aligned} &\text{minimize} && f_0(z) + \rho\|\epsilon\| && \text{(3.24a)}\\ &\text{subject to} && f_{\mathcal{I}}(z) \leq \epsilon\ , && \text{(3.24b)}\\ & && 0 \leq \epsilon\ . && \text{(3.24c)} \end{aligned}$$

Problem (3.24) is a smooth constrained optimization problem, which can be solved using standard methods for smooth convex optimization.

Common exact penalty functions are l_1- or l_∞-norm penalties allowing for a reformulation of (3.24) as an LP or QP if the cost is linear or quadratic, respectively, and the inequality constraints are affine.

3.2 Parametric Programming

Many practical problems require the solution of an optimization problem repeatedly for variations in the problem data, which can be formulated as a parameter dependence of the cost function and/or the constraints. The parameter $\theta \in \mathbb{R}^{n_\theta}$ is then given as an input to the optimization problem. This section provides a short introduction to parametric programming and states the main results related to parametric linear and quadratic programming. For more information on the general concept and properties of parametric programming see [BGK$^+$82]. Note that in the literature the term multi-parametric programming is sometimes used to emphasize the fact that the parameter is not a scalar but a vector.

In the case of convex optimization we obtain the following general form of a *parametric convex optimization problem*

$$\begin{aligned} &\text{minimize} && f_0(z,\theta) && \text{(3.25a)}\\ &\text{subject to} && f_{\mathcal{I}}(z,\theta) \leq 0\ , && \text{(3.25b)}\\ & && f_{\mathcal{E}}(z,\theta) = 0\ . && \text{(3.25c)} \end{aligned}$$

The goal is to solve this parametric optimization problem for all values of the parameter θ by determining an explicit representation of the cost function and the optimizer $z^*(\theta)$ as a function of the parameter θ. The computation of the parametric solution is, however, only possible in special cases, for example the case of parametric linear or quadratic programs, which are, however, of particular importance in the context of Model Predictive Control (MPC) and will be discussed in the following. The parametric solution in the context of MPC problems will be addressed again in Chapter 6. Following the presentation in the literature, the results are stated for an optimization problem with inequality constraints only, i.e. $\mathcal{E} = \emptyset$. Note that this formulation can be directly obtained from (3.5) or (3.6), respectively, by eliminating the equality constraints.

Parametric Linear Program

We denote a linear program, where the parameter θ enters linearly in the constraints, as a *parametric linear program (pLP)*:

$$J^*(\theta) = \min \quad c^T z \tag{3.26a}$$

$$\text{s.t.} \quad Gz \leq d + E\theta \ . \tag{3.26b}$$

For an overview of parametric linear programming the reader is referred to [Gal95]. The main properties of the parametric solution to the pLP are stated in the following theorem.

Theorem 3.5 (Solution to the pLP, [Gal95]) *Consider the parametric linear program* (3.26)*. The set of feasible parameters Θ is a closed polyhedron in $\mathbb{R}^{n_\theta}$. The optimal value function $J^* : \Theta \to \mathbb{R}$ is continuous, convex and a piecewise affine function of θ. There exists an optimizer function $z^* : \Theta \to \mathbb{R}^{n_z}$ that is a continuous and piecewise affine function of θ.*

Main difficulties for parametric linear programming algorithms are primal degeneracy, i.e. when the basis describing the optimal solution is not unique, and dual degeneracy, i.e. when the optimizer is not unique. The issue of resolving dual degeneracy is addressed in [BBM03] or [JKM07], which is based on lexicographic perturbation.

Parametric Quadratic Program

Similarly, a *parametric quadratic program (pQP)* is a quadratic program, where the parameter enters linearly in the cost and the constraints:

$$J^*(\theta) = \theta^T Y \theta + \min \quad z^T H z + \theta^T F z \tag{3.27a}$$

$$\text{s.t.} \quad Gz \leq d + E\theta \ . \tag{3.27b}$$

The main properties of the parametric solution to the pQP are given in the following, a more detailed discussion of pQPs can be found in [BGK$^+$82].

Theorem 3.6 (Solution to the pQP) *Consider the parametric quadratic program* (3.27) *with $H \succ 0$, $\left[\begin{smallmatrix} H & F \\ F^T & Y \end{smallmatrix}\right] \succeq 0$. The set of feasible parameters Θ is a closed polyhedron in $\mathbb{R}^{n_\theta}$. The optimal value function $J^* : \Theta \to \mathbb{R}$ is continuous, convex and a piecewise quadratic function of θ. The optimizer $z^* : \Theta \to \mathbb{R}^{n_z}$ is unique, continuous and a piecewise affine function of θ.*

4 System and Control Theory

This chapter introduces a common system formulation as well as the associated stability concepts, which are well-established in control theory and are used throughout this work.

4.1 System Formulation

We consider the control of a discrete-time uncertain linear system described by

$$x(k+1) = Ax(k) + Bu(k) + w(k),\ k \in \mathbb{N} \tag{4.1}$$

that is subject to the following constraints:

$$x(k) \in \mathbb{X} \subset\ \mathbb{R}^n,\ u(k) \in \mathbb{U} \subset \mathbb{R}^m\ \forall\ k \in \mathbb{N}\ \ , \tag{4.2}$$

where $x(k)$ is the state, $u(k)$ is the control input and $w(k) \in \mathcal{W} \subset \mathbb{R}^n$ is a bounded disturbance at the k'th sample time. $\mathbb{X}$ and $\mathbb{U}$ are polytopic constraints on the states and inputs that are assumed to each contain the origin in its interior. $\mathcal{W}$ is a convex and compact disturbance set that contains the origin (not necessarily in the interior).

The solution of system (4.1) at sampling time k for the initial state $x(0)$, sequence of control inputs $\mathbf{u}$ and disturbances $\mathbf{w}$ is denoted as $\phi(k, x(0), \mathbf{u}, \mathbf{w})$. If system (4.1) is controlled by the control law $u = \kappa(x)$ then the closed-loop system is given by

$$x(k+1) = Ax(k) + B\kappa(x(k)) + w(k)\ \ . \tag{4.3}$$

The solution of the closed-loop uncertain system at time k, for the initial state $x(0)$ and a sequence of disturbances $\mathbf{w}$ is denoted as $\phi_\kappa(k, x(0), \mathbf{w})$.

The nominal model of system (4.1) describes the system considering no disturbance and is given by

$$\bar{x}(k+1) = A\bar{x}(k) + B\bar{u}(k)\ \ . \tag{4.4}$$

The solution to this equation for an initial state $x(0)$ is denoted as $\bar{\phi}(k, \bar{x}(0), \mathbf{u})$. If system (4.4) is controlled by the control law $u = \kappa(x)$ then the closed-loop system is

$$\bar{x}(k+1) = A\bar{x}(k) + B\kappa(\bar{x}(k))\ \ . \tag{4.5}$$

When we focus on one time instance we sometimes drop the dependence on k and use the lighter notation $x^+ = Ax + Bu + w$ in the disturbed case or $\bar{x}^+ = A\bar{x} + B\bar{u}$ in the nominal case, where x^+ denotes the successor state at the next sampling time.

While the system under consideration may be unstable, it is assumed to satisfy the following standing assumption:

Assumption 4.1. The pair (A,B) is stabilizable.

We further assume that the state $x(k)$ of the system can be measured at each sampling time k.

Parametrization of a Steady-state:

A steady-state (x_s, u_s) of the nominal system (4.4) is characterized by the condition

$$(I - A)x_s = Bu_s \ . \tag{4.6}$$

Since matrix A may have eigenvalues on the unit circle, the steady-state cannot always be parameterized solely by the steady-state input u_s. We therefore use the following parametrization of a steady-state (x_s, u_s) of system (4.4) by the parameter $\theta \in \mathbb{R}^{n_\theta}$:

$$\begin{bmatrix} x_s \\ u_s \end{bmatrix} = M\theta = \begin{bmatrix} M_x \\ M_u \end{bmatrix} \theta \ , \tag{4.7}$$

where the columns of M form an orthonormal basis for the nullspace of the matrix $[I - A \quad -B]$, n_θ is the dimension of the nullspace and M_x, M_u are appropriate partitions of M.

4.2 Stability and Invariance

Providing a closed loop stability guarantee is one of the main goals in controller design. The notion of stability employed in this thesis is based on Lyapunov stability theory. We first state the well-known results for analyzing stability of the controlled system (4.5) with nominal dynamics by means of classic Lyapunov stability theory and then present the framework of input-to-state stability allowing for the analysis of the robust stability properties of the controlled uncertain system (4.3).

In the context of constrained systems, the notion of invariance plays an important role for the stability analysis. The following standard definitions can be found e.g. in [Bla99]:

Definition 4.2 (Positively invariant (PI) set). A set $\mathcal{X} \subseteq \mathbb{R}^{n_x}$ is a *positively invariant (PI)* set of system (4.5), if

$$A\bar{x}(k) + B\kappa(\bar{x}(k)) \in \mathcal{X} \text{ for all } \bar{x}(k) \in \mathcal{X} \enspace .$$

Definition 4.3 (Robust positively invariant (RPI) set). A set $\mathcal{X} \subseteq \mathbb{R}^{n_x}$ is a *robust positively invariant (RPI)* set of (4.3) if

$$Ax(k) + B\kappa(x(k)) + w(k) \in \mathcal{X} \text{ for all } x(k) \in \mathcal{X}, w(k) \in \mathcal{W} \enspace .$$

The *minimal RPI (mRPI)* set is the RPI set that is contained in every closed RPI set of (4.3), and the PI set that contains every closed PI set of (4.5) is called a *maximal PI (MPI)* set.

Definition 4.4. Let $u = \kappa(x)$ be a control law for system (4.4) and $\mathcal{X} \subseteq \mathbb{R}^{n_x}$ a given subset of states. The set $\mathcal{X}^\kappa = \{x \mid x \in \mathcal{X}, \kappa(x) \in \mathbb{U}\}$ is called the *input admissible* set for $\mathcal{X}$.

The following standard function definitions are used, which are common in stability theory and can, for example, be found in [Vid93]:

Definition 4.5 ($\mathcal{K}$-class function). A function $\gamma : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ is of class $\mathcal{K}$ if it is continuous, strictly increasing and $\gamma(0) = 0$.

Definition 4.6 ($\mathcal{K}_\infty$-class function). A function $\gamma : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ is of class $\mathcal{K}_\infty$ if it is a $\mathcal{K}$-class function and $\gamma(s) \to \infty$ as $s \to \infty$.

Definition 4.7 ($\mathcal{KL}$-class function). A function $\beta : \mathbb{R}_{\geq 0} \times \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ is of class $\mathcal{KL}$ if, for each fixed $t \geq 0$, $\beta(\cdot, t)$ is of class $\mathcal{K}$, for each fixed $s \geq 0$, $\beta(s, \cdot)$ is non-increasing and $\beta(s, t) \to 0$ as $t \to \infty$.

4.2.1 Lyapunov Stability

Lyapunov stability theory is a well established tool in control theory for the analysis of dynamic systems. We state some of the main results for discrete-time systems that are employed in this thesis in the following. For more details on the topic we refer the reader to the standard textbooks, e.g. [Wil70, LaS76, Vid93, Kha02].

The concept most suitable for the control of the nominal system in (4.4) is that of asymptotic stability. Due to the fact that the analysis of constrained systems is considered, all results are not stated globally but are valid only in some positively invariant set $\mathcal{X}$.

Definition 4.8 (Asymptotic stability in $\mathcal{X}$). Given a PI set $\mathcal{X}$ including the origin as an interior-point, the equilibrium point x_s of system (4.5) is said to be *asymptotically stable* in $\mathcal{X}$ if it is

(Lyapunov) stable: For all $\epsilon > 0$ there exists a $\delta(\epsilon) > 0$ such that for every $x(0) \in \mathcal{X}$:

$$\|x(0) - x_s\| \leq \delta(\epsilon) \Rightarrow \|x(k) - x_s\| < \epsilon \; \forall k \in \mathbb{N} \ ,$$

attractive in $\mathcal{X}$: $\lim_{k\to\infty} \|x(k) - x_s\| = 0$ for all $x(0) \in \mathcal{X}$.

A stronger condition is that of exponential stability:

Definition 4.9 (Exponential stability in $\mathcal{X}$). Given a PI set $\mathcal{X}$ including the origin as an interior-point, the equilibrium point x_s of system (4.5) is said to be *exponentially stable* in $\mathcal{X}$ if there exist constants $\alpha > 0$ and $\gamma \in (0, 1)$ such that

$$\|x(k) - x_s\| \leq \alpha \|x(0) - x_s\| \gamma^k, \text{ for all } x(0) \in \mathcal{X} \text{ and for all } k \in \mathbb{N} \ .$$

Lyapunov's stability theorem establishes stability by showing the existence of a so-called Lyapunov function. Since any non-zero steady-state can be shifted to the origin by an appropriate coordinate transformation, we assume without loss of generality that the stability analysis considers the equilibrium point at the origin.

Definition 4.10 (Lyapunov function). Let $\mathcal{X}$ be a PI set for system (4.5) containing a neighborhood of the origin $\mathcal{N}(0)$ in its interior and let $\underline{\alpha}(\cdot)$, $\overline{\alpha}(\cdot)$, and $\beta(\cdot)$ be $\mathcal{K}_\infty$-class functions. A non-negative function $V : \mathcal{X} \to \mathbb{R}_+$ with $V(0) = 0$ is a *Lyapunov function* in $\mathcal{X}$ if:

$$V(x) \geq \underline{\alpha}(\|x\|) \quad \forall\, x \in \mathcal{X} \ , \tag{4.8a}$$

$$V(x) \leq \overline{\alpha}(\|x\|) \quad \forall\, x \in \mathcal{N}(0) \ , \tag{4.8b}$$

$$V(Ax + B\kappa(x)) - V(x) \leq -\beta(\|x\|) \; \forall\, x \in \mathcal{X} \ . \tag{4.8c}$$

Theorem 4.11 (Asymptotic / Exponential stability)

a) If system (4.5) *admits a Lyapunov function in $\mathcal{X}$, then the equilibrium point at* $x_s = 0$ *is* asymptotically stable *with region of attraction $\mathcal{X}$.*

b) If the inequalities in (4.8) *hold with* $\underline{\alpha}(\|x\|) = \underline{a}\|x\|^\lambda, \overline{\alpha}(s) = \overline{a}\|x\|^\lambda, \beta(s) = b\|x\|^\lambda$ *for some positive constants* $\underline{a}, \overline{a}, b, \lambda > 0$*, then the origin is* locally exponentially stable *for* $x \in \mathcal{N}(0)$*. Moreover, if inequality* (4.8b) *holds with* $\mathcal{N}(0) = \mathcal{X}$*, then the equilibrium point* $x_s = 0$ *is exponentially stable in $\mathcal{X}$.*

Theorem 4.11 is often exploited in control theory, when a Lyapunov function can be derived directly from the setup of the control problem.

Note that in this work, (asymptotic) stability of a system is sometimes used to mean that the system has a (asymptotically) stable equilibrium at the origin.

4.2.2 Input-to-State Stability

In the presence of persistent disturbances, asymptotic stability of the origin or an equilibrium point, respectively, can often not be achieved. Instead, it can be shown that under certain conditions the trajectories converge to an RPI set and stability of the uncertain system can be defined using the concept of *input-to-state stability (ISS).*

The framework of ISS was first introduced in [Son89] and has evolved into an important tool to investigate stability properties by analyzing the dependence of the state trajectories on the magnitude of the inputs, which can represent control variables as well as disturbances. Several variants have been developed and applied in different areas. The following description focuses on regional ISS which is best suited for the application of ISS to MPC, where regional denotes the fact that the stability results only hold in a certain region due to input and state constraints. More details on ISS can be found e.g. in [SW99, JW01] or in the context of Model Predictive Control in [MRS06, LAR+09].

The effect of the uncertainty makes the evolution of the system differ from what is expected using the nominal system model. In order for the system to have some robustness properties despite the disturbances, it is desirable that the effect on the system is bounded and depends on the size of the disturbance.

Definition 4.12 (Input-to-state stability (ISS) in Γ). Given a set $\Gamma \subseteq \mathbb{R}^n$ including the origin as an interior point, system (4.3) is called *Input-to-State Stable (ISS)* in Γ with respect to $w \in \mathcal{W}$ if there exists a $\mathcal{KL}$-class function $\beta(\bullet, \bullet)$ and a $\mathcal{K}$-class function $\gamma(\bullet)$ such that

$$\|\phi_\kappa(k, x(0), \mathbf{w})\| \le \beta(\|x(0)\|, k) + \gamma(\|\mathbf{w}_{[k-1]}\|) \tag{4.9}$$

for all initial states $x(0) \in \Gamma$, disturbance sequences $\mathbf{w} \triangleq [w_j]_{j\ge 0}$ with $w_j \in \mathcal{W}$ and $k \in \mathbb{N}$, where $\|\mathbf{w}_{[k-1]}\| \triangleq \max_{0\le j\le k-1}\{\|w_j\|\}$.

ISS generalizes different notions of stability for uncertain systems and allows for the analysis of the effect of persistent or decaying disturbances using one framework [LAR+09]. Note, that the condition for input-to-state stability reduces to that for asymptotic stability if $w_j = 0 \; \forall j \ge 0$. The system also converges asymptotically to the origin, if the disturbance signal fades.

The ISS concept permits the extension of a Lyapunov-like stability theory to disturbed systems, which is stated in the following.

Definition 4.13 (ISS Lyapunov function in Γ). A function $V : \mathbb{R}^n \to \mathbb{R}_+$ is called an *ISS Lyapunov function* in Γ with respect to $w \in \mathcal{W}$ for system (4.3) if Γ is an RPI set containing the origin in its interior and if there exists a compact set $\mathcal{S} \subseteq \Gamma$ including the origin as an interior point, suitable $\mathcal{K}_\infty$-class functions $\underline{\alpha}(\bullet), \overline{\alpha}(\bullet), \beta(\bullet)$ and a $\mathcal{K}$-class function $\gamma(\bullet)$ such that:

$$\begin{aligned} V(x) &\geq \underline{\alpha}(\|x\|) && \forall\, x \in \Gamma \ , && (4.10a)\\ V(x) &\leq \overline{\alpha}(\|x\|) && \forall\, x \in \mathcal{S} \ , && (4.10b)\\ V(Ax + B\kappa(x) + w) - V(x) &\leq -\beta(\|x\|) + \gamma(\|w\|) && \forall\, x \in \Gamma, w \in \mathcal{W} \ . && (4.10c) \end{aligned}$$

Theorem 4.14 (ISS in Γ [JW01, SW99]) *Consider the closed-loop system* (4.3). *If the system admits an ISS Lyapunov function in Γ with respect to $w \in \mathcal{W}$, then it is ISS in Γ with respect to $w \in \mathcal{W}$.*

A property that becomes of particular interest in control theory is the question if input-to-state stability of the uncertain system can be related to stability of the corresponding nominal system. While this is not generally true for nonlinear systems if no further conditions on the nominal system are imposed, this statement follows directly in the case of linear systems:

Theorem 4.15 *Consider the closed-loop system* (4.3). *If the corresponding nominal system* (4.5) *considering no disturbance admits a Lyapunov function which is uniformly continuous in $\mathcal{X}$, then system* (4.3) *is ISS in an RPI set $\Gamma \subseteq \mathcal{X}$ for a sufficiently small bound on the disturbance size.*

Proof. Follows directly from the fact that $f_\kappa(x, w) \triangleq Ax + B\kappa(x) + w$ is uniformly continuous in w and Theorem 2 in [LAR$^+$09]. ■

5 Model Predictive Control

Model Predictive Control (MPC) is a modern control technique that has been successfully applied in practice due to its ability to handle constraints. Numerous publications on theoretical aspects and applications have been published since the first introduction of the basic concept in the 1960s and the later success in the process industry in the 1980s and MPC has evolved into a mature control technique based on a well established theoretical foundation. For a detailed overview of the field see e.g. the books [Mac00, GSdD06, Ros03, KC01, CB04, RM09] or survey papers [GPM89, BM99, ML99, MRRS00, Raw00, May01].

MPC is an optimal control method, where the control action is obtained by solving a constrained finite horizon optimal control problem for the current state of the plant at each sampling time. The sequence of optimal control inputs is computed for a predicted evolution of the system model over a finite horizon. However, only the first element of the control sequence is applied and the state of the system is then measured again at the next sampling time. This so-called *receding horizon strategy* introduces feedback to the system, thereby allowing for compensation of potential modeling errors or disturbances acting on the system. While the basic idea of MPC is well established, there exist many variants for guaranteeing closed-loop feasibility, stability, robustness or reference tracking. In the following sections we will outline some of these variants for linear systems that are important for this thesis and outline the corresponding main theoretical results.

There has been an increasing interest in the development of fast MPC methods in recent years enabling MPC to be used not only for systems with slow dynamics, such as chemical processes, but also for fast sampled systems. A short survey on this topic will be given in Chapter 6.

5.1 Nominal MPC

Considering the nominal system in (4.4) a typical control problem is the regulation of the system state $x(k)$ to the origin while minimizing the control effort and respecting constraints on inputs and states. This can be formulated as the following nominal MPC problem:

Problem $\mathbb{P}_N(x)$ (Nominal MPC problem)

$$
\begin{aligned}
V_N^*(x) = \min_{\mathbf{x},\mathbf{u}} V_N(\mathbf{x},\mathbf{u}) &\triangleq \sum_{i=0}^{N-1} l(x_i,u_i) + V_f(x_N) && \text{(5.1a)}\\
\text{s.t.} \quad x_{i+1} &= Ax_i + Bu_i,\ \ i = 0,\dots,N-1\ , && \text{(5.1b)}\\
(x_i,u_i) &\in \mathbb{X}\times\mathbb{U},\ \ i = 0,\dots,N-1\ , && \text{(5.1c)}\\
x_N &\in X_f\ , && \text{(5.1d)}\\
x_0 &= x\ , && \text{(5.1e)}
\end{aligned}
$$

where $\mathbf{x} = [x_0, x_1, \cdots, x_N]$ and $\mathbf{u} = [u_0, \cdots, u_{N-1}]$ denote the state and input sequences, $l(x,u)$ is the stage cost, $V_f(x)$ is a terminal cost function and $X_f \subseteq \mathbb{X}$ is an invariant compact convex terminal target set. Problem $\mathbb{P}_N(x)$ implicitly defines the set of feasible control sequences $\mathcal{U}_N(x) \triangleq \{\mathbf{u}\,|\,\exists\ \mathbf{x}\text{ s.t. (5.1b)} - \text{(5.1e) hold}\}$ and feasible initial states $\mathcal{X}_N \triangleq \{x\,|\,\mathcal{U}_N(x) \neq \emptyset\}$. An input sequence $\mathbf{u}$ is called feasible for problem $\mathbb{P}_N(x)$ if $\mathbf{u} \in \mathcal{U}_N(x)$. The associated nominal state trajectory to a given control sequence $\mathbf{u}(x)$ at state x is $\mathbf{x}(x) \triangleq [x_0, \cdots, x_N]$, where $x_0 = x$ and for each i, $x_i = \bar{\phi}(i, x, \mathbf{u}(x))$.

Assumption 5.1. It is assumed that $l(\cdot,\cdot) : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$ is a convex function, $l(0,0) = 0$ and there exists a $\mathcal{K}$-class function $\alpha_l(\cdot)$ such that $l(x,u) \geq \alpha_l(\|x\|)$.

Common choices for the stage cost that are considered in this work are a quadratic and a linear norm cost.

Quadratic cost: The stage cost and terminal cost are given by quadratic functions

$$
l(x,u) \triangleq x^TQx + u^TRu,\ \ V_f(x) \triangleq x^TPx\ , \tag{5.2}
$$

where $Q \in \mathbb{R}^{n_x\times n_x}$ and $P \in \mathbb{R}^{n_x\times n_x}$ are symmetric positive semi-definite matrices and $R \in \mathbb{R}^{n_u\times n_u}$ is a symmetric positive definite matrix. For a quadratic cost, we can write (5.1) as a parametric Quadratic Program (pQP) of the form (3.27), where the current state $x \in \mathbb{R}^{n_x}$ is the parameter θ. For a given state $x \in \mathcal{X}_N$, problem $\mathbb{P}_N(x)$ reduces to a QP of the form (3.6).

Linear norm cost: The stage cost and terminal cost are composed of l_1- or l_∞-norms

$$
l(x,u) \triangleq \|Qx\|_p + \|Ru\|_p,\ \ V_f(x) \triangleq \|Px\|_p \text{ with } p \in \{1,\infty\}\ , \tag{5.3}
$$

where it is assumed that $Q \in \mathbb{R}^{n_x\times n_x}$ and $R \in \mathbb{R}^{n_u\times n_u}$ are non-singular matrices and $P \in \mathbb{R}^{r\times n_x}$ is a matrix with full column rank. For a 1- or ∞-norm cost, problem (5.1) can be rewritten as a parametric linear program (pLP) of the form (3.26), where the current state $x \in \mathbb{R}^{n_x}$ is again the parameter θ. For a given state $x \in \mathcal{X}$, problem $\mathbb{P}_N(x)$ can be transformed into an LP of the form (3.5), which is the reason why this cost function is sometimes referred to as a linear cost.

Solving problem $\mathbb{P}_N(x)$ for a given state $x \in \mathcal{X}_N$ yields an optimal control sequence $\mathbf{u}^*(x)$. Alternatively, an optimal control sequence $\mathbf{u}^*(x)$ for all feasible initial states $x \in \mathcal{X}_N$ can be obtained by solving the problem $\mathbb{P}_N(x)$ parametrically, which will be addressed in more detail in Chapter 6. The implicitly defined optimal MPC control law is then given in a receding horizon fashion by

$$\kappa(x) \triangleq u_0^*(x) \ . \tag{5.4}$$

Note that the variables x_i for $i = 0, \dots, N$ in (5.1) cannot be chosen independently, but are directly defined by the initial state, the input sequence $\mathbf{u}$ and the state update equation. These variables can hence be eliminated from the optimization problem, which reduces the number of optimization variables to the state sequence $\mathbf{u}$. However, while the cost and constraints in (5.1) have a very specific and sparse structure, in fact by column/row reordering they can be shown to be block-diagonal or banded, the sparsity is lost by the elimination of the equality constraints. We therefore refer to the problem in (5.1) as the sparse formulation and that with eliminated state variables as the dense formulation. The effect of the different formulations on the solution of the MPC problem will be discussed in Chapter 6. The structure of the sparse MPC formulation in Problem $\mathbb{P}_N(x)$ with a quadratic cost will be discussed in Chapter 8, for more details see also [RWR98, WB10].

5.2 Stability and Feasibility in Nominal MPC

It is well known that the infinite horizon optimal control problem, i.e. problem $\mathbb{P}_N(x)$ with $N \to \infty$, provides a control law that is guaranteed to asymptotically stabilize system (4.4) and is recursively feasible [RM93].

Definition 5.2 (Recursive Feasibility). A control law $\kappa(x)$ is called recursively feasible for $x(0)$ if $\kappa(x(k)) \in \mathbb{U}$ and $x(k) \in \mathcal{X}$ along the closed-loop trajectory $x(k+1) = Ax + B\kappa(x(k))$ for all $k \in \mathbb{N}$.

These properties are, however, not automatically inherited by the finite horizon MPC problem $\mathbb{P}_N(x)$ without any assumptions on the problem parameters $Q, R, N, V_f(\cdot)$ and $\mathcal{X}_f$. Rules for choosing the problem setup in order to recover feasibility and stability in finite horizon MPC problems are provided in the well-known survey paper [MRRS00].

In this work we use a terminal cost and terminal set approach applied in most of the recent MPC methods, where stability can be shown by employing the optimal cost function as a Lyapunov function. The idea is to choose the terminal cost, such that the MPC cost $V_N(\bullet, \bullet)$ in (5.1a) is approximately equal to or an upper bound on the infinite horizon cost in a neighborhood of the origin. The terminal set is chosen as an invariant subset of this neighborhood, where a local stabilizing control law $\kappa_f(\bullet)$ is known and all constraints are satisfied. This motivates the following conditions on the terminal cost function $V_f(\bullet)$ and terminal set $\mathcal{X}_f$.

Assumption 5.3. In the following it is assumed that $V_f(\bullet)$ is a continuous Lyapunov function in X_f, X_f is a PI set for system (4.4) under the control law $\kappa_f(x) = Kx$ and all state and control constraints are satisfied in $\mathcal{X}_f$. These conditions are stated formally as the following two assumptions:

A1: $X_f \subseteq \mathbb{X}$, $(A+BK)X_f \subseteq X_f$, $KX_f \subseteq \mathbb{U}$

A2: $V_f((A+BK)x) - V_f(x) \leq -l(x, Kx) \, \forall x \in X_f$,
$\exists$ $\mathcal{K}_\infty$-class functions $\underline{\alpha}_f, \overline{\alpha}_f$: $\underline{\alpha}_f(\|x\|) \leq V_f(x) \leq \overline{\alpha}_f(\|x\|)$.

If $V_f(\bullet)$ and $\mathcal{X}_f$ satisfy Assumption 5.3, it can be shown that the optimal cost function $V_N^*(x)$ in (5.1a) is a Lyapunov function and recursive feasibility and closed-loop stability of the nominal system is guaranteed in the feasible set $\mathcal{X}_N$, which is stated in the following theorem.

Theorem 5.4 (Stability under $\boldsymbol{\kappa(x)}$, [MRRS00]) *Consider Problem $\mathbb{P}_N(x)$ fulfilling Assumption 5.3. The cost function $V_N^*(x)$ is a Lyapunov function in $\mathcal{X}_N$, $\mathcal{X}_N$ is a PI set and the closed-loop system $x(k+1) = Ax(k) + B\kappa(x(k))$ is asymptotically stable with region of attraction $\mathcal{X}_N$.*

Outline of the proof:
While we do not repeat the entire proof of Theorem 5.4, we outline the main idea for showing that $V_N^*(x)$ satisfies condition (4.8c) in Definition 4.10, which is applied for proving stability of many different MPC variants in the literature and also in the remainder of this thesis. The proof is based on the use of an upper bound on the optimal cost function provided by the shifted solution from the state $x \in \mathcal{X}_N$ at the previous sampling time together with a local stable control law given by

$$\mathbf{u}^{\text{shift}}(x^+) = [u_1^*(x), \; \dots \; , u_{N-1}^*(x), \; \kappa_f(x_N^*(x))] \; , \tag{5.5}$$

which is shown to be a feasible suboptimal solution for Problem $\mathbb{P}_N(x^+)$, where $x^+ = Ax + B\kappa(x)$ is the state at the next sampling time. The cost for using this suboptimal solution at x^+ provides a decrease with respect to the optimal cost at x and so does therefore the optimal solution at x:

$$V_N^*(x^+) \leq V_N(\mathbf{x}^{\text{shift}}(x^+), \mathbf{u}^{\text{shift}}(x^+)) \leq V_N^*(x) - l(x, u_0^*(x)) \ . \tag{5.6}$$

The remainder of the proof shows that conditions (4.8a) and (4.8b) are also fulfilled and the optimal cost function is a Lyapunov function, proving asymptotic stability of the closed-loop system with region of attraction $\mathcal{X}_N$ by Theorem 4.11.

Corollary 5.5 *The optimal nominal MPC control law $\kappa(x)$ in 5.4 is recursively feasible.*

Many proposed MPC methods use the result in Theorem 5.3 and differ only in the choice of the ingredients $V_f(\bullet)$ and $\mathcal{X}_f$. In case of the quadratic cost in (5.1a), the control law and cost function of the unconstrained infinite horizon problem, i.e. the solution to the LQ control problem, is usually taken for $\kappa_f(x)$ and $V_f(x)$. The terminal set $\mathcal{X}_f$ can then for example be chosen as the maximal state and input admissible set described in [GT91] or a level set of $V_f(x)$, using the result that level sets of Lyapunov functions are invariant [Bla99]. In the case that a linear cost is chosen in (5.1a), a quadratic Lyapunov function is not suitable. Algorithms for computing $1-$ or $\infty-$norm based Lyapunov functions are presented for example in [Bor03, Chr07]. The terminal set can then again be taken as a level set of the Lyapunov function.

The use of a terminal penalty function not only provides stability but also reduces the deviation of the predicted open-loop from the closed-loop trajectory. If the horizon is chosen long enough, the optimal behavior is recovered and the predicted and actual behavior of the nominal system coincide [CM96,SR98]. In order to remove the terminal state constraint from the problem formulation, several methods propose to choose the horizon such that the terminal constraint is satisfied without explicitly including it in the optimization problem, see [MRRS00, HL02, LAC03] and the references therein. While this approach reduces the number of constraints, the horizon length is, however, either likely to be large in order to satisfy this property for all feasible states $x \in \mathcal{X}_N$, or for a given N it is difficult to analyze the region in which this property is satisfied.

Other versions of stabilizing model predictive control setups include variable horizon MPC, contractive MPC or stability enforced MPC, see [MRRS00] and the references therein.

5.3 Robust MPC

In practice, model inaccuracies or disturbances usually cause violation of the nominal system dynamics in (4.4) which can lead to the loss of (recursive) feasibility and stability of the nominal optimal MPC control law. The success of MPC in many application areas motivated the development of MPC approaches with enhanced robustness characteristics in order to extend the guarantees provided for nominal MPC also to the uncertain case. This is addressed in so-called robust MPC schemes that recover recursive feasibility and thereby stability by changing the problem formulation.

Consider the discrete-time uncertain system in (4.1). The goal of robust MPC is to provide a controller that satisfies the state and input constraints and achieves some form of stability despite disturbances that are acting on the system. Asymptotic stability of the origin cannot be achieved in the presence of disturbances. It can, however, be shown that under certain conditions the trajectories converge to an RPI, which can be seen as the 'origin' for the uncertain system. This is captured in the concept of ISS introduced in Section 4.2, requiring the nominal system to be asymptotically stable and the influence of the disturbance on the evolution of the states to be bounded.

There is a vast literature on the synthesis of robust MPC controllers, see e.g. [BM99, MRRS00, MS07, LAR$^+$09] for a good overview of the different approaches. The available methods can generally be classified into three groups. The first group is formed by open-loop robust MPC methods that use the nominal MPC cost and tighten the constraints in order to achieve constraint satisfaction for all possible realizations of the disturbance w over the horizon (see e.g. [CRZ01, LAC02]). The resulting solutions are in general, however, very conservative. The second group is the so-called min-max MPC, where the MPC cost is maximized over all possible disturbance sequences (see e.g. [MRRS00, LAR$^+$09] and the references therein). Whereas in open-loop min-max a worst-case control sequence is optimized, closed-loop min-max approaches optimize for a sequence of control policies and thereby introduce feedback to the disturbance. Min-max approaches have the important drawback that they are computationally very demanding. The third group is a variant of open-loop robust MPC, the so-called tube-based approaches (see e.g. [MSR05, RTMA06]). A feedback term is introduced in order to bound the effect of the disturbances and reduce the conservativeness.

Since the aim of this work is the development of real-time MPC methods that allow for fast computation, we restrict the following description to the tube based robust MPC approach described in [MSR05] that is particularly suited for linear systems. The main steps of the procedure are outlined in the following section.

5.3.1 Tube-based Robust MPC for Linear Systems

The method is based on the use of a feedback policy of the form $u = \bar{u} + K(x - \bar{x})$ that keeps the states x of the uncertain system in (4.1) close to the states $\bar{x}$ of the nominal system (4.4). Loosely speaking, the controlled uncertain system will stay within a so-called tube with constant section $\mathcal{Z}$ and centers $\bar{x}(k)$, where $\mathcal{Z}$ is an RPI set for system $x(k+1) = (A + BK)x(k) + w(k)$. The robust MPC problem can therefore be reduced to the optimization of the tube centers, which are steered to the origin by choosing a sequence of control inputs $\bar{\mathbf{u}}$ and the initial tube center $\bar{x}(0)$. It can be shown that if the initial center is chosen according to the constraint $x = x(0) \in \bar{x}(0) \oplus \mathcal{Z}$ for a given initial state x then the trajectory of the uncertain system remains within the described tube (in fact for all $\bar{\mathbf{u}}, x(i) \in \bar{x}(i) \oplus \mathcal{Z}$ if $x(0) \in \bar{x}(0) \oplus \mathcal{Z}$). This can be formulated as a standard MPC problem with the following modifications:

- The first state $\bar{x}_0$ is also an optimization variable representing the tube center for the current state x.
- In order to guarantee that the uncertain system does not violate the constraints in 5.1 the constraints for the tube centers must be tightened in the following way:

$$\bar{\mathbb{X}} = \mathbb{X} \ominus \mathcal{Z},\ \bar{\mathbb{U}} = \mathbb{U} \ominus K\mathcal{Z} \ . \tag{5.7}$$

  The tightening of the constraints also affects the size of the terminal set, which is denoted as $\bar{\mathcal{X}}_f$.

This results in the following robust MPC problem $\mathbb{P}_N^r(x)$ which is a modification of $\mathbb{P}_N(x)$:

Problem $\mathbb{P}_N^r(x)$ [Robust MPC problem]

$$V_N^{r*}(x) = \min_{\bar{\mathbf{x}},\bar{\mathbf{u}}} \quad V_N(\bar{\mathbf{x}}, \bar{\mathbf{u}}) + V_f(x - \bar{x}_0) \tag{5.8a}$$

$$\text{s.t.} \quad \bar{x}_{i+1} = A\bar{x}_i + B\bar{u}_i,\ \ i = 0, \dots, N-1 \ , \tag{5.8b}$$

$$(\bar{x}_i, \bar{u}_i) \in \bar{\mathbb{X}} \times \bar{\mathbb{U}},\ \ i = 0, \dots, N-1 \ , \tag{5.8c}$$

$$\bar{x}_N \in \bar{\mathcal{X}}_f \ , \tag{5.8d}$$

$$x \in \bar{x}_0 \oplus \mathcal{Z} \ . \tag{5.8e}$$

Problem $\mathbb{P}_N^r$ implicitly defines the set of feasible control sequences $\mathcal{U}_N^r(\bar{x}_0) = \{\bar{\mathbf{u}} \,|\, \exists\, \bar{\mathbf{x}}\,\text{s.t.}\ (5.8\text{b}) - (5.8\text{d})\,\text{hold}\}$, feasible initial tube centers $\mathcal{X}_0^r(x) \triangleq \{\bar{x}_0 \,|\, (5.8\text{e})\}$ and feasible initial states $\mathcal{X}_N^r \triangleq \{x \,|\, \exists\, \bar{x}_0 \in \mathcal{X}_0^r(x) \text{ s.t. } \mathcal{U}_N^r(\bar{x}_0) \neq \emptyset\}$.

The resulting robust MPC control law is then given in a receding horizon control fashion by:

$$\kappa^r(x) = \bar{u}_0^{r*}(x) + K(x - \bar{x}_0^{r*}(x)) \ , \tag{5.9}$$

where $\bar{\mathbf{u}}^{r*}(x)$ and $\bar{x}_0^{r*}(x)$ is the optimal solution to $\mathbb{P}_N^r(x)$ and K is such that $A + BK$ is stable, which can be chosen as the feedback gain of the local control law $\kappa_f(x)$ in Assumption 5.3.

Note that the optimal initial center $\bar{x}_0^{r*}(x)$ is not necessarily equal to the current state x. The re-optimization of the tube center at every time step introduces feedback to the disturbance.

Compared to the robust MPC approach in [MSR05], we propose to augment the cost in (5.8a) by the term $V_f(x-\bar{x}_0)$, introducing a tradeoff between the amount of control action used for counteracting the disturbance and the effort for controlling the tube centers to the origin. The main advantage of the augmented cost is that it directly provides a ISS Lyapunov function for the closed-loop system under the robust MPC control law (Theorem 5.8).

If the RPI set $\mathcal{Z}$ is polytopic, the robustified Problem $\mathbb{P}_N^r(x)$ preserves the problem structure of $\mathbb{P}_N(x)$ and can still be cast as an LP or QP, respectively, depending on the choice of the cost function. Ideally $\mathcal{Z}$ would be taken as small as possible, i.e. as the minimal RPI (mRPI) set, which is, however, not necessarily polytopic and an explicit representation can generally not be computed. In [MSR05] the computation of a polyhedral RPI set for $\mathcal{Z}$ using the method in [RKKM05] is proposed.

Considering the robust MPC problem $\mathbb{P}_N^r(x)$, the stability results for nominal MPC can be directly extended to the case of uncertain systems by modifying Assumption 5.3 according to the new problem setup.

Assumption 5.6. It is assumed that $V_f(\bullet), \bar{\mathcal{X}}_f$ fulfill Assumption 5.3 with $\bar{\mathbb{X}}$, $\bar{\mathbb{U}}$ and $\bar{\mathcal{X}}_f$ replacing $\mathbb{X}$, $\mathbb{U}$ and $\mathcal{X}_f$.

Remark 5.7. The set $\bar{\mathcal{X}}_f \oplus \mathcal{Z}$ is an RPI set for system $x(k+1) = (A+BK)x(k)+w(k)$.

Theorem 5.8 (Stability under $\boldsymbol{\kappa^r(x)}$) *Consider Problem $\mathbb{P}_N^r(x)$ fulfilling Assumption 5.6. The closed-loop system $x(k+1) = Ax(k) + B\kappa^r(x(k)) + w(k)$ is ISS in $\mathcal{X}_N^r$ with respect to $w(k) \in \mathcal{W}$.*

Proof. The proof is stated in the following assuming a quadratic stage and terminal cost in (5.8a), for a linear norm in (5.3) cost the result can be obtained following the same steps. From convexity we get $\frac{1}{2}\|x+y\|_Q^2 \leq \|x\|_Q^2 + \|y\|_Q^2$. From Assumption 5.6 we have that $V_f(x)$ is an upper bound on the infinite horizon cost using the local control law $\kappa_f(x)$ and further $\|x\|_Q^2 \leq \|x\|_P^2$. It then follows that there exist a $\mathcal{K}_\infty$-class function $\underline{\alpha}(\bullet)$, such that

$$V_N^{r*}(x) \geq \|\bar{x}_0^{r*}(x)\|_Q^2 + \|x - \bar{x}_0^{r*}(x)\|_P^2 \tag{5.10}$$

$$\geq \|\bar{x}_0^{r*}(x)\|_Q^2 + \|x - \bar{x}_0^{r*}(x)\|_Q^2 \geq \frac{1}{2}\|x\|_Q^2 \geq \underline{\alpha}(\|x\|) \;\; \forall x \in \mathcal{X}_N^r \; . \tag{5.11}$$

In order to derive an upper bound, we make use of the following considerations. Take $\mathcal{S} \triangleq \{x \mid \|x\|_2 \leq \epsilon\}$, where $\epsilon > 0$ is such that $\mathcal{S} \subseteq \mathcal{Z}$. Then $V_N^{r*}(x) \leq V_N(0,0)+V_f(x) =$

$V_f(x)$, since $\bar{x}_0 = 0, \bar{u}_i = 0 \ \forall i = 0, \ldots, N-1$ is a feasible solution. Therefore, there exists a suitable $\mathcal{K}_\infty$-class function $\overline{\alpha}(\bullet)$, such that

$$V_N^{r*}(x) \leq V_f(x) \leq \overline{\alpha}(\|x\|) \ \forall x \in \mathcal{S} \ . \tag{5.12}$$

Feasibility of the shifted solution $\bar{\mathbf{u}}^{\text{shift}}(x^+) = [\bar{u}_1^{r*}(x), \ldots, \bar{u}_{N-1}^{r*}(x), K\bar{x}_N^{r*}(x)]$, $\bar{x}_0^{\text{shift}}(x^+) = \bar{x}_1^{r*}(x)$ for $\mathbb{P}_N^r(x^+)$, with $x^+ \in Ax + B\kappa^r(x) \oplus \mathcal{W}$, was shown in [MSR05]. Since $V_f(x)$ is a continuous function, there exists a $\mathcal{K}$-class function $\gamma(\bullet)$ such that $|V_f(y) - V_f(x)| \leq \gamma(\|y-x\|)$. Using Assumption 5.3 and the fact that

$$\begin{aligned} x^+ &= Ax - A\bar{x}_0^{r*}(x) + A\bar{x}_0^{r*}(x) + B\bar{u}_0^{r*}(x) + BK(x - \bar{x}_0^{r*}(x)) + w \\ &= \bar{x}_1^{r*}(x) + (A+BK)(x - \bar{x}_0^{r*}(x)) + w \ , \end{aligned}$$

we obtain

$$\begin{aligned} &V_f(x^+ - \bar{x}_1^{r*}(x)) - V_f(x - \bar{x}_0^{r*}(x)) \\ &= V_f((A+BK)(x - \bar{x}_0^{r*}(x)) + w) - V_f((A+BK)(x - \bar{x}_0^{r*}(x))) \\ &\quad + V_f((A+BK)(x - \bar{x}_0^{r*}(x))) - V_f(x - \bar{x}_0^{r*}(x)) \\ &\leq -\|x - \bar{x}_0^{r*}(x)\|_Q^2 + |V_f((A+BK)(x - \bar{x}_0^{r*}(x)) + w) - V_f((A+BK)(x - \bar{x}_0^{r*}(x)))| \\ &\leq -\|x - \bar{x}_0^{r*}(x)\|_Q^2 + \gamma(\|w\|) \ . \end{aligned}$$

By comparing the sequences using Assumption 5.3 and standard arguments in MPC (see also the proof of Theorem 1 in [MSR05]), it can then be shown that there exists a $\mathcal{K}_\infty$-class function $\beta(\bullet)$, such that

$$\begin{aligned} V_N^{r*}(x^+) - V_N^{r*}(x) \leq & V_N(\bar{\mathbf{x}}^{\text{shift}}(x), \bar{\mathbf{u}}^{\text{shift}}(x)) - V_N(\bar{\mathbf{x}}^{r*}(x), \bar{\mathbf{u}}^{r*}(x)) \\ & + V_f(x^+ - \bar{x}_1^{r*}(x)) - V_f(x - \bar{x}_0^{r*}(x)) \\ \leq & - \|\bar{x}_0^{r*}(x)\|_Q^2 - \|x - \bar{x}_0^{r*}(x)\|_Q^2 + \gamma(\|w\|) \\ \leq & - \frac{1}{2}\|x\|_Q^2 + \gamma(\|w\|) \leq -\beta(\|x\|) + \gamma(\|w\|) \ \forall x \in \mathcal{X}_N^r \ . \end{aligned}$$

The optimal cost is hence a ISS Lyapunov function and by Theorem 4.14 the closed-loop system is ISS. ■

5.4 MPC for Tracking

Many control applications in practice require tracking of a desired sequence of steady-states rather than regulation around the origin or a particular steady state. In order to achieve tracking of piecewise constant references, the MPC problem (5.1) can be modified by means of a change of variables into penalizing the deviation from the state and input reference and requiring the terminal state to lie in a terminal set around the state reference [Mac00]. For small reference changes this approach allows one to steer the system to the new reference point. There are, however, two main disadvantages. If the reference changes significantly, it might not be reachable from the current state with the given horizon length and terminal state constraint and recursive feasibility may be lost. In addition, the solution computed at the previous time instant may not be feasible for the current state measurement after a reference change, since the terminal constraint depends on the current reference value, which leads to a loss of the stability guarantee by means of the standard stability proof in MPC. This property is also particularly relevant when using warm-start techniques to solve the MPC problem, which will be discussed in more detail in Chapter 6.

Several approaches have been proposed in the literature for recovering feasibility of the MPC control system after a reference change, see e.g. [BCM97, GK99, CZ03] and the references therein. One approach is the introduction of a reference governor that provides an artificial reference to the system ensuring convergence to the desired reference target while satisfying the constraints. In the following we focus on the method introduced in [LAAC08], which provides recursive feasibility despite a significant reference change, feasibility of the solution from the previous state measurement and an enlarged feasible set. These features are exploited in the real-time and soft-constrained MPC approaches developed in Chapter 8 and Chapter 9. The main ideas and properties of the method are outlined in the following section, for a detailed description see [LAAC08].

5.4.1 MPC for Tracking of Piecewise Constant References

The main difference to other tracking approaches is the introduction of an artificial reference into the optimization problem. An optimal artificial reference and control sequence is then computed in one optimization problem, allowing the artificial reference to deviate from the real reference if the latter is not a feasible target from the current state. This provides not only recursive feasibility but also renders the shifted solution computed at the previous time instant feasible for the current state measurement.

The tracking approach can be formulated in the form of an MPC problem, where the following components are introduced:

- The artificial steady state and input are introduced as decision variables in the MPC problem.
- The cost penalizes the deviation from the states and inputs to the artificial reference instead of the real reference.

- An offset term for the deviation between the artificial and the real reference is then added to the MPC cost in order to ensure convergence to the desired steady state.
- A terminal weight on the deviation between the terminal state and artificial reference as well as an extended terminal constraint on the terminal state that depends on the artificial reference provide stability of the optimal MPC controller.

We consider the task of tracking a piecewise constant reference signal, by steering the system state x to the target steady-state x_r. A target input u_r is associated to every target state x_r fulfilling the steady-state condition in (4.6). The state and input constraints limit the set of feasible steady-states to $(x_r, u_r) \in \mathbb{X} \times \mathbb{U}$.

The resulting MPC problem for reference tracking $\mathbb{P}_N^{tr}(x)$ is then given by:

Problem $\mathbb{P}_N^{tr}(x)$ [MPC for reference tracking]

$$V_N^{tr}(\mathbf{x}, \mathbf{u}, x_s, u_s, x_r, u_r) \triangleq \sum_{i=0}^{N-1} l(x_i - x_s, u_i \ - u_s) + V_f(x_N - x_s) + V_o(x_r - x_s, u_r - u_s) \tag{5.13a}$$

$$\begin{aligned} V_N^{tr*}(x, x_r, u_r) = \min_{\mathbf{x},\mathbf{u},x_s,u_s} \quad & V_N^{tr}(\mathbf{x}, \mathbf{u}, x_s, u_s, x_r, u_r) && (5.13b) \\ \text{s.t.} \quad & x_{i+1} = Ax_i + Bu_i, \ \ i = 0, \ldots, N-1 \ , && (5.13c) \\ & (x_i, u_i) \in \mathbb{X} \times \mathbb{U}, \ \ i = 0, \ldots, N-1 \ , && (5.13d) \\ & x_N \in \mathcal{X}_f^{tr}(x_s, u_s) \ , && (5.13e) \\ & (x_s, u_s) \in \Theta \ , && (5.13f) \\ & x = x_0 \ , && (5.13g) \end{aligned}$$

where (x_s, u_s) denotes the artificial steady-state, (x_r, u_r) is the desired target steady-state and $V_o(x_r - x_s, u_r - u_s)$ is the tracking offset cost.

$$\Theta \triangleq \{(x_s, u_s) \mid x_s \in \operatorname{int} \mathbb{X}, u_s \in \operatorname{int} \mathbb{U}, (A - I)x_s + Bu_s = 0\} \tag{5.14}$$

is the set of feasible steady states and $\mathcal{X}_f^{tr}(x_s, u_s)$ is an invariant terminal target set for tracking that is parameterized by the artificial reference (x_s, u_s). Problem $\mathbb{P}_N^{tr}(x)$ implicitly defines the set of feasible control sequences $\mathcal{U}_N^{tr}(x, x_s, u_s) \triangleq \{\mathbf{u} \mid \exists\, \mathbf{x} \text{ s.t. (5.13c)–(5.13e), (5.13g) hold}\}$ and feasible initial states $\mathcal{X}_N^{tr} \triangleq \{x \mid \exists\, (x_s, u_s) \in \Theta \text{ s.t. } \mathcal{U}_N^{tr}(x, x_s, u_s) \neq \emptyset\}$. Note that we use a slightly different formulation in the problem statement of $\mathbb{P}_N^{tr}(x)$ than proposed in [LAAC08], which does not employ a parametrization of the steady-state (x_s, u_s).

The resulting MPC control law for tracking is given in a receding horizon fashion by:

$$\kappa^{tr}(x) = u_0^{tr*}(x) \ , \tag{5.15}$$

where $\mathbf{u}^{tr*}(x)$ is the optimal solution to $\mathbb{P}_N^{tr}(x)$.

In order to show stability and convergence of the MPC scheme for tracking, Assumption 5.3 is adapted to the new problem setup.

Assumption 5.9. It is assumed that for a given $(x_s, u_s) \in \Theta$, $V_f(x - x_s)$ is a Lyapunov function in $\mathcal{X}_f^{tr}(x_s, u_s)$ satisfying Assumption 5.3 and that $\mathcal{X}_f^{tr}(x_s, u_s)$ is a PI set for the nominal system (4.4) under the local control law for tracking $\kappa_f^{tr}(x) = K(x - x_s) + u_s$ containing x_s in its interior, which can be stated as the following conditions:

A3: $\mathcal{X}_f^{tr}(x_s, u_s) \subseteq \mathbb{X}, x_s \in \operatorname{int} \mathcal{X}_f^{tr}(x_s, u_s),$
$Ax + B\kappa_f^{tr}(x) \in \mathcal{X}_f^{tr}(x_s, u_s), \kappa_f(x) \in \mathbb{U} \ \forall x \in \mathcal{X}_f^{tr}(x_s, u_s)$.

Assumption 5.10. It is assumed that $V_o(\cdot, \cdot) : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$ is a convex function, $V(0, 0) = 0$ and there exists a $\mathcal{K}$-class function $\alpha_o(\cdot)$ such that $V_o(x, u) \geq \alpha_o(\|x\|)$.

Convergence of the closed-loop system to the steady-state (x_r, u_r) is then provided by the following theorem.

Theorem 5.11 (Convergence of $\kappa^{tr}(x)$, [LAAC08]) *Consider Problem $\mathbb{P}_N^{tr}(x)$ fulfilling Assumptions 5.9 and 5.10. The closed-loop system $x(k+1) = Ax(k) + B\kappa^{tr}(x(k))$ converges to x_r $\forall x \in \mathcal{X}_N^{tr}$, i.e. x_r is asymptotically stable with region of attraction $\mathcal{X}_N^{tr}$.*

A further advantage of the described tracking scheme is that it provides an enlarged domain of attraction compared to a standard MPC approach for reference tracking, i.e. $\mathcal{X}_N^{tr} \supseteq \mathcal{X}_N$, due to the use of an artificial reference and corresponding target set [LAAC08]. The tracking approach can be directly combined with the tube based robust MPC method described in Section 5.3.1, which was presented in [ALA$^+$07].

Choice of the Offset Cost $V_o(\cdot, \cdot)$:

In order for problem $\mathbb{P}_N^{tr}(x)$ to be cast as a convex optimization problem, the offset cost and the terminal set have to be chosen appropriately.

We consider two main choices for the offset cost:

- the sum of weighted l_1- or l_∞-norms: $V_o(x, u) = \|T_x x\|_p + \|T_u u\|_p$, where $p \in \{1, \infty\}$ and $T_x \in \mathbb{R}^{n_x \times n_x}$ and $T_u \in \mathbb{R}^{n_u \times n_u}$ are nonsingular matrices,
- the sum of quadratic costs: $V_o(x, u) = x^T T_x x + u^T T_u u$, where $T_x \in \mathbb{R}^{n_x \times n_x}$ and $T_u \in \mathbb{R}^{n_u \times n_u}$ are positive definite matrices,

or a combination of the two options. While a quadratic offset cost can be beneficial especially if the MPC cost is quadratic, an l_1- or l_∞-norm provide an exact penalty function for an appropriately chosen weight and can thereby enforce the optimal behavior of the system, i.e. $(x_s^{tr*} u_s^{tr*}) = (x_r, u_r)$, wherever the given reference is feasible and can be enforced. This is sometimes referred to as the local optimality property in the literature. The use of l_1- or l_∞-norms for the offset cost and its consequences are described in more detail in [FLA+09].

Computation of the Terminal Set for Tracking $\mathcal{X}_f(x_s, u_s)$:

The authors in [LAAC08] propose a method to compute a polytopic invariant set for tracking $\mathcal{X}_f(x_s, u_s)$, by using a parametrization of the steady-state capturing the dependence of x_s and u_s through the steady-state equation in (4.6). We describe a slight modification of this approach in the following that we feel is more transparent.

Using the local control law $\kappa_f^{tr}(x)$ in Assumption 5.9 the closed-loop system is given by

$$x^+ - x_s = (A + BK)(x - x_s) \ ,$$

i.e. we obtain the same closed-loop dynamics considered in Assumption 5.3 for the regulation case for the shifted system $x - x_s$. The main difference is in the constraints, which are now dependent on the steady-state (x_s, u_s).

Consider that the steady-state is parameterized by the parameter θ as described in Section 4.1: $x_s = M_x\theta, u_s = M_u\theta$. One possibility to compute the invariant set $\mathcal{X}_f^s(x_s, u_s)$ is to consider the augmented system $v^T = [\Delta x^T \ \theta^T]$, where $\Delta x = x - x_s$, with the closed-loop dynamics

$$v^+ = A_{v,K}v, \text{ where } A_{v,K} = \begin{bmatrix} A + BK & 0 \\ 0 & I \end{bmatrix} \ . \qquad (5.16)$$

Condition A3 in Assumption 5.9 can then be reformulated as

$$\Omega_f \subseteq \mathcal{V}_\alpha, v^+ \in \Omega_f \ \forall \ v \in \Omega_f \ ,$$

where $\mathcal{V}_\alpha = \{(\Delta x, \theta) \mid (\Delta x + M_x\theta) \in \mathbb{X}, K\Delta x + M_u\theta \in \mathbb{U}, M_x\theta \in \alpha\mathbb{X}, M_u\theta \in \alpha\mathbb{U}\}$ ensures satisfaction of the state and input constraints for some positive parameter $\alpha \in (0, 1)$ providing that (x_s, u_s) is in the interior of $\mathbb{X} \times \mathbb{U}$. An invariant set Ω_f can then be computed using standard methods, see the discussion at the end of Section 5.2.

The terminal set for tracking $\mathcal{X}_f^s(x_s, u_s)$ is directly obtained from Ω_f using the relationship $\theta = M_x^T x_s + M_u^T u_s$. For a polytopic set Ω_f, the terminal set for tracking results in a polytopic set of the form

$$\mathcal{X}_f^s(x_s, u_s) \triangleq \{x \mid Gx \leq f - f_x x_s - f_u u_s\}$$

and problem $\mathbb{P}_N^{tr}(x)$ can be transformed into a standard LP or QP (under the previously described choice of $V_o(\cdot, \cdot)$).

The authors in [LAAC08] propose the computation of the maximal state and input admissible invariant set for a similar augmented system, which in our case is given by $\mathcal{O}^v_{\infty,\alpha} = \{v \mid A^t_{v,K} v \in \mathcal{V}_\alpha, t \in \mathbb{N}\}$. For any $\alpha \in (0,1)$ this set is finitely determined, i.e. determined by a finite recursive procedure, and results in a polyhedron [GT91]. Note that for $\alpha = 1$ it would not be finitely determined due to the unitary eigenvalues of $A_{v,K}$. It further holds that $\alpha\mathcal{O}^v_\infty \subset \mathcal{O}^v_{\infty,\alpha} \subset \mathcal{O}^v_{\infty,1}$ [LAAC08].

The computation of an invariant set for tracking will be further discussed in Chapter 8 and in the context of soft constraints in Chapter 9.

Part II

Real-time MPC for Linear Systems

6 Fast MPC – Survey

The computation of a model predictive control law, as described in Chapter 5, amounts to the solution of a convex optimization problem at each sampling instant. Reliable general purpose solvers are available for this class of optimization problems and classically MPC problems are solved by directly applying one of the optimization routines online, which has, however, restricted the applicability of MPC to slow dynamic processes. In recent years, various methods have been developed with the goal of enabling MPC to be used for fast sampled systems, which are called fast MPC methods in the following. These approaches can generally be classified into two main paradigms: *explicit MPC* and *online MPC* methods. We will give an overview of the main ideas presented in the literature in the following, with an emphasis on online MPC methods, which are relevant for the work presented in Chapters 7 and 8.

6.1 Explicit MPC

The goal of *explicit MPC* is to remove the main limitation of MPC, namely the solution of an optimization problem online, and move the computational burden offline. The optimal control action is determined as a function of the state for a given pre-defined (compact) set, the so-called explicit solution, which can be computed by means of *parametric programming* methods (see Section 3.1). It has been shown that in special cases when the parametric optimal control problem can be formulated as a parametric LP or QP of the form (3.26) or (3.27), which is e.g. obtained for the MPC problem $\mathbb{P}_N(x)$ described in Section 5.1 with a quadratic or linear norm cost, the resulting optimal control law is a piecewise affine function of the state x defined over a polyhedral partition $\mathcal{P}$ of the feasible set of states [BBM02, BMDP02]:

$$u_0^*(x) \triangleq F^j x + g^j, \text{ if } x \in P_j \; \forall \; P_j \in \mathcal{P} \; . \tag{6.1}$$

Algorithms for computation of the parametric solution to these problem classes in the context of model predictive control are presented in [Bor03, BMDP02, Bao02, TJB03a] for parametric QPs (pQPs) and in [Sch87, BBM02, BBM03, JBM07] for parametric LPs (pLPs). A method to solve parametric QPs and LPs based on linear complementarity problem reformulations is proposed in [JM06].

The explicit control law in (6.1) can be stored as a lookup table. This reduces the online computational effort to locating the measured initial state in the polyhedral partition $\mathcal{P}$, the so-called point location problem, and an affine function evaluation. The main online effort is given by the point location, which depends on the complexity of the partition, i.e. the number of state-space regions over which the control law is defined, and has to be performed efficiently in order to obtain high-speed computation times. This topic is discussed in [BBBM08, TJB03b, JGR06, WJM07] and the recent work [NFJM10]. A good survey on explicit MPC can be found in [AB09].

Advantages
By pre-computing the optimal solution, explicit MPC can provide extremely high sampling rates, enabling MPC to be used for high-speed systems. In addition, the explicit solution offers the main advantage that control-relevant properties such as closed-loop feasibility, stability, robustness and performance of the MPC control law can be certified in a post-processing analysis, which is of particular interest in safety critical applications. In addition, the look-up table offers an easy and cheap implementation.

Disadvantages
Both the computation time and the complexity of the partition grow in the worst case exponentially with the problem size (length of the prediction horizon, dimension of the states and inputs, number of constraints) due to the combinatorial nature of the problem [BBM02]. This limits the applicability of explicit MPC to relatively small problem dimensions. If the explicit solution can be computed, it can still be highly complex, which may prohibit its application due to restricted storage space and online computation time.

This has given rise to an increasing interest in the development of new methods to approximate explicit solutions. The following section gives an overview of some of the various approaches in the literature, see also the survey in [AB09].

Approximate Explicit MPC

Approximate explicit methods inherit the advantages of optimal explicit MPC while trying to eliminate the main disadvantage and reduce the complexity of the explicit solutions. They can generally be classified into two groups: techniques post-processing the optimal explicit solution and methods that directly allow for sub-optimality during its computation. In the first category, [GTM08] derive a minimal representation of the explicit solution that is obtained by optimally merging regions with the same affine control law. In [CZJM07], the number of regions is reduced by eliminating regions as long as stability can be guaranteed using a neighboring control law. Post-processing techniques suffer from the main disadvantage that they are based on the optimal parametric solution, which might be prohibitively complex and can often not be computed.

Direct methods in turn do not require the explicit solution and generally offer a higher reduction in complexity than post-processing methods. In [BF03] a sub-optimal solution to the pQP is obtained by relaxing the KKT conditions, which is shown to result in reduced complexity solutions. Determination of an approximate explicit solution by means of recursive rectangular partitions of the state space is proposed in [JG03, Joh04] and using simplicial partitions in [BF06]. The more recent approach in [SJLM09] is conceptually similar to [Joh04] but improves the approach by providing an approximate control law through barycentric function interpolation allowing for an evaluation of feasibility and stability. The method in [GM03] solves MPC subproblems of horizon length $N = 1$ sequentially and similarly [GKBM04] is based on a minimum-time control approach. Approximate pLP solutions obtained by relaxing optimality within a predefined bound using a dynamic programming formulation are proposed in [LR06]. [JBM07] present an approximation method for pLPs based on the beneath/beyond (B/B) algorithm, which is extended to generic convex cost functions in [JM10], where the computation of inner and outer polyhedral approximations is proposed using a modification of the double description method. In [PRW07], a complexity reduction is achieved by means of partial enumeration of all possible combinations of active constraints after identifying the most important combinations of active constraints in off-line simulation. A combination of this approach with [CZJM07] is presented in [AB09]. [JM09] propose a method for directly approximating the non-convex optimal control law rather than the optimal cost function based on bilevel optimization.

Approximate explicit MPC methods successfully reduce the complexity of explicit approaches and have the advantage of providing verifiable approximate solutions that can be analyzed for stability and performance in a post-processing step. All described approaches are, however, still limited to smaller and medium problem dimensions due to the inherent explicit character of the algorithms. In addition, even though a desired complexity can be imposed for some of the methods, an explicit solution providing stability or a pre-specified performance is likely to be highly complex or may even be intractable. In order to address these issues, there has been a reconsideration of online approaches and the development of fast online MPC methods, which will be discussed in the next section.

6.2 Fast Online MPC

Online MPC methods have the significant advantage that they can be applied to all problem dimensions. The challenge is their application to high-speed systems requiring very fast sampling rates. Various approaches trying to reduce the computation time in online MPC have been proposed recently. Many methods are based on the idea that a significant reduction of the computational complexity can be achieved by exploiting the particular structure or sparsity of the optimization problem that is inherent in the MPC problem. Considerable computational savings can also be achieved by means of so-called warm-start techniques trying to reduce the total number of iterations to the optimal solution by starting the optimization from a good initial point. In the context of MPC, this idea is motivated by the fact that a sequence of similar optimization problems is solved, suggesting that the solution computed at one time instant shifted one step forward can be a good initial approximation for the solution at the next time instant.

An overview of some of the ideas proposed in the literature is given in the following, organized by the employed optimization method. The focus is on the case of linear MPC with a quadratic or linear norm cost and fast MPC schemes based on interior-point and active set methods, the two currently most prominent iterative algorithms, whose properties with respect to computational complexity and warm-starting are outlined. A comparison between active set and interior-point methods is also provided in [BWB00, HNW08].

6.2.1 Interior-Point Methods

Interior-point methods (IPMs) are of polynomial complexity, which means that the effort for finding a solution to the optimization problem or for finding a certificate for the non-existence of a solution grows polynomially with the size of the optimization problem and the required accuracy of the solution [NN94]. In practice, IPMs generally require a small number of online iterations, where each iteration is computationally expensive, since the matrices for the Newton step computation have to be re-factored at every iteration. A significant reduction in the computational complexity can therefore be achieved by exploiting structure and sparsity and fast MPC methods are usually applied to the sparse formulation of the MPC problem. Interior-point methods are generally difficult to warm-start, since they can only benefit from a starting point close to the central path [YW02], whereas the initialization obtained from shifting the previous MPC solution is often located close to the constraints. It is therefore unclear how a good initial warm-start can be obtained. The shifted solution is often still applied to warm-start the IPM when solving an MPC problem, in order to avoid the computation of an initial feasible starting point in a so-called Phase I procedure.

The main effort in an interior-point method is the computation of the Newton step and all fast MPC approaches using IPMs described in the following focus on the efficient solution of this linear system, which, by column/row reordering, can be shown to be banded or have block structure, see e.g. [RWR98].

In early works, optimal control problems and their structure were investigated in an interior-point framework [DB89, Wri93, Ste94, LMF96, Wri97a]. By efficiently computing the Newton step using block factorization/Riccati recursions, the computational complexity is reduced from $\mathcal{O}(N^3(n_u + n_x)^3)$ growing cubic with the horizon length to $\mathcal{O}(N(n_u+n_x)^3)$ growing linearly with the horizon, where n_u and n_x denote the number of input and state variables. In [RWR98] this approach is tailored to the MPC application using an IPM based on Mehrotra's algorithm [Meh92]. The results are shown for a quadratic MPC formulation including a terminal constraint as well as soft constraints. This tailored method achieves computation times that are about 10 times faster than methods not exploiting the structure. The difficulty of warm-starting is discussed and the authors suggest to use an earlier iterate of the IPM from the previous time instant rather than the optimal one or a well-centered point as an initial solution. An extension of the QP method in [RWR98] to the robust output tracking problem is presented in [VBN02].

[GB98] investigate the use of interior-point methods for large-scale linear and nonlinear programming. A tailored IPM for robust optimal control problems resulting in robust QP formulations is developed in [Han00], which is based on the use of an iterative solver in conjunction with a Riccati-recursion invertible pre-conditioner to compute an approximate search direction. In [ART03] a primal-dual IPM is applied for the solution of large-scale conic quadratic optimization problems, which is based on Mehrotra's method and employs sparse linear algebra for computational efficiency. A method based on Riccati recursions for the efficient solution of robust control problems with a quadratic cost and bounded disturbances resulting in a block-bordered form is proposed in [GKR08].

[WB10] present an infeasible start primal barrier Newton method for fast MPC that is terminated after a fixed number of steps. A tailored solver was developed that exploits the sparse structure of the MPC problem by using block elimination and a block factorization of the block tri-diagonal Schur complement, requiring a complexity of $\mathcal{O}(N(n_u+n_x)^3)$ per iteration. A variant of the primal barrier interior-point method is proposed, where the barrier parameter is fixed to a predefined value instead of gradually decreasing it to zero. A standard warm-starting strategy employing the shifted solution from the previous time step is applied. The method is available as the *fast_mpc* package [WB08].

6.2.2 Active Set Methods

For active set methods (ASMs), there only exists an exponential worst case bound on the computational complexity, as was shown by the famous Klee and Minty example in the context of linear programming [KM82]. Active set methods usually require a larger number of online iterations, but each iteration is computationally less expensive than an interior-point iteration. The KKT matrix depends on the working set and only changes by adding or removing active constraints that are represented by rows and columns in the matrix. It is therefore not necessary to re-factor the matrix at each iteration, but instead cheap updates of the factorization can be performed. The active set method can either be applied to the sparse formulation and the banded structure be exploited in the updates, or to the dense formulation, which reduces the number of optimization variables and equality constraints and thereby also the computational effort. The benefit of either approach depends on the particular problem and is influenced e.g. by the number of constraints that are likely to be active. In contrast to interior-point methods, active set methods can take advantage of a feasible starting point with an active set that is close to the optimal active set. They are therefore considered well suited for warm-starting the MPC problem with the shifted solution computed at the last time instant, assuming that the active set does not change much for small changes of the parameters given by the state.

Similarly to the Newton step in an interior-point method, the main effort in an active set procedure is given by the solution of the KKT system for computing the step direction, which also results in a banded or block structured matrix for the sparse MPC formulation. Most fast MPC methods based on an active set approach are therefore concerned with the efficient computation of the step direction, where some are based on the dense and others on the sparse MPC formulation.

Very early results already show that the factorization of a structured Lagrangian for an optimal control problem in an active set framework yields a Riccati recursion [GJ84, ATW94] and can therefore be solved with a computational complexity that grows linearly with the horizon length N if the sparse problem formulation is used. In [BBBG02,BB06] a dual active set method is developed for solving large-scale structured QPs, where the active set updates are computed using a Schur complement strategy. An object oriented implementation called *QPSchur* is described in which the Hessian and constraint normals can be defined and implemented in a flexible manner allowing for the exploitation of problem-specific structure.

The method presented in [FBD08] is a warm-start based homotopy approach exploiting the piecewise affine structure of the optimal explicit solution. The KKT system obtained from the dense problem formulation is solved using a nullspace approach which requires $\mathcal{O}((Nn_u)^2)$ operations per active set change. The implementation of the method *qpOASES* can be found at [FBD09]. [CLK08] propose a quadratic programming solver for an input constrained MPC problem with terminal state constraints, where the matrix factorizations are replaced by recursions of state and co-state variables using Pontryagin's minimum principle, thereby reducing the number of optimization variables. An ASM using Riccati recursions is applied to solve the resulting equality constrained problems, leading to a complexity per iteration that grows linearly with the horizon length. The method can take advantage of warm-starts provided by the previous MPC solution. In [MD08] a non-feasible ASM for QPs is developed, reducing the number of iterations by updating the active set in blocks instead of using single updates. Active set changes on early stages of the control vector are considered with higher priority with the goal of increasing the quality of the applied control input in early steps of the method.

6.2.3 Other Methods

Apart from interior-point and active set methods, the following approaches have been proposed in the literature for fast computation of the MPC problem.

A dual gradient projection method for QPs is proposed in [AH08]. The method is similar to an active set method but allows for large changes of the working set at each iteration, thereby reducing the number of iterations in comparison with active set methods. The good warm-start properties of active-set methods are inherited and it is shown how all computations of major complexity can be tailored to the structure of MPC problems. In [KRS00] an augmented autonomous formulation of the system dynamics is proposed allowing one to impose a stabilizing constraint at the current time step rather than at the end of the horizon. Together with the use of ellipsoidal invariant sets this results in a QCQP that can be solved efficiently. The solution by means of a Newton-Raphson method is investigated in [KCR02], which together with [LKC10] provide further improvements and aim at removing the conservatism in the original approach. An algorithm that copes with time-varying computational resources by sequentially computing the components of the control input sequence is presented in [Gup09].

[RJM09] employ the fast gradient method with optimal first order convergence rates introduced in [Nes83] for the solution of MPC problems with input constraints. A priori bounds on the number of iterations to achieve a certain solution accuracy for warm- and cold-starting techniques are derived and ensure certified computation at high sampling rates. The approach presented in [PSS10] reduces the solution of the MPC problem to the solution of an unconstrained minimization of a convex quadratic spline, which is solved by applying a local Newton method without particularly exploiting the structure of the MPC problem. The method is compared to existent QP solvers using a series of benchmark problems.

6.2.4 Conclusions

All previously presented fast online MPC approaches show that the computation times for solving an MPC problem can be pushed into the range where online optimization becomes a reasonable alternative for the control of high-speed systems. There are, however, several limitations in the present approaches. The generally applied warm-start in MPC given by the input sequence computed at the previous state measurement is often an infeasible solution at the current time instant due to disturbances acting on the system. In addition, it is generally not possible to run the optimization procedure to optimality in a real-time setting but the optimization has to be terminated early when the time constraint is hit. Apart from [RJM09], where an a-priori bound on the runtime to achieve a certain solution accuracy is provided, all of the described fast online MPC methods sacrifice feasibility and/or stability guarantees in such a real-time environment. These limitations are addressed by the real-time methods presented in the following Chapters 7 and 8.

7 Real-time MPC Using a Combination of Explicit MPC and Online Optimization

7.1 Introduction

As discussed in the survey in Chapter 6, both the explicit and online MPC methods relying on optimality of the solution have limitations. The main disadvantage of a standard online approach is that it is in general only applicable for controlling slow dynamic processes. In the case of the explicit solution, the number of state-space regions over which the control law is defined, the so-called complexity of the partition, grows in the worst case exponentially due to the combinatorial nature of the problem [BBM02]. This has given rise to an increasing interest in the development of new methods to either improve online optimization (e.g. [Han00, KCR02, ART03, FBD08, WB10, CLK08, PSS10]) or to approximate explicit solutions (e.g. [RWR98, LR06, YW02, BF03, JG03, BF06, JBM07]). Depending on the particular problem properties and implementation restrictions, the user then has to decide for one of the two approaches.

This work aims at enlarging the possibilities to tradeoff solution properties through the combination of these two methods. It aims at an existing gap of problem sizes and types, which are either intractable for explicit MPC or the complexity of the explicit solution exceeds the storage capacity, but where an online MPC solution cannot meet the required online computation times. Specifically, ideas from approximation are combined with warm-start techniques. In this work we use a PWA approximation of the optimal control law that has been computed offline to warm-start the online optimization. The optimization executes a finite number of active set iterations before returning a feasible, suboptimal control action, which is then applied to the system. The goal is to choose a good tradeoff between the complexity of the PWA approximation and the number of active set iterations required in order to satisfy system constraints in terms of online computation, storage and performance. Conditions are derived which guarantee that the suboptimal solution is closed-loop stabilizing, feasible and has a bounded performance deterioration. The provided analysis has the important benefit that it is not based on the optimal parametric solution to the MPC problem, which may be prohibitively complex.

We also raise the question of an optimal combination of warm-start and online computational effort, with respect to certain requirements on the solution. Considering computation time and performance as two exemplifying requirements, this can be informally stated in the form of the following optimization problems:

1. Minimize online computation time while respecting a bound on the performance deterioration.
2. Maximize performance within the available online computation time.

The outline of this chapter is as follows: We first introduce some preliminary results and then present the main idea of using an offline approximation to warm-start an active set linear programming procedure in Section 7.3. An explicit representation of the proposed control law is derived in Section 7.4 and in Section 7.5 a preprocessing method is introduced that allows for an analysis of the properties of the control input that will be applied online. The question of an optimal combination between warm-start and online computation is discussed in Section 7.6 and finally, we illustrate the proposed method and ideas using numerical examples in Section 7.7.

7.2 Preliminaries

Consider the nominal MPC problem $\mathbb{P}_N(x)$ in (5.1) with a linear norm cost given by (5.3) satisfying Assumption 5.3. If the norm p is taken to be the $1-$ or the $\infty-$norm, we can write (5.1) as a parametric Linear Program (pLP) of the form:

$$\begin{aligned} z^*(x) = \underset{z}{\operatorname{argmin}} \quad & c^T z && \text{(7.1a)} \\ \text{subject to} \quad & G_{\mathcal{I}} z \le f_{\mathcal{I}} \ , && \text{(7.1b)} \\ & G_{\mathcal{E}} z = F_{\mathcal{E}} x \ , && \text{(7.1c)} \end{aligned}$$

where $z \in \mathbb{R}^{n_z}$ is a vector containing the sequence of control inputs $[u_0, \dots, u_{N-1}]$, states $[x_1, \dots, x_N]$ and appropriate slack variables introduced to rewrite the state and input penalties as a linear cost. The current state $x \in \mathbb{R}^{n_x}$ is the parameter, $G \in \mathbb{R}^{m \times n_z}$, $\mathcal{E} \subset \{1, \cdots, m\}$ and $\mathcal{I} = \{1, \cdots, m\} \setminus \mathcal{E}$. For a description of how the optimal control problem in (5.1) is transformed into the pLP (7.1), i.e. how the state and input constraints, the dynamics and the cost are converted into $G_{\mathcal{I}}, G_{\mathcal{E}}, f_{\mathcal{I}}, F_{\mathcal{E}}$ and c, see e.g. [Bor03]. (Note that for simplicity we use the same indexing for f and F as for G although we distinguish the vector f from the matrix F in order to account for the different dimension).

Definition 7.1. Let z be the vector of decision variables in (7.1). We define $\pi : \mathbb{R}^{n_z} \to \mathbb{R}^{n_u}$ to be a linear mapping that returns the first control input u_0 contained as a component in z.

By solving the pLP (7.1) the optimal solution $z^*(x)$ can be computed for each feasible value of the state $x \in \mathcal{X}_N$. The implicit optimal MPC control law is then given in a receding horizon fashion by $\kappa(x) = \pi(z^*(x))$.

Remark 7.2. Note that in this work the $1-$ or $\infty-$norm is chosen to penalize the states and inputs in the cost function (5.1a) instead of the commonly used 2-norm, as the resulting pLP (7.1) allows for an exact analysis of the control law obtained by the proposed procedure, as will be shown in Section 7.4.

Theorem 7.3 (Solution to the MPC problem, [BBM02, MRRS00]) *Consider Problem $\mathbb{P}_N(x)$ in* (5.1) *fulfilling Assumption 5.3. The optimal value function $V_N^*(x)$ is a continuous PWA Lyapunov function of the state x defined over the feasible set $\mathcal{X}_N$. There exists an optimizer function $z^*(x)$ that is a continuous PWA function of x defined over $\mathcal{X}_N$. The closed-loop system $x(k+1) = Ax(k) + B\kappa(x(k))$ is asymptotically stable with region of attraction $\mathcal{X}_N$.*

Remark 7.4. Note that even in the case of dual degeneracy, there always exists a polyhedral partition such that the optimizer $z^*(x)$ in (7.1) and thereby the optimal control law $\kappa(x)$ is unique and continuous in $\mathcal{X}_N$ [BBM02]. Continuity of $\kappa(x)$ is, however, not a required assumption for the proposed method.

Approximation of the MPC Problem

We first define an approximation of the MPC problem (5.1) and some useful properties that will be used in Section 7.5 in order to give guarantees on the control law proposed in this work. Let $z^*(x)$ be the optimizer of the optimal control problem (7.1) and $V_N^*(x) = c^T z^*(x)$ be the corresponding optimal cost and a Lyapunov function for the closed-loop system $x(k+1) = Ax(k) + B\kappa(x(k))$.

Definition 7.5 (Feasibility). A function $\tilde{z} : \mathbb{R}^{n_x} \to \mathbb{R}^{n_z}$ is called feasible for (7.1) in $\mathcal{X}_N$, if it satisfies the constraints in (7.1b) and (7.1c) for all $x \in \mathcal{X}_N$.

The *approximation error* is defined as follows:

Definition 7.6 (Approximation error). Let $\tilde{z}(x)$ be a suboptimal solution to (7.1). A function $\tilde{\kappa}(x) = \pi(\tilde{z}(x))$ is called an approximate control law for (5.1) if $\tilde{z}(x)$ is feasible for (7.1) in $\mathcal{X}_N$. The approximate control law $\tilde{\kappa}(x)$ is an ϵ-approximation if for all $x \in \mathcal{X}_N$ the condition $\tilde{V}(x) - V_N^*(x) \leq \epsilon$ is satisfied, where $\tilde{V}(x) \triangleq c^T \tilde{z}(x)$ and ϵ is the smallest value satisfying this inequality.

Remark 7.7. If $\tilde{z}(x)$ is feasible for (7.1) in $\mathcal{X}_N$, then $\tilde{\kappa}(x) = \pi(\tilde{z}(x)) \in \mathbb{U}$ and $x(k{+}1) = Ax(k) + B\tilde{\kappa}(x(k)) \in \mathcal{X}_N$ for all $x \in \mathcal{X}_N$.

Remark 7.8. The approximation error ϵ denotes the worst accuracy over the feasible set $\mathcal{X}_N$ and is hence uniform while the accuracy of the approximate cost $\tilde{V}(x)$ is varying with x.

A standard condition to test if an approximate solution is *stabilizing* is given by the following theorem:

Theorem 7.9 (Stability under $\tilde{\kappa}(x)$) *Let $\tilde{\kappa}(x)$ be an approximate control law of (5.1). The optimal value function $V_N^*(x)$ is a Lyapunov function for the closed-loop system $x(k+1) = Ax(k) + B\tilde{\kappa}(x(k))$ if*

$$\tilde{V}(x) - V_N^*(x) \leq \rho l(x, \tilde{\kappa}(x)), \text{ for all } x \in \mathcal{X}_N \text{ and some } \rho \in [0, 1) \ , \tag{7.2}$$

where $l(\cdot, \cdot)$ is the stage cost in (5.1a), and the closed-loop system $x(k+1) = Ax(k) + B\tilde{\kappa}(x(k))$ is asymptotically stable with region of attraction $\mathcal{X}_N$.

Proof. The result can be obtained from the proof of Theorem 1 in [JM09]. ∎

7.3 Proposed Control Law

In order to overcome the limitations of the offline and online methods mentioned in the introduction, several authors recently proposed new approaches to speed up online optimization or to reduce the complexity of explicit solutions by means of approximation. The authors in [RWR98, WB10] for example utilize new developments in interior-point methods and show how these can be applied to efficiently solve the optimal control problem. Another paradigm that is frequently applied to improve online optimization is warm-starting (see e.g. [YW02, FBD08]). In explicit MPC, approximation methods have been proposed that either modify the original MPC problem (5.1), retrieve a suboptimal solution or postprocess the computed optimal solution, with the goal of reducing the complexity of the explicit controller, see e.g. [BF03, BF06, JBM07, JG03] and Section 6.1 for an overview of methods in the literature. All of the cited explicit approximation methods provide an approximate control law and allow verification of closed-loop stability by means of Theorem 7.9 for some minimally required complexity. Almost all currently available online MPC methods, however, lack the possibility of giving guarantees on the suboptimal solution, e.g. closed-loop stability or feasibility, in a real-time setting.

The strategy proposed in this chapter combines the idea of offline approximation with warm-start techniques from online optimization with the goal of providing hard real-time, stability and performance guarantees. Warm-start techniques aim at identifying advanced starting points for the optimization in order to reduce the number of iterations required to reach the optimum. They often try to make use of the information gained during the solution of one problem to solve the next one in a sequence of closely related problems. When solving MPC problems in a receding horizon fashion an LP is computed for every measured state. However in practice, the optimal control sequence from a previous measurement might be an infeasible solution to (5.1) at the current time instance, due to disturbances acting on the system. We therefore propose a warm-start strategy that utilizes a PWA approximation of (7.1) to provide a good and feasible starting point. The pre-knowledge of the initial solution for all feasible values of the state x allows us to analyze the solution obtained by the online optimization.

The following two parameters are used to classify the warm-start solution: the complexity N_P (number of regions) and its approximation error ϵ, given by Definition 7.6.

Definition 7.10 (Warm-start Solution). A function $\nu(x, N_P)$ is called a warm-start solution of (7.1) if $\pi(\nu(x, N_P))$ is a feasible, PWA approximate control law of (5.1) defined over N_P polytopic regions.

Lemma 7.11 *There exists a function $\nu(x, N_{P_{opt}})$ of finite complexity $N_{P_{opt}} \in \mathbb{N}$, such that $z^*(x) = \nu(x, N_{P_{opt}})$ for all $x \in \mathcal{X}_N$, where $z^*(x)$ is the optimal solution to* (7.1).

Proof. Result follows directly from the fact that the approximation is PWA and Theorem 7.3.

By means of the parameter N_P requirements can be set on the complexity of the warm-start solution $\nu(\bullet, N_P)$ that is computed and stored in an offline preprocessing step.

Remark 7.12. As was shown in [JBM07] there exists an analytical relation between the approximation error ϵ and the complexity N_P of a PWA offline approximation. Requirements on the approximation error can therefore also be imposed using the parameter N_P.

In the online control procedure, the warm-start solution is evaluated for the measured state and used to initialize the online optimization. A standard active set method (ASM) is applied to compute the control action since it allows us to take advantage of a feasible starting point that is not necessarily located at a vertex. As described in Section 3.1, active set methods generate a sequence of feasible iterates that converge to the optimal solution. At each iterate z, the active set is given by $\mathcal{A}(z)$ defined in Definition 3.3. In an active set iteration, a subset of the active set is chosen as the working set $W \subseteq \mathcal{A}(z)$ using standard heuristics. From the current iterate z, the maximal step in the search direction is then computed, which is the direction that minimizes the objective in (7.1) while keeping the constraints in W active.

Assumption 7.13 (Non-degeneracy). It is assumed that the active set is non-degenerate, i.e. the active constraints are linearly independent.

Note that it is possible to extend the approach to degenerate cases by using one of the standard approaches for anti-cycling (e.g. [Wol63, Bla77, GMSW89]) or lexicographic perturbation (e.g. [Mur83]) in active set methods.

Whereas in standard active set methods iterations are performed until the optimality conditions are met, the online optimization procedure is stopped early after exactly K active set iterations and the current suboptimal control input is applied to the system.

Definition 7.14 (Warm-start optimization). Let $\tilde{z}(x)$ be a feasible solution of (7.1) for the parameter x. We define $\sigma(\tilde{z}(x), K)$ to be the decision variable of (7.1) after K iterations of the linear programming active set method starting from the point $\tilde{z}(x)$.

Definition 7.15 (Proposed control law). Let $\nu(\cdot, N_P)$ be a warm-start solution to (7.1) and $\sigma(\cdot, K)$ be as defined in 7.14. The proposed control law is

$$\kappa_{\text{on}}(x) = \pi(\sigma(\nu(x, N_P), K)), \text{ for } x \in \mathcal{X}_N \ . \tag{7.3}$$

Lemma 7.16 (Properties of $\boldsymbol{\sigma(\cdot, \cdot)}$) *The proposed control law* (7.3) *is feasible for all $x \in \mathcal{X}_N$, and for each N_P there exists a finite $K_{opt} \in \mathbb{N}$, such that $\kappa(x) = \pi(\sigma(\nu(x, N_P), K_{opt}))$.*

Proof. Feasibility is ensured by the procedure of the ASM and the fact that $\nu(x, N_P)$ is feasible for all $x \in \mathcal{X}_N$. The existence of a finite K_{opt} is guaranteed by the convergence of the ASM in finite time [Fle87, NW06]. ■

The warm-start linear programming procedure for a fixed complexity N_P and number of iterations K is summarized in Algorithm 1. In Section 7.5 an offline analysis is introduced providing guarantees for the proposed control law $\kappa_{\text{on}}(x)$ in (7.3) to be closed-loop stabilizing, feasible and to have a bounded performance deterioration compared to the optimal solution.

Algorithm 1 Warm-start linear programming procedure

```
Input: Warm-start solution $\nu(\bullet, N_P)$ and current measured state $x_{\text{meas}}$
Output: Approximate control input $\kappa_{\text{on}}(x_{\text{meas}})$
run point location algorithm: $\tilde{z} = \nu(x_{\text{meas}}, N_P)$
                                    ▷ [BBBM08, TJB03b, JGR06, WJM07]
for $k = 1, \cdots, K$ do
    perform an active set iteration              ▷ [GMW82, Fle87]
    update iterate $\tilde{z}$
end for
$\kappa_{\text{on}}(x_{\text{meas}}) = \pi(\tilde{z})$
```

The above described procedure of using an approximation to warm-start an online optimization offers the possibility to decide on the complexity and approximation error of the warm-start solution $\nu(\bullet, \bullet)$. A tradeoff can be made between the degree of approximation realized by the warm-start and the effort expended in online optimization. The goal is to identify a good if not optimal combination that achieves the best properties of the online control input applied to the system for given requirements on the approximation error and/or limitations on the online computation time or storage.

The proposed procedure and algorithms are detailed in the following sections.

7.4 Parametric Calculation of the Online Control Law

Our goal is to give guarantees on the proposed suboptimal control law in (7.3). Apart from feasibility, which is guaranteed by Lemma 7.16, we want to ensure stability and a certain bound on the approximation error. In order to analyze the solution properties, we need an explicit representation of the approximate solution $\sigma(\bullet, \bullet)$ for the entire feasible set $\mathcal{X}_N$. We will show that starting from the warm-start solution, the iterative path taken by the active set method is a function of the state x, defined over a polyhedral subdivision of $\mathcal{X}_N$.

Remark 7.17 (Offline Complexity). Note that the complexity (number of regions) of this subdivision does not affect the actual optimization carried out online, since the parametric solution is only used for offline analysis.

The operations performed during an active set iteration can be formulated as functions of the parameter x. Let $z^k(x)$ be the iterate and W the working set at the k'th iteration step. A search direction Δz from $z^k(x)$ is computed that maintains activity of the constraints in W. In this work, we use the projection of the gradient c onto the subspace defined by the constraints in the working set, which corresponds to:

$$\Delta z = \underset{\Delta z}{\operatorname{argmin}} \left\{ c^T \Delta z \;\middle|\; G_W \Delta z = 0, \frac{1}{2} \Delta z^T \Delta z = \delta \right\} \ , \tag{7.4}$$

where $\delta \in \mathbb{R}_{\geq 0}$ is a scaling parameter.

The steps performed during one active set iteration are outlined in the following (see e.g. Section 3.1 or [GMW82, Fle87]):

1. Compute step direction by solving the KKT conditions of (7.4)

$$\begin{bmatrix} I & G_W^T \\ G_W & 0 \end{bmatrix} \begin{bmatrix} \Delta z \\ \lambda \end{bmatrix} = \begin{bmatrix} -c \\ 0 \end{bmatrix} \ , \tag{7.5}$$

   where I denotes the identity matrix of size $n_z \times n_z$ and λ are the Lagrange multipliers.
   If $\Delta z = 0$ and $\lambda \geq 0$: Stop with optimal solution $z^*(x) = z^k(x)$.
   If $\Delta z = 0$ and at least one $\lambda_i < 0$: Remove constraint a from W, i.e. $W = W \backslash \{a\}$, where $a = \operatorname{argmin}_j \{\lambda_j \,|\, j \in W \cap \mathcal{I}\}$. Recompute (7.5).

2. Compute maximum step length for which all constraints are satisfied:

$$\tau(x) = \min_{i \in \mathcal{I} \backslash W} \left\{ \frac{f_i - G_i z^k(x)}{G_i \Delta z} \;\middle|\; G_i \Delta z > 0 \right\} \ . \tag{7.6}$$

3. Update W: $W = W \cup e$, where the new active constraint e is the optimizer of (7.6).

4. Update iterate:

$$z^{k+1}(x) = z^k(x) + \tau(x) \Delta z \ . \tag{7.7}$$

Remark 7.18 (ASM). A similar parametric formulation can also be obtained using a different search direction or any of the other standard implementations found in the literature; the simplex-type algorithm in [BR85] for instance. For simplicity, the strategy of adding or removing at most one inequality constraint in each iteration to or from the working set is considered. Note that $\mathcal{E} \subseteq W$, since the equality constraints $\mathcal{E}$ always have to remain in the working set.

Theorem 7.19 *At every iteration $k \in \{1, \cdots, K\}$, the step length $\tau(x)$ in (7.6) and the current iterate $z^k(x)$ are PWA functions of x defined over a polyhedral partition $\mathcal{P}^{N_k}$ of the feasible set $\mathcal{X}_N$.*

Proof. Assume that the statement is true for k and that $z^k(x) \triangleq C^j x + D^j$ if $x \in P_j \; \forall \; P_j \in \mathcal{P}^{N_k}$. For one region $\mathcal{P}_j \in \mathcal{P}^{N_k}$ the line search (7.6) determining the next constraint to become active is given by the pLP

$$\tau(x) = \min_i \{\alpha_i(x)\}, \text{ with } \alpha_i(x) = \frac{-G_i C^j}{G_i \Delta z} x + \frac{f_i - G_i D^j}{G_i \Delta z}, \; i \in \mathcal{I}_{cand}$$

and $\mathcal{I}_{cand} = \{i \in \mathcal{I} \backslash W \mid G_i \Delta z > 0\}$. This shows that $\tau(x)$ is a PWA function of x, since the optimal cost of a pLP is PWA [GN72, BBM02]. With $\tau(x)$ and $z^k(x)$ being PWA, the $k+1$'th iterate (7.7) is as well a PWA function of x. Since the initial solution is $z^0(x) = \nu(x, N_P)$, the statement is true for $k = 0$ and hence for all $k \in \{1, \cdots, K\}$.■

Corollary 7.20 *The proposed control law $\kappa_{on}(x)$ is a PWA function of x defined over a polyhedral partition $\mathcal{P}^{N_K}$ of the feasible set $\mathcal{X}_N$.*

With $z^k(x)$ being a PWA function, equation (7.6) results in a parametric LP. The approximate control law at iteration K is obtained by solving (7.5) and (7.6) iteratively for all $k \in \{1, \cdots, K\}$. With each iteration, the parametric solution of (7.6) causes a further refinement of the polyhedral partition $\mathcal{P}^{N_k}$.

Remark 7.21. Note that the computation of the PWA function $\tau(x)$ in (7.6) can be reduced to redundancy elimination of the halfspaces $h(x) \geq \frac{f_i - G_i z^k(x)}{G_i \Delta z}$. All halfspaces $j \in \mathcal{I} \backslash W$ with

$$G_j \Delta z > 0, \frac{f_j - G_j z^k(x)}{G_j \Delta z} \geq \frac{f_i - G_i z^k(x)}{G_i \Delta z} \; \forall i \neq j, i \in \mathcal{I} \backslash W, \; G_i \Delta z > 0$$

are redundant and can be disregarded, the irredundant halfspaces form the PWA parametric solution of (7.6). Redundancy elimination problems can be easily solved by computing one linear program per constraint.

Theorem 7.19 enables us to compute an explicit PWA representation of the approximate control law $\sigma(\nu(x, N_P), K)$ for a fixed complexity N_P and number of iterations K using Algorithm 2.

Remark 7.22. The parametric calculation of the proposed control law in (7.3) using Algorithm 2 requires the iterative solution of a parametric program. Although the computational complexity depends on the properties of the considered control problem it grows significantly with the problem size. This currently limits the execution of the offline analysis in practice, where in our experience the biggest problem to be computed was for a system with 8 states. Note that the warm-start linear programming procedure is however not affected by this complexity and could still be applied without the rigorous guarantees provided by the analysis.

Algorithm 2 Offline Analysis

```
Input: Warm-start solution $\nu(x, N_P) = \nu^j(x)$ if $x \in P_j \ \forall\ P_j \in \mathcal{P}^{N_P}$
Output: Explicit representation of the proposed control law $\kappa_{\text{on}}(x)$
initialize stack $S = \emptyset$
for all $P_j \in \mathcal{P}^{N_P}$ do push $(\nu^j(x), P_j)$ onto $S$
end for
for all $k \in \{1, \cdots, K\}$ do                                ▷ [$K$ active set iterations]
    initialize stack $\hat{S} = \emptyset$
    while $S \neq \emptyset$ do                                    ▷ [subdivide each region]
        pop $(z_P(x), P)$ from $S$
        compute $\Delta z$                                         ▷ (7.5)
        compute $a^j, b^j$ for $z_P(x),\ x \in P$,                 ▷ (7.6)
        such that $\tau(x) = a^j x + b^j$ if $x \in R_j \ \forall\ R_j \in \mathcal{R}$
                                                  ▷ [$\mathcal{R}$ is a polyhedral partition of $P$]
        for all $R_j \in \mathcal{R}$ do
            push $(z_P(x) + (a^j x + b^j)\Delta z, R_j)$ onto $\hat{S}$
        end for
    end while
    $z^k(x) \triangleq z^{k,j}(x)$ if $x \in P_j \ \forall\ (z^{k,j}(x), P_j) \in \hat{S}$
    $\mathcal{P}^{N_k} \triangleq \{P_j\}_{j=1}^{N_k}$             ▷ [$P_j$ form polyhedral partition $\mathcal{P}^{N_k}$]
    $S = \hat{S}$
end for
$\kappa_{\text{on}}(x) = \pi(z^K(x))$
```

7.5 Analysis of the Proposed Control Law According to Stability, Suboptimality, Storage Space and Computation Time

We introduce a preprocessing analysis that investigates the following properties of the approximate control law $\kappa_{\text{on}}(x)$ in (7.3): stability, approximation error, storage space and online computation time.

Lemma 7.23 (Approximation error of $\boldsymbol{\kappa_{\text{on}}(x)}$) *If $\sigma(\cdot, K) \triangleq C^j x + D^j$ if$x \in P_j \,\forall\ P_j \in \mathcal{P}^{N_K}$ and $\kappa_{on}(x) = \pi(\sigma(\cdot, K))$ is the proposed control law in (7.3) at iteration K, then the approximation error defined in Definition 7.6 is given by*

$$\epsilon_K = \max_{j \in \{1,\ldots,N_K\}} d_j \ , \ \textit{with} \tag{7.8}$$

$$
\begin{aligned}
d_j = \quad & \max_{z,x} \; c^T(C^j x + D^j) - c^T z \\
& s.t. \quad G_{\mathcal{I}} z \le f_{\mathcal{I}},\; G_{\mathcal{E}} z = F_{\mathcal{E}} x,\; x \in P_j \;\; .
\end{aligned} \tag{7.9}
$$

Proof. We first prove that equation (7.9) computes the largest distance between the approximate and the optimal cost. We take $x = \bar{x}$ fixed and show by contradiction that equation (7.9) computes the distance to the optimal cost at $\bar{x}$. Assume $z^*(\bar{x})$ is the optimal solution to (7.1) and $z^o(\bar{x})$ the optimal solution to (7.9) for $x = \bar{x}$. If $z^o(\bar{x})$ is not the optimal solution to (7.1), i.e. $c^T z^o(\bar{x}) \ge c^T z^*(\bar{x})$, then it follows that $c^T(C^j\bar{x} + D^j) - c^T z^o(\bar{x}) \le c^T(C^j\bar{x} + D^j) - c^T z^*(\bar{x})$. Therefore $z^o(\bar{x})$ cannot be the optimal solution to (7.9) at $\bar{x}$, which proves that $z^o(\bar{x})$ also has to be the optimal solution to (7.1). Now letting x vary and simultaneously taking the maximum over all $x \in P_j$ gives the worst case distance in P_j. Finally, the biggest error over all $x \in \mathcal{X}_N$ and hence over all the regions in $\mathcal{P}^{N_K}$ is the smallest ϵ that fulfills the condition in Definition 7.6 for all $x \in \mathcal{X}_N$. ■

Stability can be easily tested using the conditions of Theorem 7.9:

Theorem 7.24 (Stability of $\boldsymbol{\kappa_{\mathrm{on}}(x)}$) *Let $\sigma(\bullet, K) \triangleq C^j x + D^j$ if $x \in P_j \; \forall \; P_j \in \mathcal{P}^{N_K}$ and $\kappa_{on}(x) = \pi(\sigma(\bullet, K))$ be the proposed control law in (7.3) at iteration K. The closed-loop system $x(k+1) = Ax(k) + B\kappa_{on}(x(k))$ is asymptotically stable with region of attraction $\mathcal{X}_N$, if for some $\rho \in [0, 1)$*

$$
\begin{aligned}
& \max_{j \in \{1,\dots,N_K\}} s_j \le 0 \;\; , \; \textit{with} \\
s_j = \; & \max_{z,x} \; c^T(C^j x + D^j) - c^T z - \rho l(x, \kappa_{on}(x)) \\
& s.t. \quad G_{\mathcal{I}} z \le f_{\mathcal{I}},\; G_{\mathcal{E}} z = F_{\mathcal{E}} x,\; x \in P_j \;\; .
\end{aligned} \tag{7.10}
$$

Proof. Condition (7.10) follows directly from Theorem 7.9. If the difference between the approximate cost $\tilde{V}(x) \triangleq c^T(C^j x + D^j)$ and the optimal cost $V_N^*(x)$ is less than $l(x, \kappa_{\mathrm{on}}(x))$ for all $x \in P_j$ and for all the regions P_j in $\mathcal{P}^{N_K}$, then the condition in Theorem 7.9 is fulfilled for all $x \in \mathcal{X}_N$. ■

Whereas condition (7.10) is sufficient to prove stability of the proposed control law at iteration K, it does not guarantee stability of the control laws at iterations $k \ge K$. However, by using a more conservative condition instead of (7.10) we can modify Theorem 7.24 in order to ensure stability not only at iteration K but also at all subsequent active set iterations.

Corollary 7.25 *Let $\sigma(\bullet, K) \triangleq C^j x + D^j$ if $x \in P_j \; \forall \; P_j \in \mathcal{P}^{N_K}$ and $\kappa_{on}(x) = \pi(\sigma(\bullet, K))$ be the proposed control law in (7.3) at iteration K. The closed-loop system $x(k+1) = Ax(k) + B\kappa_{on}(x(k))$ is asymptotically stable with region of attraction*

$\mathcal{X}_N$ *for all control laws* $\kappa_{on}(x) = \pi(\sigma(\bullet, k))$ *for* $k \geq K$*, if for some* $\rho \in [0, 1)$

$$\begin{aligned} \max_{j \in \{1,\dots,N_K\}} & s_j \leq 0 \ , \ \textit{with} \\ s_j = \max_{z,x} \ & c^T(C^j x + D^j) - c^T z - \rho l_{min}(x) \\ \textit{s.t.} \quad & G_{\mathcal{I}} z \leq f_{\mathcal{I}}, \ G_{\mathcal{E}} z = F_{\mathcal{E}} x, \ x \in P_j \ , \end{aligned} \tag{7.11}$$

where $l_{min}(x) = \min_u \{l(x, u) \mid u \in \mathbb{U}\}$ *.*

Proof. Follows from Theorem 7.24 and the fact that $l_{\min}(x) \leq l(x, \kappa_{\text{on}}(x))$. ■

Remark 7.26. Note that if the origin is contained in $\mathbb{U}$, then $l_{\min}(x) = \|Qx\|_p$.

Remark 7.27 (Stability/Performance test). Note that equations (7.10) and (7.11) compute the maximum distance to the optimal cost plus the stage cost without computation of the parametric optimal solution to problem (7.1), which can be easily shown following the first part of the proof of Lemma 7.23. Using Theorem 7.24 or Corollary 7.25 stability of the proposed control law (7.3) can hence be tested without the need to compute the optimal and potentially complex parametric solution to problem (7.1).

Storage space is determined by the complexity (number of regions) N_P of the warm-start solution since only the warm-start has to be stored.

Online computation time will be estimated in terms of floating point operations (flops) for the calculations that have to be performed online. First, the region of the current state is identified using a point location algorithm (e.g. [BBBM08, TJB03b, JGR06, WJM07]), then the corresponding affine control law is evaluated and finally K online iterations are executed.

Remark 7.28 (Sparsity of the MPC problem). In the case of the MPC problem (5.1), the matrices $G_{\mathcal{I}}$ and $G_{\mathcal{E}}$ in (7.1) have a special structure, resulting from the particular problem setup. The matrices are extremely sparse and by reordering can be shown to be in fact block diagonal or banded. We can exploit the banded structure of the matrices to solve equations (7.5) and (7.6), achieving significant computational savings [Wri97a].

Theorem 7.29 (Flop count) *If the input and state dimensions are* n_u *and* n_x*, respectively, the number of constraints on each state-input pair is* m_c*,* N *is the horizon in* (5.1a) *and the number of slack variables introduced for each state-input pair to write problem* (5.1) *as pLP* (7.1) *is* n_s*, then the number of flops to calculate the control action* $\kappa_{on}(x_{meas})$ *for a measured state* x_{meas} *can be bounded by:*

$$f_{on} = N_P f_{ws} + f_{eval} + K f_{ASM} \ , \tag{7.12}$$

$$\begin{aligned} \textit{where}\quad f_{eval} &= 2n_x(Nn_x + n_u + n_s)\ , \\ f_{ASM} &= N(n_x(82n_x + 26m_c + 68n_s + 11) \\ &\quad + n_u(40n_u + 116n_x + 56n_s + 20m_c + 9) \\ &\quad + n_s(12m_c + 3) - 3m_c) \end{aligned}$$

f_{ws} denotes the flop number for point location per region, f_{eval} the flops for evaluation of an affine warm-start solution and f_{ASM} the flops per active set iteration.

Proof. The flop counts for active set iterations use the fact that the matrices in (7.1) have banded structure. An LU factorization and LU updates as described in [Wri97a] and [FT72] is considered, where L and U have half-bandwidth $5n_x + 4n_u + m_c - 1$. The worst case is taken in form of the maximal number of active constraints. The flop counts for calculation of the search direction, step length and the next iterate follow directly from the equations (7.5), (7.6) and (7.7). ■

Remark 7.30. The number of flops for point location f_{ws} in Theorem 7.29 depends on the particular point location algorithm used in Algorithm 1, e.g. the one described in [BBBM08] requires $2n_x$ flops.

The worst-case estimates for the properties of the proposed control law $\sigma(\cdot, K)$ in terms of stability, approximation error, storage space and online computation time can be calculated. For fixed parameters N_P and K this allows us to give guarantees on the properties of the control law that is applied to the system.

Remark 7.31. In the case that one only wants to prove stability for a fixed K, the analysis can be stopped early after stability is guaranteed for a certain iteration $K_s \leq K$ using Corollary 7.25. In the actual warm-start procedure, the online optimization can then be carried out to K iterations while still guaranteeing stability of the proposed control law.

Remark 7.32 (Computational effort). Note that the computational effort for Algorithm 2 can be reduced by applying ideas from tree search. A node in the tree represents a single region and each depth level corresponds to an active set iteration. Using depth-first-search, for instance, an error bound for the K'th active set iteration can be derived without calculating the full parametric solution. Since in this case there are regions for which we infer the solution at the K'th level from an earlier iteration, we have to employ the more conservative stability test using Corollary 7.25.

7.6 Optimization over the Parameters

In this section we will now try to optimize over the parameters that determine the applied control input: the complexity N_P of the warm-start solution and the number of iterations K. The choice of the warm-start solution $\nu(\bullet, N_P)$ determines the computational effort for point location f_{ws} in (7.12) on the one hand and the quality of the warm-start on the other and therefore the number of active set iterations required in order to provide stability and a certain performance. This offers the possibility to trade off the amount of online computation time spent on the warm-start with that spent on online optimization. The challenge is to identify an optimal combination of explicit approximation and online optimization that achieves the best properties of the applied control input.

We consider the problem of optimizing the online time to compute a control law that guarantees stability, a certain performance bound ϵ_{max} and a limit on the storage space $N_{P,max}$. Computation time is again measured in the form of flops (7.12), resulting in the following optimization problem

$$F_{min} = \min_{N_P,K} N_P f_{\text{ws}} + K f_{\text{ASM}} \tag{7.13a}$$

$$\text{subject to} \quad \epsilon_K \leq \epsilon_{max} \ , \tag{7.13b}$$

$$N_P \leq N_{P,max} \ , \tag{7.13c}$$

$$(7.10)/(7.11) \ . \tag{7.13d}$$

While the exact solution of this optimization problem is not possible with currently available methods, it can be used as a quality measure to evaluate different combination possibilities. We demonstrate in the following how the identification of a good combination can be pursued for a particular offline approximation method.

There are several approximation methods that can be used to create a PWA warm-start solution (e.g. [JG03,BF06,LR06]). In this work the method introduced in [JBM07] was chosen that is based on the beneath/beyond (B/B) algorithm, a common approach for convex hull calculation [BDH96,Grü61]. An approximation $\tilde{V}(x)$ of $V_N^*(x)$ in (5.1a) is constructed by computing the convex hull of a subset of vertices of the epigraph of $V_N^*(x)$. The approximation can be iteratively improved by adding one vertex at a time and updating the convex hull. When all vertices of the polytope are included, the optimal solution of (5.1) is reached. The approximate control law is obtained by interpolating between the optimal control inputs at the vertices. The main advantage of the beneath/beyond method is that it is an incremental approach, allowing one to set requirements on either the complexity or the error of the approximation $\tilde{V}(x)$. In addition, it is based on an implicit rather than on an explicit representation of the optimal solution and is hence not dependent on the computability of the optimal parametric solution to pLP (7.1).

Theorem 7.33 (B/B warm-start solution [JBM07]) *Given a parameter $N_P \in \mathbb{N}$ the B/B method returns a feasible PWA approximation $\nu(\bullet, N_P)$ of (7.1).*

Remark 7.34. The approximate control law generated by the B/B method is not necessarily defined over the entire feasible set. For small dimensional problems the B/B algorithm can be initialized with all the vertices of the boundary of $\mathcal{X}_N$, which can be computed by projection of the feasible set defined by the constraints in (7.1) onto the x-space, in order to provide a feasible control law for all $x \in \mathcal{X}_N$. For higher dimensional problems the method can be extended so that one can reduce the complexity of the approximation by considering only a subset of $\mathcal{X}_N$ [JBM07].

The error of a B/B approximation is related to its complexity by the following lemma.

Lemma 7.35 (Complexity/Approx. Error, [SW81, BI75]) *Let $\nu(\cdot, N_P)$ be an approximation to* (7.1) *generated by the B/B method, ϵ_{BB} the approximation error as defined in Definition 7.6 and N_P its complexity. For every ϵ_{BB} there exists an $N_{BB} \in \mathbb{N}$ such that the approximation error of $\nu(\cdot, N_{BB})$ is less than ϵ_{BB}.*

Remark 7.36. Note that whereas the approximation error of a B/B approximation is monotonically decreasing with every B/B improvement, the complexity might not be monotonic (see [JBM07]).

Using a B/B approach, problem (7.13) is a function of only the complexity N_P of the warm-start solution. This follows from the fact that for each complexity N_P of the B/B warm-start there exists exactly one minimal number of iterations K to achieve a certain approximation error ϵ_{max}. For each $N_P \leq N_{P,max}$ the smallest number of optimization steps K that satisfies the constraint on the approximation error can be computed using Algorithm 2. This is the case not only for the B/B method, but for all offline approximation methods for which there exists a one-to-one relationship between the approximation error ϵ and the complexity N_P of the PWA warm-start solution. Since the calculation of B/B approximations and the solution of Algorithm 2 can be computationally expensive, we propose to solve a subproblem of (7.13), in order to identify a good combination instead of the optimal one. We use a subset of values for $N_P \leq N_{P,\max}$ and compute the minimal K for each of them. The best combination of explicit approximation and online optimization is represented by the solution with the minimum cost value in (7.13a). The samples can then be iteratively refined to improve the obtained result and approach the optimal solution to problem (7.13).

Remark 7.37. The problem of identifying the optimal combination of explicit approximation and online optimization to minimize the approximation error subject to constraints on the online computation time and the storage space can be approached following the same procedure described for problem (7.13).

7.7 Numerical Examples

In this section we will illustrate the proposed warm-start linear programming procedure and demonstrate its advantages using four numerical examples. The point location algorithm in [BBBM08] was used for Step 1 of Algorithm 1.

7.7.1 Illustrative Example

We first exemplify the main procedure for a small 1D randomly generated toy system:

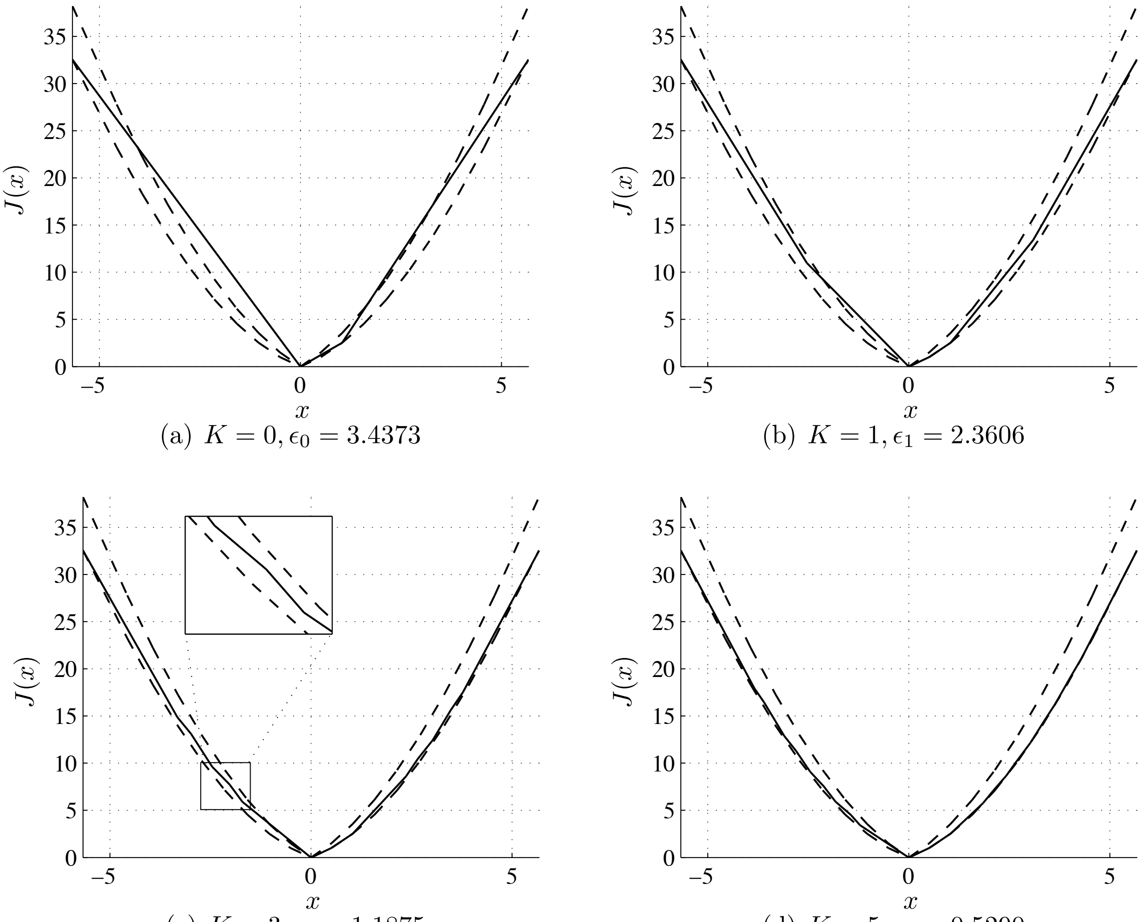
$$x(k+1) = -0.969x(k) + 0.4940u(k) \ , \tag{7.14}$$

with a prediction horizon $N = 10$ and the constraints $\|u\|_\infty \leq 1$, $\|x\|_\infty \leq 5$ on the input and state respectively. The norm p for the stage cost is taken as the 1-norm and the weights are taken as $Q = 1$ and $R = 1$.

A warm-start solution with $N_P = 3$ regions is computed by means of the B/B approximation algorithm [JBM07]. We investigate the proposed control law after each active set iteration until the optimal solution is reached. The cost of the warm-start B/B solution and the proposed control laws at $K = 1, 3$ and 5 are shown in Fig. 7.1, as well as the stability bound given by Theorem 7.9. Once the approximate cost lies sxtrictly within this bound, stability of the proposed control law is guaranteed. It can be seen in Fig. 7.1(d) that for this example stability is achieved after $K = 5$ iterations. The procedure converges to the optimal solution after $K = 8$ iterations. Whereas the warm-start solution has an approximation error of $\epsilon_{BB} = 3.4373$, the stable control law at $K = 5$ with an approximation error of $\epsilon_5 = 0.5290$ is already very close to optimality. In Fig. 7.1(c) we can observe another characteristic of the warm-start procedure, namely the fact that the approximate cost function for a proposed control law is in general not convex. Note that this small example problem can be solved explicitly with an optimal partition of $N_P = 18$ regions suggesting a pure offline solution using explicit MPC rather than a combined procedure.

7.7.2 Three- and Four-dimensional Examples

After illustrating the procedure on a small problem we now exemplify the approach for assessing the optimal combination problem (7.13) as discussed in Section 7.6.



(a) $K = 0, \epsilon_0 = 3.4373$

(b) $K = 1, \epsilon_1 = 2.3606$

(c) $K = 3, \epsilon_3 = 1.1875$

(d) $K = 5, \epsilon_5 = 0.5290$

Figure 7.1: Warm-start linear programming procedure for Example 1. The solid line represents the cost after K online iterations starting from the B/B approximation in (a) at $K = 0$ and ϵ_K denotes the corresponding approximation error. The lower dashed line is the optimal cost that together with the upper dashed line represents the stability bound.

3D Example Problem

Consider the randomly generated 3D-system:

$$x(k+1) = \begin{bmatrix} -0.5 & 0.3 & -1.0 \\ 0.2 & -0.5 & 0.6 \\ 1.0 & 0.6 & -0.6 \end{bmatrix} x(k) + \begin{bmatrix} -0.601 & -0.890 \\ 0.955 & -0.715 \\ 0.246 & -0.184 \end{bmatrix} u(k) \ ,$$

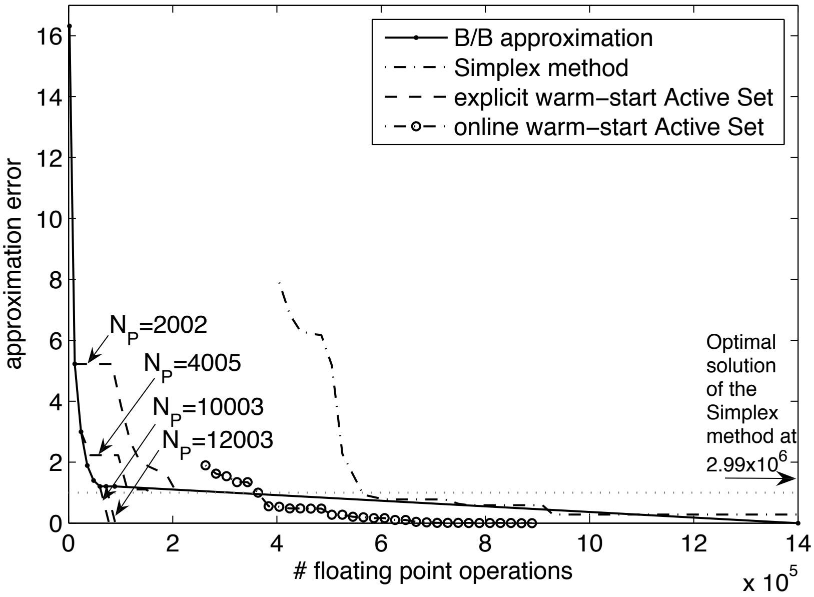
with a prediction horizon $N = 5$ and the constraints $\|u\|_\infty \leq 1$, $\|x\|_\infty \leq 5$ on the input and state respectively. The norm p for the stage cost is taken as the ∞-norm and the weight matrices Q and R are taken as identity matrices.

We first try to solve the problem with the two classic approaches using online and explicit MPC. The explicit solution could not be fully computed due to the high complexity of the example problem and the solution was terminated after 24 hours at a complexity of 14×10^5 regions, which therefore represents a lower bound on the complexity or computation time for the optimal explicit solution. For the online solution, the worst case in the number of iterations and the error was taken over a large number of sample points. A cold start simplex method as well as a warm-start active set method is considered that uses the solution computed at the previous measurement as an initial guess. In order to compare the online warm-start approach, a worst-case additive disturbance $x(k+1) = Ax(k) + Bu(k) + w(k)$ with $\|w\| \leq 0.5$ keeping the state inside the feasible set is considered. The restricted optimal combination problem (7.13) was then solved for a set of warm-start solutions with $N_P \in \{2002, 4005, 10003, 12003\}$ and a maximum approximation error $\epsilon_{max} = 1$ that corresponds to a performance deterioration of about 0.03%, taken over a large number of sample points. The performance deterioration is measured as the relative difference between the cost of the closed loop trajectory using the optimal control input and the one using the suboptimal control input, given by

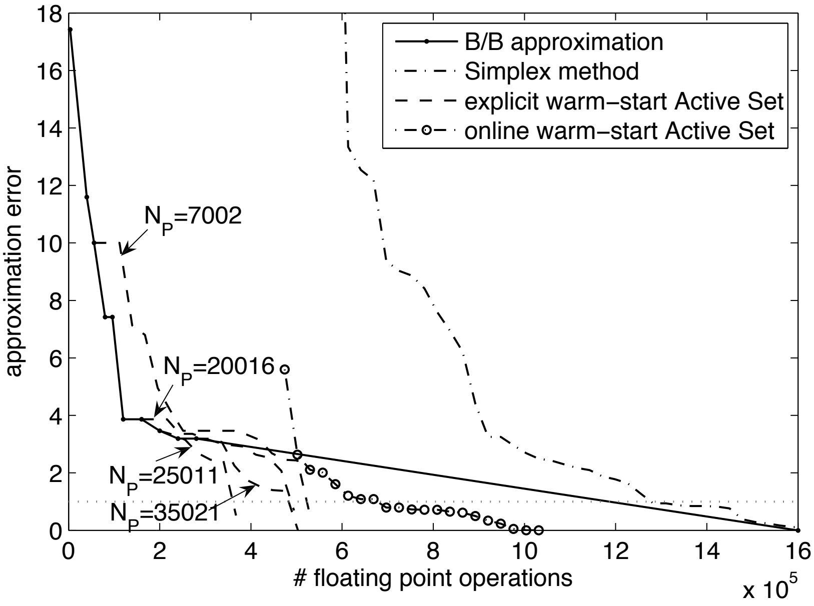
$$\left[\sum_{i=0}^{\infty} \left(l(x_i, \kappa_{\text{on}}(x_i)) - l(x_i, u^*(x_i)) \right) \right] / \sum_{i=0}^{\infty} l(x_i, u^*(x_i)) \ . \tag{7.15}$$

The results are shown in Fig. 7.2(a). The proposed control laws at an approximation error of $\epsilon \leq 1$ are additionally guaranteed to be stabilizing (by Corollary 7.25).

If there are no storage limitations, the best combination of approximation and online iterations is given by the lower envelope of the curves, as it represents the best online computation time for a certain approximation error (or the other way round). Additional limitations on the storage space would restrict the possible warm-start solutions by a maximum number of regions N_P. The online solution using a simplex method starts at a comparably low error after the first feasible point is encountered, but as



(a) Warm-start procedure for Example 2 starting from four different PWA B/B approximations with $N_P \in \{2002, 4005, 10003, 12003\}$.



(b) Warm-start procedure for Example 3 starting from four different PWA B/B approximations with $N_P \in \{7002, 20016, 25011, 35021\}$.

Figure 7.2: The solid line represents the offline B/B approximations. The dashed lines show the improvement by the active set method warm-started from the B/B approximations. The dash-dotted line is a sampled worst-case estimate of a pure online solution using the simplex method and the dotted line using the online warm-start active set method, shown after the first feasible solution is found. The flop number for zero approximation error of the B/B approximation was extrapolated, since the optimal solution could not be computed due to its excessive complexity.

often mentioned in the literature, phase I already takes up a large amount of computation time. In the worst case, the initial solution when using an online warm-start active set method is infeasible requiring several iterations to reach feasibility. It provides a significant improvement when compared to the cold start method but always requires more computation time than the combined approach. The first two warm-start approximations with $N_P = 2002$ and $N_P = 4005$ take significantly more time to achieve a certain approximation error in comparison with the pure B/B approximation. In contrast, the two solutions starting from $N_P = 10003$ and $N_P = 12003$ achieve a clear improvement over the offline approximation.

For any error above 1.2, a pure approximation by the B/B method results in the fastest computation times. The best combination for any error below 1.2 is given by a warm-start solution of complexity $N_P = 10003$ with active set iterations. Note that a further refinement of the warm-start solution does not improve the results and hence this particular combination of warm-start and online solution represents the best combination to achieve an approximation error below 1.2.

4D Example Problem

We will now investigate the optimal combination problem for the 4D randomly generated system:

$$x(k+1) = \begin{bmatrix} -0.251 & 0.114 & 0.123 & -0.433 \\ 0.319 & -0.658 & 0.905 & 0.118 \\ 0.459 & -0.484 & -0.175 & -0.709 \\ 0.016 & 0.116 & -0.002 & -0.505 \end{bmatrix} x(k) + \begin{bmatrix} -0.873 & 0.879 \\ 0.669 & 0.936 \\ -0.353 & 0.777 \\ 0.268 & -0.336 \end{bmatrix} u(k)$$

with a prediction horizon $N = 5$ and the constraints $\|u\|_\infty \leq 5$, $\|x\|_\infty \leq 5$ on the input and state respectively. The norm p for the stage cost is taken as the ∞-norm and the weight matrices Q and R are taken as the identity matrix and two times the identity matrix.

As in the previous example, the explicit solution for this problem is highly complex and could not be fully computed within 24 hours, when it was terminated with a complexity of 16×10^5 regions. The online solution was computed as described in the previous example with $\|w\| \leq 2$. The restricted optimal combination problem (7.13) was solved for a set of warm-start solutions with $N_P \in \{7002, 20016, 25011, 35021\}$ and a maximum approximation error $\epsilon_{max} = 1$ that corresponds to a performance deterioration (7.15) of about 0.03%, taken over a large number of sample points. The results are shown in Fig. 7.2(b). The proposed control laws at an approximation error of $\epsilon \leq 1$ are again guaranteed to be stabilizing (by Corollary 7.25).

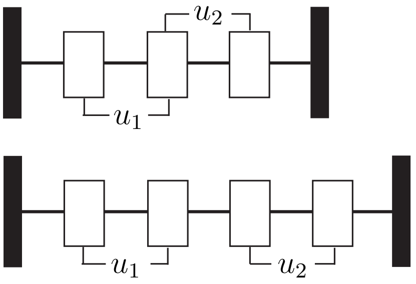


Figure 7.3: Two systems of three and four oscillating masses. The bold lines represent the spring-damper system, the dark blocks on the side represent the walls.

For an approximation error up to 3.2, a pure approximation by the B/B method results in the fastest computation times, but the solution is not guaranteed to be stabilizing. For any error below 3.2, a combination of a warm-start solution of complexity $N_P = 25011$ with active set iterations represents the best combination. Note that a further improvement of the warm-start solution does not provide any benefit. In comparison with a pure online solution, the warm-start procedure is again always superior.

7.7.3 Oscillating Masses Example

After illustrating the proposed procedures on smaller examples we will now demonstrate their application to bigger problem dimensions. We consider the problem of regulating a system of oscillating masses as described in [WB10], one consisting of three and the other of four masses, which are interconnected by spring-damper systems and connected to walls on the side, as shown in Fig. 7.3. The two actuators exert tension between different masses. The masses have value 1, the springs constant 1 and the damping constant is 0.5. The state and input constraints are $\|u\|_\infty \leq 1$, $\|x\|_\infty \leq 4$, the horizon is chosen to $N = 5$, the norm p for the stage cost is taken as the ∞-norm and the weight matrices Q and R are taken as identity matrices. The MPC problem for the 3 masses example has 6 states and 2 inputs, resulting in an LP with 50 optimization variables. The MPC problem for the 4 masses example consists of 8 states and 2 inputs and the LP has 60 optimization variables.

We identify the best combination of explicit approximation and online optimization for providing a hard real-time stability guarantee for these two example problems out of a selection of warm-start solutions by computing the minimal number of optimization steps $K_{\min}$ as well as the number of floating point operations required to provide a suboptimal control law that is guaranteed to be stabilizing. While the optimal explicit solution is not computable for the considered problem dimensions, we compare the flop numbers for the warm-start combinations to the computational effort for a pure online MPC solution. A lower bound on the flop number for computing the optimal solution using a simplex method is estimated by taking the worst case number of iterations over a large number of sampling points. For the 6-dimensional example a worst case number of 51 iterations including phase I and for the 8-dimensional example of 61 iterations was observed. The average closed loop performance deterioration taken over a large number of sampling points is calculated for all warm-start combinations using (7.15). The results are shown in Table 7.1. For the 6-dimensional problem we choose

Table 7.1: Simulation Results for Example 7.7.3

(a) 6D Oscillating Masses Example

# Regions for B/B approximation N_P	3064	5080	7152
# Pivots to stability guarantee $K_{\min}$	15	8	5
Online computation time in kilo flops	789	462	337
Closed loop performance deterioration	6.94%	3.30%	0.63%
Flops for simplex method in kilo flops	2558		

(b) 8D Oscillating Masses Example

# Regions for B/B approximation N_P	3202	5312	7164
# Pivots to stability guarantee $K_{\min}$	17	17	15
Online computation time in kilo flops	1384	1418	1291
Closed loop performance deterioration	5.56%	1.85%	0.82%
Flops for simplex method in kilo flops	4779		

$N_P \in \{3061, 5080, 7152\}$. The number of required optimization steps to reach stability decreases with the refinement of the warm-start solution, resulting in a decreasing number of flops. The best combination is therefore represented by the warm-start solution consisting of 7152 regions, where stability can be guaranteed after only 5 online steps. The average closed loop performance deterioration for this combination is 0.63%. For the 8-dimensional problem a set of warm-start solutions $N_P \in \{3202, 5312, 7164\}$ is computed. While the first two warm-start combinations require at least 17 optimization steps to guarantee stability, the number of iterations is reduced to 15 online iterations for the more complex warm-start solution. Identifying the best combination among these 3 options, we see that it is given by the warm-start solution of 7164 regions combined with 15 online iterations having the lowest flop number, which is additionally supported by the low average performance deterioration of 0.82% observed for this combination.

The offline analysis hence provides a real-time stability guarantee in both test cases for a comparably small number of regions that need to be stored and a small number of optimization steps. The warm-start procedure is always superior to a pure online optimization approach, which takes more than 7 or 4 times the computation time of the best combination for the 6-dimensional or the 8-dimensional example, respectively. An explicit approximation with a stability guarantee could not be computed within 24 hours in both cases due to its high complexity.

While the examples in Section 7.7.2 are small randomly generated example systems that are generally observed to result in problems of high complexity, Example 7.7.3 represents a physical system model of higher dimension and average complexity that is related to many applications involving spring-damper systems (e.g. active suspension). We hereby cover test cases reflecting the scope of the presented approach. The examples in 7.7.2 show that the optimal strategy to achieve a certain set of solution properties is often not to compute the best warm-start solution, but a particular combination of warm-start and online optimization. Example 7.7.3 demonstrates that even for a higher dimensional example the combined procedure clearly outperforms an online solution when minimizing the online computation time and a hard real-time stability guarantee can be provided. The optimal combination of explicit approximation and online optimization does, however, highly depend on the particular problem structure and the given requirements on storage space and performance. For certain problems the best solution procedure will be a particular combination of the two methods whereas for others it will as well be a pure offline or online approximation, e.g. in the case of extremely small and simple or large problems, which can be identified by means of the presented analysis.

7.8 Conclusions

We presented a new approach that combines the two paradigms of online and explicit MPC and hereby offers new possibilities in the applicability of MPC to real problems that often have limits on the storage space or the available computation time. The proposed method computes a piecewise affine approximation of the optimal solution offline that is stored and used to warm-start an active set method. By means of a preprocessing analysis hard real-time, stability and performance guarantees for the proposed controller are provided. The analysis does not require the calculation of the optimal parametric solution to the MPC problem, since it may be prohibitively complex and could restrict the applicability of the method.

The warm-start procedure enlarges the possibilities to tradeoff solution properties in order to satisfy constraints in terms of online computation time, storage and performance. The best solution method is dependent on the particular system as well as the given hardware and performance restrictions. We show how the offline analysis can be utilized to compare different MPC methods and identify the best approach for a considered application and set of requirements. In addition to the discussed aspects, desired implementation properties may affect the choice of the MPC method. By using a combination of explicit approximation and online MPC, the method inherits the properties of both paradigms and therefore also the numerical and computational challenges of online optimization, such as the need for floating point computations or software verification and maintenance.

The presented numerical examples illustrate the proposed procedures and confirm the fact that a warm-start solution can often outperform either a pure offline or online method. The warm-start procedure provides hard real-time guarantees on the applied suboptimal controller where an approximate explicit or online approach is either intractable or can not meet the given requirements.

8 On Robust Real-time MPC Using Online Optimization

8.1 Introduction

Computation of the optimal Model Predictive Control (MPC) law is generally not practical when controlling high speed systems, which impose a strict real-time constraint on the solution of an MPC problem. The goal is then to provide a suboptimal control action within the time constraint that still guarantees stability of the closed-loop system and achieves acceptable performance. In this chapter we develop a real-time MPC scheme that guarantees stability and constraint satisfaction for all time constraints and allows for fast online computation. The a-priori stability guarantee then allows one to trade the performance of the suboptimal controller for lower online computation times.

This work is motivated by recent results showing that the computation times for solving an MPC problem can be pushed into a range where an online optimization becomes a reasonable alternative for the control of high-speed systems, see also the overview in Chapter 6. Significant reduction of the computational complexity can be achieved by exploiting the particular structure and sparsity of the optimization problem given by the MPC problem using tailored solvers. Available methods for fast online MPC do not give guarantees on either feasibility or stability of the applied control action in a real-time implementation. A method providing these guarantees by combining online and explicit MPC was introduced in Chapter 7, which is, however, still limited to smaller problem dimensions.

This work makes the following contributions: it is shown how feasibility and input-to-state stability can be guaranteed in a real-time MPC approach for linear systems under additive disturbances using robust MPC design and a Lyapunov constraint, while allowing for low computation times with the same complexity as MPC methods without guarantees. All computational details required for a fast implementation based on a barrier interior-point method are provided and results in the robust MPC literature are consolidated into a step-by-step implementation for large-scale systems. Existing results for the utilization of sparsity do not apply in the considered real-time robust setting, resulting in a quadratically constrained quadratic program (QCQP) for large-scale systems and it is shown how the new structure can be exploited.

The developed real-time method employs a standard warm-start procedure, in which the optimization problem for the current state is initialized with the shifted suboptimal control sequence computed at the previous time instance. The optimization is terminated early when a specified time constraint τ is hit. While stability of the robust real-time control law may seem to follow directly from the warm-start procedure, we will show that it is not automatically provided if one uses interior-point methods, as is required for solving the resulting QCQPs. In order to guarantee input-to-state stability in real-time, a constraint is introduced that explicitly enforces that the real-time MPC cost is a Lyapunov function. Feasibility of the resulting real-time control law is provided by the use of a robust MPC framework, recovering recursive feasibility in an uncertain setting. The proposed real-time method is then extended to the problem of tracking piecewise constant references that is faced in many control applications in practice, using a recently proposed MPC formulation for tracking [ALA$^+$07, LAAC08] described in Section 5.4.

An implementation of the presented real-time MPC procedure for uncertain linear systems is developed based on the robust MPC approach in [MSR05] in order to allow for fast computation, see Section 5.3 for details on the method. A primal barrier interior-point method (see Section 3.1 or [BV04, NW06]) is developed to realize the robust real-time control law. The tracking formulation with stability guarantees significantly modifies the structure of the considered optimization problem and the results on structure exploitation in MPC problems presented in the literature, see e.g. [Wri97a, Han00, WB10], can no longer be applied. We show how the new structure and sparsity can be exploited and solved efficiently using a solver tailored for the resulting optimization problem. A custom solver was developed for this work that achieves computation times that are equal or even faster when compared to existing methods with no guarantees. For a 12-dimensional example system an MPC problem with a limit of 5 interior-point iterations was solved in 2msec with an average performance deterioration of less than 1%. The corresponding computation times for a 30-dimensional system were 10msec.

The outline of the chapter is as follows: In Section 8.2 the challenges of real-time MPC are introduced. Section 8.3 presents the proposed real-time robust MPC method and proves input-to-state stability of the closed-loop system under the real-time control law. The results are extended to the more general case of tracking piecewise constant references in Section 8.4. Section 8.5 provides the implementation details necessary for the robust MPC problem setup. In Section 8.6 a real-time method is developed based on a warm-start approximate primal barrier interior-point method and it is shown how the structure of the resulting optimization problem can be exploited for fast computation. Finally, in Section 8.7 we illustrate the proposed approach and its advantages using numerical examples and provide a comparison with the literature.

8.2 Real-time MPC Based on Online Optimization – Problem Statement

High-speed applications impose a real-time constraint on the solution of the MPC problem, i.e. a limit on the computation time that is available to compute the control input, which often prevents the computation of the optimal MPC control law. This can lead to constraint violation, and more importantly, instability when using a general optimization solver. A suboptimal control input therefore has to be provided within the real-time constraint that ensures constraint satisfaction and stability. In the following a control law is called τ-real time (τ-RT) if it is computed in τ seconds.

Various approaches aimed at reducing the computation time in online MPC have been proposed recently. Many methods are based on the development of custom solvers that take advantage of the particular sparse structure in an MPC problem see e.g. [Wri97a, WB10, MD08, AH08] and the overview in Section 6.2. In [WB10], for example, an infeasible start Newton method is applied that is terminated after a fixed number of steps. A tailored solver was developed that exploits the sparse structure of the MPC problem resulting in computation times in the range of milliseconds. The authors in [FBD08] develop a warm-start based homotopy approach that is terminated early in case of a time constraint. Available approaches sacrifice constraint satisfaction and/or stability in order to achieve a real-time guarantee. In a recent work [LH09] a relation between the level of suboptimality and a guarantee of stability is derived. These results can, however, not be applied to the considered case of real-time MPC, since it is currently not possible to determine the level of suboptimality that a given solver will achieve in a fixed amount of time.

A real-time MPC scheme based on a warm-start method that is commonly applied in practice is described in Algorithm 3. An initial feasible solution at state $x(k)$ is constructed from the solution computed for the measurement $x(k-1)$ at the previous sampling time [SMR99, MRRS00]:

$$\mathbf{u}^{\mathrm{ws}}(x(k)) = \left[u_1(x(k-1)), \ldots, u_{N-1}(x(k-1)), \kappa_f(\bar{\phi}(N, x(k-1), \mathbf{u}(x(k-1))))\right] \ , \tag{8.1}$$

where $\kappa_f(x)$ is the local control law in Assumption 5.3. The warm-start solution is then improved using online optimization until the real-time constraint is hit. We show in the following why directly applying Algorithm 3 to the standard MPC problem $\mathbb{P}_N(x)$ does

Algorithm 3 Warm-start real-time procedure

Input: feasible control sequence $\mathbf{u}(x(k-1))$ for $x(k-1)$, current state measurement $x(k)$ and auxiliary control law $\kappa_f(x)$

Output: τ-RT control sequence $\mathbf{u}^\tau(x(k))$

```
Warm-start:
$\mathbf{u}^{\mathrm{ws}}(x(k)) = \big[u_1(x(k-1)), \ldots, u_{N-1}(x(k-1)), \kappa_f(\bar{\phi}(N, x(k-1), \mathbf{u}(x(k-1))))\big]$
$\tilde{\mathbf{u}} = \mathbf{u}^{\mathrm{ws}}(x(k))$
while clock$< \tau$ do
    improve $\tilde{\mathbf{u}}$ in one optimization step
end while
$\mathbf{u}^\tau(x(k)) = \tilde{\mathbf{u}}$
```

not a priori provide constraint satisfaction and stability in order to then outline how a τ-RT control law can be derived that guarantees stability of the closed-loop system.

It is well-known that in the nominal case the warm-start solution $\mathbf{u}^{\mathrm{ws}}(x(k))$ in (8.1) provides a feasible and stabilizing control law for the nominal system in (4.4) if no online optimization steps are executed (i.e. running the computed MPC control sequence in open loop will drive the system to the origin if there is no noise). In order to guarantee feasibility and stability of the closed-loop system under the τ-RT control law for all subsequent times $\tau \geq 0$, the optimization method then has to maintain feasibility and ensure that the cost function decreases with respect to the cost at the last sampling time [SMR99, MRRS00, LAR$^+$09]. While one would expect this property to be automatically satisfied, it is in fact not provided by interior-point methods. As will be shown in Section 8.6, interior-point methods are, however, required in order to solve an MPC problem with the desired guarantees efficiently, while pivoting methods like e.g. active set methods do not provide a suitable alternative for the solution of the resulting quadratically constrained QPs.

The solution that is obtained from directly applying Algorithm 3 together with a barrier interior-point method, which is proposed in existing methods (e.g. [WB10]), is therefore not guaranteed to provide asymptotic stability of the origin. This is due to the fact that in a barrier interior-point method the inequality constraints are replaced by a barrier penalty in the cost function. At each interior-point iteration the augmented cost including the barrier penalty is decreased, which does not simultaneously enforce a decrease of the MPC cost. The MPC cost can therefore not be employed as a Lyapunov function and the standard stability proof in MPC fails [MRRS00]. Furthermore, the solution of the augmented problem with the barrier term only approaches the solution of the original problem in the limit, if the barrier parameter is taken to zero. Early termination of the optimization then causes the system to converge to a steady-state that minimizes the augmented cost and results in a steady-state offset.

In addition, feasibility of the warm-start in (8.1) is lost in practice due to model inaccuracies or disturbances causing the system to deviate from the nominal system dynamics in (4.4). Since it cannot be guaranteed that feasibility is recovered by the optimization procedure in a fixed amount of time when starting from an initial infeasible solution, feasibility cannot be guaranteed in such a real-time framework. This can be prevented by the use of robust MPC approaches, providing recursive feasibility in an uncertain environment.

Many control applications in practice require tracking of a desired sequence of steady-states rather than regulation around the origin or to a particular steady state. In order to achieve tracking, it is standard practice to modify the MPC problem $\mathbb{P}_N(x)$ by means of a change of variables such that the deviation from the state and input reference is penalized and the terminal state constraint is modified, requiring the terminal state to lie in an invariant set around the state reference (see e.g. [Mac00, RM09]). This problem formulation does, however, not provide feasibility of the warm-start in (8.1) since the terminal constraint depends directly on the reference, rendering the sequence computed at the last sampling time infeasible for the new terminal constraint after a reference change.

In this work, feasibility is achieved by means of the previously described warm-start procedure using a primal feasible optimization routine together with a tube-based robust MPC scheme that recovers recursive feasibility by tightening the constraints and a slight change of the problem formulation [MSR05]. In order to guarantee input-to-state stability in real-time, a Lyapunov constraint is introduced, ensuring a decrease in the MPC cost with respect to the last sampling time if the system satisfied the nominal dynamics and providing a Lyapunov function. First, the proposed real-time robust MPC method for regulation is described in the following section and input-to-state stability is proven. We then show how the presented ideas can be extended to the case of reference tracking using a recently proposed tracking approach [LAAC08]. The remainder of the chapter focuses on the practical and fast implementation of the proposed robust real-time MPC scheme.

8.3 Real-time Robust MPC with Guarantees

Consider the discrete-time uncertain system in (4.1). We propose the following real-time robust MPC problem formulation to realize a τ-RT control law that guarantees input-to-state stability of the closed loop system:

Problem $\mathbb{P}_N^\tau(x)$ (Real-time robust MPC problem)

$$\begin{aligned}
\min_{\bar{\mathbf{x}},\bar{\mathbf{u}}} \quad & V_N(\bar{\mathbf{x}},\bar{\mathbf{u}}) + V_f(x-\bar{x}_0) && = \sum_{i=0}^{N-1} l(\bar{x}_i,\bar{u}_i) + V_f(\bar{x}_N) + V_f(x-\bar{x}_0) && \text{(8.2a)}\\
\text{s.t.} \quad & \bar{x}_{i+1} && = A\bar{x}_i + B\bar{u}_i, \;\; i = 0,\dots,N-1 \;\; , && \text{(8.2b)}\\
& (\bar{x}_i,\bar{u}_i) && \in \bar{\mathbb{X}} \times \bar{\mathbb{U}}, \;\; i = 0,\dots,N-1 \;\; , && \text{(8.2c)}\\
& \bar{x}_N && \in \bar{\mathcal{X}}_f \;\; , && \text{(8.2d)}\\
& V_N(\bar{\mathbf{x}},\bar{\mathbf{u}}) + V_f(x_{\text{nom}}^\tau - \bar{x}_0) && \leq \pi_{\text{prev}}^\tau \;\; , && \text{(8.2e)}\\
& x && \in \bar{x}_0 \oplus \mathcal{Z}_{\mathcal{W}} \;\; , && \text{(8.2f)}
\end{aligned}$$

where $\mathcal{Z}_{\mathcal{W}}$ is an RPI set for the controlled system $x(k+1) = Ax(k) + B\kappa_f(x(k)) + w(k)$ with $w(k) \in \mathcal{W}\ \forall k \in \mathbb{N}$, $\bar{\mathbb{X}} = \mathbb{X} \ominus \mathcal{Z}_{\mathcal{W}}$, $\bar{\mathbb{U}} = \mathbb{U} \ominus K\mathcal{Z}_{\mathcal{W}}$ are tightened constraints on the states and inputs and $\bar{\mathcal{X}}_f$ is an invariant terminal target set. A quadratic stage cost and terminal penalty function is chosen, i.e. $l(x,u) \triangleq \|x\|_Q^2 + \|u\|_R^2$, $V_f(x) \triangleq \|x\|_P^2$, where Q, R and P are positive definite matrices. The constants x_{nom}^τ and π_{prev}^τ are defined in Definition 8.2 below. Problem $\mathbb{P}_N^\tau(x)$ implicitly defines the set of feasible control sequences $\mathcal{U}_N^\tau(\bar{x}_0) = \{\bar{\mathbf{u}} \mid \exists\ \bar{\mathbf{x}} \text{ s.t. (8.2b)} - \text{(8.2e) hold}\}$, feasible initial tube centers $\mathcal{X}_0^\tau(x) \triangleq \{\bar{x}_0 \mid \text{(8.2f)}\}$ and feasible initial states $\mathcal{X}_N^\tau = \{x \mid \exists\ \bar{x}_0 \in \mathcal{X}_0^\tau(x) \text{ s.t. } \mathcal{U}_N^\tau(\bar{x}_0) \neq \emptyset\}$.

Assumption 8.1. It is assumed that $V_f(\bullet)$ and $\bar{\mathcal{X}}_f$ fulfill Assumption 5.3 with $\bar{\mathbb{X}}$, $\bar{\mathbb{U}}$ and $\bar{\mathcal{X}}_f$ replacing $\mathbb{X}$, $\mathbb{U}$ and $\mathcal{X}_f$.

Instead of solving problem $\mathbb{P}_N^\tau(x)$ to optimality, the optimization is executed at $x(k)$ for no more than τ seconds before returning the variables $\bar{\mathbf{u}}^\tau(x(k))$ and $\bar{\mathbf{x}}^\tau(x(k))$ by applying Algorithm 4 together with a primal feasible optimization routine to Problem $\mathbb{P}_N^\tau(x)$. The robust τ-RT control law is then given in a receding horizon control fashion by

$$\kappa^\tau(x) = \bar{u}_0^\tau(x) + K(x - \bar{x}_0^\tau(x)) \;\; . \tag{8.3}$$

Definition 8.2 (Lyapunov constraint). For each $x(k)$, we take

$$x_{\text{nom}}^\tau = Ax(k-1) + B\kappa^\tau(x(k-1))$$

to be the state that would have been obtained in the absence of disturbances and

$$\pi_{\text{prev}}^\tau = V_N(\bar{\mathbf{x}}^\tau(x(k-1)), \bar{\mathbf{u}}^\tau(x(k-1))) + V_f(x - \bar{x}_0^\tau(x(k-1))) - \epsilon\frac{1}{2}l(x(k-1),0) \;\; , \tag{8.4}$$

where $\epsilon \in (0,1]$ is a user-specified constant, $x(k-1)$ denotes the state and $V_N(\bar{\mathbf{x}}^\tau(x(k-1)), \bar{\mathbf{u}}^\tau(x(k-1))) + V_f(x - \bar{x}_0^\tau(x(k-1)))$ the cost at the previous, i.e. the $(k-1)$'th, sample time.

Algorithm 4 Warm-start real-time robust procedure

Input: feasible control sequence $\bar{\mathbf{u}}^{\text{in}}(x(k-1))$, tube center $\bar{x}_0^{\text{in}}(x(k-1))$ and corresponding state sequence $\bar{\mathbf{x}}^{\text{in}}(x(k-1))$ for $x(k-1)$, current state measurement $x(k)$, auxiliary control law $\kappa_f(x) = Kx$ and parameter $\epsilon_f > 0$

Output: τ-RT control sequence $\bar{\mathbf{u}}^{\tau}(x(k))$ and tube center $\bar{x}_0^{\tau}(x(k))$

```
1: Warm-start: $\bar{x}_0^{\text{ws}}(x(k)) = \bar{x}_1^{\text{in}}(x(k-1)))$
   $\bar{\mathbf{u}}^{\text{ws}}(x(k)) = \left[\bar{u}_1^{\text{in}}(x(k-1)), \ldots, \bar{u}_{N-1}^{\text{in}}(x(k-1)), \kappa_f(\bar{x}_N^{\text{in}}(x(k-1)))\right]$
2: $\tilde{\mathbf{u}} = \bar{\mathbf{u}}^{\text{ws}}(x(k))$, $\tilde{x}_0 = \bar{x}_0^{\text{ws}}(x(k))$
3: while clock< $\tau$ do
4:     improve $\tilde{\mathbf{u}}, \tilde{x}_0$ in one primal feasible optimization step
5: end while
6: if $\|x(k)\|_P \le \epsilon_f$ and $V_N(\tilde{\mathbf{x}}, \tilde{\mathbf{u}}) + V_f(x(k) - \tilde{x}_0) > V_f(x(k))$ then
7:     $\tilde{x}_0 = 0$, $\tilde{\mathbf{u}} = 0$
8: end if
9: $\bar{\mathbf{u}}^{\tau}(x(k)) = \tilde{\mathbf{u}}, \bar{x}_0^{\tau}(x(k)) = \tilde{x}_0$
```

Remark 8.3. Note that the real-time robust MPC problem $\mathbb{P}_N^{\tau}(x)$, the real-time state and input sequences $\bar{\mathbf{u}}^{\tau}(x(k))$ and $\bar{\mathbf{x}}^{\tau}(x(k))$ as well as the resulting robust τ-RT control law $\kappa^{\tau}(x)$ in (8.3) are not only a function of the current state, but also of the previous state, the real-time solution $\bar{\mathbf{u}}^{\tau}(x(k-1))$ and $\bar{\mathbf{x}}^{\tau}(x(k-1))$ computed at the previous time step as well as the available computation time τ, which are, however, given constants at the time of computation. We omit this dependence for ease of notation, but denote all variables and functions that are dependent on the solution from the previous time step and the computation time by the index τ.

The real-time robust Problem $\mathbb{P}_N^{\tau}(x)$ differs from the nominal MPC problem $\mathbb{P}_N(x)$ in two main aspects: the use of a robust MPC problem setup and a Lyapunov constraint.

- *Robust MPC design:* Constraints (8.2c) and (8.2f) result from the tube based robust MPC approach in [MSR05], providing robustness with respect to the additive disturbance w in (4.1). The method is based on the use of a feedback policy of the form $u = u + K(x - \bar{x})$ that bounds the effect of the disturbances and keeps the states x of the uncertain system in (4.1) close to the states $\bar{x}$ of the nominal system in (4.4). The robust MPC problem can therefore be reduced to the control of the tube centers $\bar{x}$, which are steered to the origin by choosing a sequence of control inputs $\bar{\mathbf{u}}$ and the first tube center $\bar{x}_0$. The tightened state and input constraints in (8.2c) are used in order to ensure feasibility of the uncertain system in (4.1) despite the disturbance w. Note that the first tube center $\bar{x}_0$ is not necessarily equal to the current state measurement x, but is an optimization variable satisfying (8.2f). Compared to [MRRS00] the cost is augmented by the term $V_f(x - \bar{x}_0)$, which introduces a tradeoff between the amount of control action used for counteracting the disturbance and the effort for controlling the nominal state to the origin. See Section 5.3 for more details and properties of this approach.

- *Lyapunov decrease constraint:* Constraint (8.2e) ensures that the suboptimal cost achieved after τ seconds satisfies the Lyapunov decrease condition (4.10c), which is essential in order for the MPC cost to provide a ISS Lyapunov function. It explicitly enforces that the MPC cost (8.2a) at $x(k)$ decreases with respect to the cost at the last sample time $V_N(\bar{\mathbf{x}}(x(k-1)), \bar{\mathbf{u}}(x(k-1))) + V_f(x - \bar{x}_0^\tau(x(k-1)))$ if the system satisfies the nominal dynamics and thereby recovers the stability properties of the optimal robust MPC approach in a real-time setting. The Lyapunov decrease constraint (8.2e) represents a convex quadratic constraint on the optimization variables.

The proposed Algorithm 4 differs from the standard real-time Algorithm 3 in one main aspect:

- *Upper bound on Lyapunov function:* Steps 6-8 ensure that the real-time cost can be upper bounded by a $\mathcal{K}_\infty$-class function of the state in a set that includes the origin in its interior. Here $V_f(x)$ is chosen, since it represents an upper bound on the optimal MPC solution and in particular on the cost for using $\tilde{x}_0 = 0$, $\tilde{\mathbf{u}} = 0$. Although it is a more technical requirement than the decrease in the Lyapunov function, the upper bound together with the Lyapunov decrease constraint (8.2e) provide that the MPC cost is a ISS Lyapunov function (Lemma 8.6). While Steps 6,7 are therefore mainly motivated from theoretical considerations, they are beneficial in a real-time environment, since the auxiliary control law is only used if it provides a lower cost than the solution obtained in the real-time optimization.

In order to guarantee feasibility and stability of the closed-loop system under the τ-RT control law in (8.3), the warm-start computed in Step 1 of Algorithm 4 has to be feasible for $\mathbb{P}_N^\tau(x)$ with respect to all state and input constraints as well as the Lyapunov decrease constraint.

Lemma 8.4 (Feasibility of the warm-start) *Let $\epsilon_f > 0$ be a positive constant such that $\mathcal{S} \triangleq \{x \mid \|x\|_P \leq \epsilon_f\} \subseteq \mathcal{Z}_\mathcal{W}$. The warm-start solution provided by Algorithm 4 is feasible for $\mathbb{P}_N^\tau(x^+)$, where $x^+ \in Ax + B\kappa^\tau(x) \oplus \mathcal{W}$, i.e. $\bar{x}_0^{ws}(x^+) \in \mathcal{X}_0^\tau(x^+)$, $\bar{\mathbf{u}}^{ws}(x^+) \in \mathcal{U}_N^\tau(\bar{x}_0^{ws}(x^+))$.*

Proof. Feasibility of the warm-start for Problem $\mathbb{P}_N^\tau(x)$ without the Lyapunov constraint (8.2e) was shown in [MSR05]. Note that

$$x_{\text{nom}}^\tau = Ax + B\kappa^\tau(x) = Ax - A\bar{x}_0^{\text{in}}(x) + A\bar{x}_0^{\text{in}}(x) + B\bar{u}_0^{\text{in}}(x) + BK(x - \bar{x}_0^{\text{in}}(x))$$

$$= \bar{x}_1^{\text{in}}(x) + (A + BK)(x - \bar{x}_0^{\text{in}}(x)) \ .$$

From Assumption 8.1 and standard arguments in MPC, we obtain $V_N(\bar{\mathbf{x}}^{\text{ws}}(x^+), \bar{\mathbf{u}}^{\text{ws}}(x^+)) - V_N(\bar{\mathbf{x}}^{\text{in}}(x), \bar{\mathbf{u}}^{\text{in}}(x)) \leq -\|\bar{x}_0^{\text{in}}(x)\|_Q^2$. Using similar arguments as in the proof of Theorem 5.8, Assumption 8.1 and convexity of $\|\bullet\|_Q^2$, providing $\frac{1}{2}\|x+y\|_Q^2 \leq \|x\|_Q^2 + \|y\|_Q^2$, it can then be shown that

$$\begin{aligned} & V_N(\bar{\mathbf{x}}^{\text{ws}}(x^+), \bar{\mathbf{u}}^{\text{ws}}(x^+)) - V_N(\bar{\mathbf{x}}^{\text{in}}(x), \bar{\mathbf{u}}^{\text{in}}(x)) + V_f(x_{\text{nom}}^\tau - \bar{x}_0^{\text{ws}}(x^+)) - V_f(x - \bar{x}_0^{\text{in}}(x)) \\ & \leq -\|\bar{x}_0^{\text{in}}(x)\|_Q^2 + V_f((A+BK)(x - \bar{x}_0^{\text{in}}(x))) - V_f(x - \bar{x}_0^{\text{in}}(x)) \\ & \leq -\|\bar{x}_0^{\text{in}}(x)\|_Q^2 - \|x - \bar{x}_0^{\text{in}}(x)\|_Q^2 \leq -\frac{1}{2}\|x\|_Q^2 \end{aligned}$$

and the warm-start therefore also satisfies the Lyapunov constraint (8.2e). ■

Remark 8.5 (Initialization). It is assumed that at time $k = 0$, before starting the real-time control of the system, enough computation time is available to compute a feasible solution to Problem $\mathbb{P}_N^\tau(x)$ without the Lyapunov constraint (8.2e) and initialize Algorithm 4.

While asymptotic stability of the origin cannot be achieved in the presence of disturbances, it can be shown that under certain conditions the closed-loop system is input-to-state stable. Note that in the considered real-time case, stability cannot be achieved by the approach described in [LRH+08], where a constraint on the Lyapunov decrease is only introduced in the first step, since the solutions are not recursively feasible.

Due to the Lyapunov constraint (8.2e), feasibility of $\mathbb{P}_N^\tau(x)$ implies input-to-state stability, which is stated in the following theorem.

Lemma 8.6 *Let $\tilde{x}_0^\tau(x(k)) \in \mathcal{X}_0^\tau(x(k))$ be a feasible tube center, $\tilde{\mathbf{u}}^\tau(x(k)) \in \mathcal{U}_N^\tau(\tilde{x}_0(x(k)))$ a feasible control sequence for $\mathbb{P}_N^\tau(x(k))$ for all $k \in \mathbb{N}$ and $\tilde{\kappa}^\tau(x(k)) = \tilde{u}_0^\tau(x(k)) + K(x - \tilde{x}_0^\tau(x(k)))$ the resulting control law. Let $\epsilon_f > 0$ be a positive constant such that $\mathcal{S} \triangleq \{x \mid \|x\|_P \leq \epsilon_f\} \subseteq \mathcal{Z}_\mathcal{W}$ and assume that $V_N(\tilde{\mathbf{x}}^\tau(x(k)), \tilde{\mathbf{u}}^\tau(x(k))) + V_f(x - \tilde{x}_0^\tau(x(k))) \leq V_f(x(k))$ if $x(k) \in \mathcal{S}$. The closed-loop system $x(k+1) = Ax(k) + B\tilde{\kappa}^\tau(x(k)) + w(k)$ is ISS with respect to $w(k) \in \mathcal{W}$ with region of attraction $\mathcal{X}_N^\tau$.*

Proof. We define $V_L^\tau(x) \triangleq V_N(\tilde{\mathbf{x}}^\tau(x), \tilde{\mathbf{u}}^\tau(x)) + V_f(x - \tilde{x}_0^\tau(x))$. Assumption 8.1 provides that $\|\bullet\|_Q^2 \leq \|\bullet\|_P^2$. Using convexity of $\|\bullet\|_Q^2$, it can be shown that there exists a $\mathcal{K}_\infty$-class function $\underline{\alpha}(\bullet)$ such that

$$V_L^\tau(x) \geq \|\tilde{x}_0^\tau(x)\|_Q^2 + \|x - \tilde{x}_0^\tau(x)\|_Q^2 \geq \frac{1}{2}\|x\|_Q^2 \geq \underline{\alpha}(\|x\|) \ \forall x \in \mathcal{X}_N^\tau \ . \qquad (8.5)$$

By the assumptions in the lemma, there exists a $\mathcal{K}_\infty$-class function $\overline{\alpha}(\bullet)$ such that

$$V_L^\tau(x) \leq V_f(x) \leq \overline{\alpha}(\|x\|) \ \forall x \in \mathcal{S} \ . \tag{8.6}$$

Furthermore, $V_N(\tilde{\mathbf{x}}^\tau(x^+), \tilde{\mathbf{u}}^\tau(x^+)) + V_f(x^\tau_{\text{nom}} - \tilde{x}^\tau_0(x^+)) - V_L(x) \leq -\epsilon\frac{1}{2}\|x\|^2_Q$ for all $x \in \mathcal{X}_N^\tau$, where $x^+ = Ax + B\kappa^\tau(x) + w$, by the Lyapunov constraint (8.2e), which implies that

$$\begin{aligned}
&V_L(x^+) = V_N(\tilde{\mathbf{x}}^\tau(x^+), \tilde{\mathbf{u}}^\tau(x^+)) + V_f(x^+ - \tilde{x}^\tau_0(x^+)) \\
&\ = V_N(\tilde{\mathbf{x}}^\tau(x^+), \tilde{\mathbf{u}}^\tau(x^+)) + V_f(x^\tau_{\text{nom}} - \tilde{x}^\tau_0(x^+)) + V_f(x^+ - \tilde{x}^\tau_0(x^+)) - V_f(x^\tau_{\text{nom}} - \tilde{x}^\tau_0(x^+)) \\
&\ \leq V_L(x) - \epsilon\frac{1}{2}\|x\|^2_Q + |V_f(x^+ - \tilde{x}_0(x^+)) - V_f(x^\tau_{\text{nom}} - \tilde{x}_0(x^+))| \ .
\end{aligned}$$

Since $V_f(x)$ is a continuous function, there exists a $\mathcal{K}$-class function $\gamma(\bullet)$ such that $|V_f(y) - V_f(x)| \leq \gamma(\|y - x\|)$. Therefore, there exists a suitable $\mathcal{K}_\infty$-class function $\beta(\bullet)$ such that

$$\begin{aligned}
V_L^\tau(x^+) - V_L^\tau(x) &\leq -\frac{1}{2}\epsilon\|x\|^2_Q + \gamma(\|x^+ - x^\tau_{\text{nom}}\|) && \text{(8.7a)}\\
&\leq -\beta(\|x\|) + \gamma(\|w\|) \ \forall x \in \mathcal{X}_N^\tau \ . && \text{(8.7b)}
\end{aligned}$$

The cost function in (8.2a) is hence a ISS Lyapunov function proving ISS of the closed-loop system by Theorem 4.14. ■

We can now state the main result of this section and prove ISS of the closed-loop system under the τ-real-time robust control law $\kappa^\tau(x)$ in (8.3).

Theorem 8.7 (Stability under $\boldsymbol{\kappa^\tau(x)}$) *Consider Problem $\mathbb{P}_N^\tau(x)$ fulfilling Assumption 8.1. The closed-loop system $x(k+1) = Ax(k) + B\kappa^\tau(x) + w(k)$ under the τ-RT control law in* (8.3) *that is obtained from Algorithm 4 is ISS w.r.t. $w(k) \in \mathcal{W}$ with region of attraction $\mathcal{X}_N^\tau$ for all $\tau \geq 0$.*

Proof. Let $\tilde{\mathbf{x}}^\tau(x), \tilde{\mathbf{u}}^\tau(x)$ denote the variables obtained at Step 9 of Algorithm 4. We define again $V_L^\tau(x) \triangleq V_N(\tilde{\mathbf{x}}^\tau(x), \tilde{\mathbf{u}}^\tau(x)) + V_f(x - \tilde{x}^\tau_0(x))$ and $\mathcal{S} \triangleq \{x \mid \|x\|_P \leq \epsilon_f\} \subseteq \mathcal{Z}_\mathcal{W}$. Feasibility of the warm-start provided by Algorithm 4 was shown in Lemma 8.4, which is maintained by the use of a primal feasible optimization method. If Steps 6-8 are never applied, ISS follows directly from Lemma 8.6. If Steps 6-8 are applied for some $x(k) \in \mathcal{S}$, it is known that $\tilde{\mathbf{u}} = 0, \tilde{x}_0 = 0$ is feasible for $\mathbb{P}_N^\tau(x(k))$ without the Lyapunov constraint and $V_L^\tau(x(k)) = V_N(0,0) + V_f(x(k)) = V_f(x(k))$. This provides the upper bound in (8.6). The lower bound is given by (8.5). Let $\tilde{\mathbf{u}}^\tau_{\text{opt}}, \tilde{\mathbf{x}}^\tau_{\text{opt}}$ denote the solution that is returned by the optimization. Since it is feasible for $\mathbb{P}_N^\tau(x(k))$ including the Lyapunov constraint, it follows from (8.7) in the proof of Lemma 8.6 that

$$V_N(\tilde{\mathbf{x}}^\tau_{\text{opt}}, \tilde{\mathbf{u}}^\tau_{\text{opt}}) + V_f(x(k) - \tilde{x}^\tau_{0,\text{opt}}) \leq V_L(x(k-1)) - \beta(\|x(k-1)\|) + \gamma(\|w(k-1)\|) \ \forall x(k) \in \mathcal{X}_N^\tau$$

for some $\mathcal{K}$-class function $\gamma(\cdot)$ and $\mathcal{K}_\infty$-class function $\beta(\cdot)$. Finally, since the conditions in Step 6 are fulfilled, it follows that

$$\begin{aligned} V_L^\tau(x(k)) = V_f(x(k)) &\leq V_N(\tilde{\mathbf{x}}_{\text{opt}}^\tau, \tilde{\mathbf{u}}_{\text{opt}}^\tau) + V_f(x(k) - \tilde{x}_{0,\text{opt}}^\tau) \\ &\leq V_L^\tau(x(k-1)) - \beta(\|x(k-1)\|) + \gamma(\|w(k-1)\|) \ \forall x(k) \in \mathcal{X}_N^\tau \ . \end{aligned}$$

$V_L^\tau(x)$ is therefore a ISS Lyapunov function, proving the result. ■

Theorem 8.7 guarantees stability of the uncertain system (4.1) in a real-time MPC implementation by using the robustified problem formulation in $\mathbb{P}_N^\tau(x)$ with an additional Lyapunov constraint (8.2e). The optimization solving $\mathbb{P}_N^\tau(x)$ can be stopped after an arbitrary available time τ.

Remark 8.8. Note that τ can be arbitrarily time-varying, which makes the presented approach suitable for operation in a wide range of standard multi-tasking real-time computational platforms.

Remark 8.9. The use of $\tilde{x}_0 = 0$, $\tilde{\mathbf{u}} = 0$ results in the control law $\kappa^\tau(x) = Kx$. By using the auxiliary control law in a neighborhood $\mathcal{S}$ of the origin (if the cost cannot be upper bounded by $V_f(x)$), Algorithm 4 is similar to a dual mode strategy. The difference is that the control strategy does not switch to this control law once the state is inside this set, since $\mathcal{S}$ is not robustly invariant.

Remark 8.10. While the existence of a $\mathcal{K}_\infty$-class function of the state that upper bounds the suboptimal cost in a neighborhood of the origin is often just assumed in suboptimal or real-time methods (e.g. [SMR99, LH09]), Algorithm 4 provides a constructive procedure to satisfy this condition.

Remark 8.11. The closed-loop system under the control law that would be obtained from directly applying Algorithm 4 to Problem $\mathbb{P}_N^\tau(x)$ without the Lyapunov constraint (8.2e) will remain within the feasible set $\mathcal{X}_N^\tau$ due to the feasibility guarantee provided by the robust MPC framework.

Remark 8.12. The re-optimization of the first tube center at every time step introduces additional feedback to the disturbance. A feasible and stable controller could, however, also be obtained by keeping the initial tube center fixed and only optimizing over the sequence of tube centers from $\bar{x}_1$ to $\bar{x}_N$.

Remark 8.13. The crucial property of recursive feasibility is guaranteed by all available robust MPC methods (see e.g. [BM99, MRRS00, LAR+09, RM09]) any of which could be used to derive a real-time MPC controller for the uncertain system (4.1). In order to allow for fast computation we use the tube based robust MPC approach for linear systems described in [MSR05] in this work.

After establishing feasibility and stability for the τ-RT control law in the regulation case, the following section extends the presented results to the more general case of robust tracking of piecewise constant reference signals.

8.4 Real-time Robust MPC for Tracking of Piecewise Constant References

Consider the task of tracking a piecewise constant sequence of steady-states, by steering the system state x to the target steady-state x_r. A target input u_r is associated with every target steady-state x_r fulfilling the steady-state condition $x_r = Ax_r + Bu_r$. The state and input constraints limit the set of feasible steady-states to $(x_r, u_r) \in \bar{\Theta}$, where $\bar{\Theta} \triangleq \{(x_r, u_r) \mid x_r \in \bar{\mathbb{X}}, u_r \in \bar{\mathbb{U}}, (A - I)x_r + Bu_r = 0\}$.

Remark 8.14. If tracking of an output signal is required, one can translate the output reference y_r into a state and input reference (x_r, u_r) using the following relation: $\left[\begin{smallmatrix} A-I & B \\ C & D \end{smallmatrix}\right]\left[\begin{smallmatrix} x_r \\ u_r \end{smallmatrix}\right] = \left[\begin{smallmatrix} 0 \\ y_r \end{smallmatrix}\right]$, where $y = Cx + Du$ is the output model.

In order to apply the proposed real-time method to reference tracking, we make use of the tracking approach presented in [LAAC08], which was included in a tube based robust MPC framework in [ALA+07], see also Section 5.4 for details on the tracking approach. The real-time robust MPC problem for reference tracking $\mathbb{P}_N^{\tau,tr}(x, x_r, u_r)$ is then given by:

Problem $\mathbb{P}_N^{\tau,tr}(x, x_r, u_r)$ (Real-time robust MPC for reference tracking)

$$V_N^{tr}(\mathbf{x}, \bar{\mathbf{u}}, \bar{x}_s, \bar{u}_s, x_r, u_r) = \sum_{i=0}^{N-1} l(\bar{x}_i - \bar{x}_s, \bar{u}_i - \bar{u}_s) + V_f(\bar{x}_N - \bar{x}_s) + V_o(\bar{x}_s - x_r, \bar{u}_s - u_r) \tag{8.8a}$$

$$\min_{\bar{\mathbf{x}}, \bar{\mathbf{u}}, \bar{x}_s, \bar{u}_s} V_N^{tr}(\bar{\mathbf{x}}, \bar{\mathbf{u}}, \bar{x}_s, \bar{u}_s, x_r, u_r) + V_f(x - \bar{x}_0) \tag{8.8b}$$

$$\begin{aligned}
\text{s.t.} \quad & (8.2\text{b}),\ (8.2\text{c}),\ (8.2\text{f}), && && (8.8\text{c})\\
& (\bar{x}_s, \bar{u}_s) && \in \bar{\Theta}\ , && (8.8\text{d})\\
& \bar{x}_N && \in \bar{\mathcal{X}}_f^{tr}(\bar{x}_s, \bar{u}_s)\ , && (8.8\text{e})\\
& V_N^{tr}(\bar{\mathbf{x}}, \bar{\mathbf{u}}, \bar{x}_s, \bar{u}_s, x_r, u_r) + V_f(x_{\text{nom}}^{\tau,tr} - \bar{x}_0) && \leq \pi_{\text{prev}}^{\tau,tr}\ , && (8.8\text{f})
\end{aligned}$$

where $(\bar{x}_s, \bar{u}_s)$ denotes the artificial steady-state, (x_r, u_r) is the desired steady-state and $V_o(\bar{x}_s - x_r, \bar{u}_s - u_r) = \|\bar{x}_s - x_r\|_{T_x}^2 + \|\bar{u}_s - u_r\|_{T_u}^2$ is the tracking offset cost, where T_x and T_u are positive definite matrices. $\bar{\mathcal{X}}_f^{tr}(\bar{x}_s, \bar{u}_s)$ is an invariant terminal target set for tracking as described in Section 5.4.1, see also [LAAC08, ALA+07].

Problem $\mathbb{P}_N^{\tau,tr}(x, x_r, u_r)$ implicitly defines the set of feasible control sequences $\mathcal{U}_N^{\tau,tr}(\bar{x}_0, \bar{x}_s, \bar{u}_s) = \{\bar{\mathbf{u}} \,|\, \exists\ \bar{\mathbf{x}} \text{ s.t. } (8.2\text{b}), (8.2\text{c}), (8.8\text{e}), (8.8\text{f}) \text{ hold}\}$, and feasible initial states $\mathcal{X}_N^{\tau,tr} = \{x \,|\, \exists\ \bar{x}_0 \in \mathcal{X}_0^{\tau}(x), (\bar{x}_s, \bar{u}_s) \in \bar{\Theta} \text{ s.t. } \mathcal{U}_N^{\tau,tr}(\bar{x}_0, \bar{x}_s, \bar{u}_s) \neq \emptyset\}$.

Assumption 8.15. It is assumed that $V_f(\bullet)$ and $\bar{\mathcal{X}}_f^{tr}(\bar{x}_s, \bar{u}_s)$ satisfy Assumption 5.9 with $\bar{\mathbb{X}}$ and $\bar{\mathbb{U}}$ replacing $\mathbb{X}$ and $\mathbb{U}$.

The real-time robust MPC problem for reference tracking $\mathbb{P}_N^{\tau,tr}(x, x_r, u_r)$ introduces the following components:

- An artificial steady state and input $(\bar{x}_s, \bar{u}_s)$, where the cost penalizes the deviation from the states and inputs to the artificial reference instead of the real reference. An offset term accounting for the deviation between the artificial and the real reference is added to the cost ensuring convergence to the desired steady-state (x_r, u_r).
- A terminal weight on the deviation between the terminal state and artificial reference as well as an extended terminal constraint on the terminal state and the artificial reference provide stability of the optimal MPC controller.

See Section 5.4 for more details on the tracking approach. The artificial steady-state and the control sequence are computed by solving a single optimization problem, which provides not only recursive feasibility of the warm-start solution but also permits a hard real-time guarantee and is the reason for the somewhat unusual tracking formulation.

A solution to Problem $\mathbb{P}_N^{\tau,tr}(x, x_r, u_r)$ is again computed in real-time by applying Algorithm 5 that is proposed in the following together with a primal feasible optimization routine, returning the variables $\bar{\mathbf{u}}^{\tau,tr}(x)$, $\bar{x}_0^{\tau,tr}(x)$, $\bar{x}_s^{\tau,tr}(x)$, $\bar{u}_s^{\tau,tr}(x)$. The robust τ-RT control law for tracking is then given in a receding horizon control fashion by:

$$\kappa^{\tau,tr}(x) = \bar{u}_0^{\tau,tr}(x) + K(x - \bar{x}_0^{\tau,tr}(x)) \ . \tag{8.9}$$

Definition 8.16 (Lyapunov constraint for tracking). For each $x(k)$, we again take $x_{\text{nom}}^{\tau,tr} = Ax(k-1) + B\kappa^{\tau,tr}(x(k-1))$ and

$$\begin{aligned} \pi_{\text{prev}}^{\tau,tr} =& V_N^{tr}(\bar{\mathbf{x}}^{\tau,tr}(x(k-1)), \bar{\mathbf{u}}^{\tau,tr}(x(k-1)), \bar{x}_s^{\tau,tr}(x(k-1)), \bar{u}_s^{\tau,tr}(x(k-1)), x_r, u_r) \\ &+ V_f(x(k-1) - \bar{x}_0^{\tau,tr}(k-1)) - \epsilon_k \frac{1}{2} l(x(k-1) - x_r, 0) \ , \end{aligned} \tag{8.10}$$

where $\epsilon_k \in (0, 1]$ is a small positive constant that has to be chosen at each sampling time such that the warm-start solution is strictly feasible, see Algorithm 5. The existence of such a constant at all times will be shown in Lemma 8.20.

Algorithm 5 Warm-start real-time robust procedure for tracking

Input: feasible control sequence $\bar{\mathbf{u}}^{\mathrm{in}}(x(k-1))$, tube center $\bar{x}_0^{\mathrm{in}}(x(k-1))$, corresponding state sequence $\bar{\mathbf{x}}^{\mathrm{in}}(x(k-1))$ and artificial steady-state $(\bar{x}_s^{\mathrm{in}}(x(k-1)), \bar{u}_s^{\mathrm{in}}(x(k-1)))$ for $x(k-1)$, current state measurement $x(k)$, desired steady-state (x_r, u_r) auxiliary control law $\kappa_f^{tr}(x) = \bar{u}_s^{\mathrm{in}}(x(k-1)) + K(x - \bar{x}_s^{\mathrm{in}}(x(k-1)))$, $A_K = A + BK$ and parameters $\epsilon_f, \epsilon_s > 0$

Output: τ-RT control sequence $\bar{\mathbf{u}}^{\tau,tr}(x(k))$, tube center $\bar{x}_0^{\tau,tr}(x(k))$ and artificial steady-state $(\bar{x}_s^{\tau,tr}(x(k)), \bar{u}_s^{\tau,tr}(x(k)))$

```
Warm-start: $\bar{x}_0^{\mathrm{ws}} = \bar{x}_1^{\mathrm{in}}(x(k-1))$
if $\|\bar{x}_0^{\mathrm{ws}} - \bar{x}_s^{\mathrm{in}}(x(k-1))\|_P \geq \epsilon_s$ then
        $\bar{\mathbf{u}}^{\mathrm{ws}} = \Big[\bar{u}_1^{\mathrm{in}}(x(k-1)), \dots, \bar{u}_{N-1}^{\mathrm{in}}(x(k-1)), \kappa_f^{tr}(\bar{x}_N^{\mathrm{in}}(x(k-1)))\Big]$,
else Generate $\bar{\mathbf{u}}^{\mathrm{ws}}$ from control law $\kappa_f^{tr}(x)$,
        i.e. $\bar{u}_i^{\mathrm{ws}} = \bar{u}_s^{\mathrm{in}}(x(k-1)) + KA_K^i(\bar{x}_0^{\mathrm{ws}} - \bar{x}_s^{\mathrm{in}}(x(k-1)))$ for $i = 0, \dots, N-1$
end if
    $\alpha_{\min} = \min_{\alpha \in [0,1]} \alpha$
            s.t. $\bar{x}_N^{\mathrm{ws}} \in \bar{\mathcal{X}}_f^{tr}(\bar{x}_s, \bar{u}_s)$
                $V_N^{tr}(\bar{\mathbf{x}}^{\mathrm{ws}}, \bar{\mathbf{u}}^{\mathrm{ws}}, \bar{x}_s, \bar{u}_s, x_r, u_r) + V_f(x_{\mathrm{nom}}^{\tau,tr} - \bar{x}_0) \leq \pi_{\mathrm{prev}}^{\tau}$ for $\epsilon_k = \frac{1}{2}(1-\alpha)^2$
                $\bar{x}_s = \alpha \bar{x}_s^{\mathrm{in}}(x(k-1)) + (1-\alpha)x_r,\ \bar{u}_s = \alpha \bar{u}_s^{\mathrm{in}}(x(k-1)) + (1-\alpha)u_r$
$\bar{x}_s^{\mathrm{ws}} = \alpha_{\min}\bar{x}_s^{\mathrm{in}}(x(k-1)) + (1-\alpha_{\min})x_r,\ \bar{u}_s^{\mathrm{ws}} = \alpha_{\min}\bar{u}_s^{\mathrm{in}}(x(k-1)) + (1-\alpha_{\min})u_r$
Choose $\epsilon_k$ such that warm-start strictly satisfies (8.8f)
$\tilde{\mathbf{u}} = \mathbf{u}^{\mathrm{ws}},\ \tilde{x}_0 = \bar{x}_0^{\mathrm{ws}},\ \tilde{x}_s = \bar{x}_s^{\mathrm{ws}},\ \tilde{u}_s = \bar{u}_s^{\mathrm{ws}}$
while clock$< \tau$ do
    improve $\tilde{\mathbf{u}}, \tilde{x}_0, \tilde{x}_s, \tilde{u}_s$ in one primal feasible optimization step
end while
if $\|x - x_r\|_P \leq \epsilon_f$ and $V_N^{tr}(\tilde{\mathbf{x}}, \tilde{\mathbf{u}}, \tilde{x}_s, \tilde{u}_s, x_r, u_r) + V_f(x(k) - \tilde{x}_0) > V_f(x(k) - x_r)$ then
    $\tilde{\mathbf{u}} = [u_r, \dots, u_r], \tilde{x}_0 = x_r, \tilde{x}_s = x_r, \tilde{u}_s = u_r$
end if
$\bar{\mathbf{u}}^{\tau,tr}(x(k)) = \tilde{\mathbf{u}},\ \bar{x}_0^{\tau,tr}(x(k)) = \tilde{x}_0,\ \bar{x}_s^{\tau,tr}(x(k)) = \tilde{x}_s,\ \bar{u}_s^{\tau,tr}(x(k))) = \tilde{u}_s$
```

Remark 8.17. Note that the index τ again denotes the dependence of variables and functions on the solution computed at the previous time step and the computation time, see also Remark 8.3.

The tracking formulation in $\mathbb{P}_N^{\tau,tr}(x, x_r, u_r)$ is designed to regulate the system state to the artificial steady-state $(\bar{x}_s, \bar{u}_s)$, which is simultaneously steered to the target steady-state. Convergence of this scheme to (x_r, u_r) using the optimal control law was shown in [LAAC08, ALA$^+$07] using the fact that $\bar{x}_s \neq x_r$ cannot be the optimal solution. In order to achieve convergence in a real-time setting, the Lyapunov constraint (8.8f) is introduced. In contrast to the regulation case, feasibility of this constraint for some strictly positive ϵ_k using the shifted sequence as a warm-start does not follow directly. Algorithm 5 therefore differs from the real-time procedure in Algorithm 4 proposed for the regulation case in the following aspects:

- *Initial warm-start (Step 2-4):* A warm-start sequence is generated either from the shifted initial solution together with an auxiliary control law, which was shown to be feasible in [LAAC08, ALA$^+$07], or, if the initial tube center is very close to $\bar{x}_s^{\text{in}}$, by applying the auxiliary control law, which provides the optimal input sequence for a given artificial steady-state $(\bar{x}_s^{\text{in}}, \bar{u}_s^{\text{in}})$.
- *Computation of α (Step 5):* This step is crucial for providing feasibility of the warm-start. Initially, a warm-start is generated that regulates the system to the artificial steady-state $(\bar{x}_s^{\text{in}}, \bar{u}_s^{\text{in}})$. If this warm-start strategy is, however, applied recursively without changing $(\bar{x}_s^{\text{in}}, \bar{u}_s^{\text{in}})$, the Lyapunov constraint will no longer be feasible at some point, since the tube centers converge to $\bar{x}_s^{\text{in}}$ instead of x_r. This motivates the minimization of α in Step 5. If the optimal solution is $\alpha_{\min} < 1$, the artificial steady-state $(\bar{x}_s^{\text{ws}}, \bar{u}_s^{\text{ws}})$ is improved by moving it from $(\bar{x}_s^{\text{in}}, \bar{u}_s^{\text{in}})$ towards (x_r, u_r) while guaranteeing satisfaction of the state and input constraints as well as the Lyapunov decrease constraint. Note that $\alpha = 1$ is always a feasible solution to this optimization problem. While the initial tube center is not too close to the artificial steady-state, the choice of $\alpha = 1$, i.e. keeping the artificial steady-state at $(\bar{x}_s^{\text{in}}, \bar{u}_s^{\text{in}})$ and using the standard warm-start, provides a sufficient decrease in the cost function to satisfy the Lyapunov constraint for some $\epsilon_k > 0$. If, in contrast, the first tube center is close to $\bar{x}_s^{\text{in}}$, the minimization in Step 5 will provide $\alpha_{\min} < 1$, ensuring feasibility of the warm-start solution. These facts are shown in Lemma 8.20.

Note that Steps 12-14 of Algorithm 5 implement the same strategy as proposed in Steps 6-8 of Algorithm 4, but for regulation to a non-zero steady-state (x_r, u_r) instead of the origin, ensuring that the real-time cost can be upper bounded by a $\mathcal{K}_\infty$-class function of $\|x - x_r\|$.

Remark 8.18. The minimization of α in Step 5 of Algorithm 5 amounts to the solution of a 1-dimensional convex optimization problem that can be rewritten in the following form by using $\delta = 1-\alpha$:

$$\max\{\delta \in [0,1] \mid \delta^2 + a_1\delta + b_1 \leq c_1, \delta^2 + a_2\delta + b_2 \leq c_2\} \ . \tag{8.11}$$

The solution can be obtained by computing the maximum separately for each constraint, where for each of them an analytical solution can be derived, and then taking the smaller of the two values.

Feasibility of the warm-start provided by Algorithm 5 for Problem $\mathbb{P}_N^{\tau,tr}(x^+, x_r, u_r)$ will be proven in detail in Lemma 8.20. This is key to showing convergence of the closed-loop system under the proposed τ-RT control law for tracking to an RPI set around x_r in Theorem 8.22. We first state a lemma that is required for this result.

Lemma 8.19 *Consider Problem $\mathbb{P}_N^{\tau,tr}(x, x_r, u_r)$. Let (x_r, u_r) be a reference steady-state, $(\tilde{x}_s, \tilde{u}_s)$ a steady-state and $\tilde{x}_0 \in \bar{\mathcal{X}}_f^{tr}(\tilde{x}_s, \tilde{u}_s)$ a first tube center. Let $\tilde{\mathbf{u}}, \tilde{\mathbf{x}}$ be the input and state sequence generated by applying the auxiliary control law $\kappa_f^{tr}(x) = \tilde{u}_s + K(x - \tilde{x}_s)$ starting from $\tilde{x}_0$, i.e. $\tilde{u}_i = K(A+BK)^i(\tilde{x}_0 - \tilde{x}_s) + \tilde{u}_s$, for $i = 0, \ldots, N-1$. Denote $\tilde{x}_{s,\alpha} = \alpha\tilde{x}_s + (1-\alpha)x_r$, $\tilde{u}_{s,\alpha} = \alpha\tilde{u}_s + (1-\alpha)u_r$. There exists an $\alpha < 1$ such that if $\|\tilde{x}_0 - \tilde{x}_s\|_P \leq (1-\alpha)\|\tilde{x}_s - x_r\|_P$, then*

$$V_N^{tr}(\tilde{\mathbf{x}}, \tilde{\mathbf{u}}, \tilde{x}_{s,\alpha}, \tilde{u}_{s,\alpha}, x_r, u_r) \leq V_N^{tr}(\tilde{\mathbf{x}}, \tilde{\mathbf{u}}, \tilde{x}_s, \tilde{u}_s, x_r, u_r) - (1-\alpha)^2\|\tilde{x}_s - x_r\|_P^2 \ . \tag{8.12}$$

Proof. We denote $A_K = A + BK$. From the use of the auxiliary control law we obtain

$$\begin{aligned} l(\tilde{x}_i - \tilde{x}_s, \tilde{u}_i - \tilde{u}_s) =& \|A_K^i(\tilde{x}_0 - \tilde{x}_s)\|_Q^2 + \|KA_K^i(\tilde{x}_0 - \tilde{x}_s)\|_R^2 \ , \\ l(\tilde{x}_i - \tilde{x}_{s,\alpha}, \tilde{u}_i - \tilde{u}_{s,\alpha}) =& \|A_K^i(\tilde{x}_0 - \tilde{x}_s) + (1-\alpha)(\tilde{x}_s - x_r)\|_Q^2 \\ &+ \|KA_K^i(\tilde{x}_0 - \tilde{x}_s) + (1-\alpha)(\tilde{u}_s - u_r)\|_R^2 \ . \end{aligned}$$

By Assumption 8.15, $V_f(x - \bar{x}_s)$ is a Lyapunov function and $P \succeq Q + K^TRK$, therefore $\|A_K^i x\|_Q \leq \|A_K^i x\|_P \leq \|x\|_P$ and $\|A_K^i x\|_{K^TRK} \leq \|A_K^i x\|_P \leq \|x\|_P$, which together with $\|\tilde{x}_0 - \tilde{x}_s\|_P \leq (1-\alpha)\|\tilde{x}_s - x_r\|_P$ provides that

$$\begin{aligned} & l(\tilde{x}_i - \tilde{x}_{s,\alpha}, \tilde{u}_i - \tilde{u}_{s,\alpha}) - l(\tilde{x}_i - \tilde{x}_s, \tilde{u}_i - \tilde{u}_s) \\ &= \ 2(1-\alpha)(A_K^i(\tilde{x}_0 - \tilde{x}_s))^TQ(\tilde{x}_s - x_r) + (1-\alpha)^2\|\tilde{x}_s - x_r\|_Q^2 \\ &\quad + 2(1-\alpha)(KA_K^i(\tilde{x}_0 - \tilde{x}_s))^TR(\tilde{u}_s - u_r) + (1-\alpha)^2\|\tilde{u}_s - u_r\|_R^2 \\ &\leq \ 2(1-\alpha)\|A_K^i(\tilde{x}_0 - \tilde{x}_s)\|_Q\|\tilde{x}_s - x_r\|_Q + (1-\alpha)^2\|\tilde{x}_s - x_r\|_Q^2 \\ &\quad + 2(1-\alpha)\|A_K^i(\tilde{x}_0 - \tilde{x}_s)\|_{K^TRK}\|\tilde{u}_s - u_r\|_R + (1-\alpha)^2\|\tilde{u}_s - u_r\|_R^2 \\ &\leq \ 2(1-\alpha)^2\|\tilde{x}_s - x_r\|_P^2 + (1-\alpha)^2\|\tilde{x}_s - x_r\|_P^2 \end{aligned}$$

$$+ 2(1-\alpha)^2\|\tilde{x}_s - x_r\|_P\|\tilde{u}_s - u_r\|_R + (1-\alpha)^2\|\tilde{u}_s - u_r\|_R^2$$

and similarly

$$V_f(\tilde{x}_N - \tilde{x}_{s,\alpha}) - V_f(\tilde{x}_N - \tilde{x}_s) \leq 3(1-\alpha)^2\|\tilde{x}_s - x_r\|_P^2 \ .$$

In the following we denote $\Delta\tilde{x}_s \triangleq \|\tilde{x}_s - x_r\|_P$ and $\Delta\tilde{u}_s \triangleq \|\tilde{u}_s - u_r\|_R$. From convexity of $V_o(\cdot,\cdot)$ we obtain

$$V_o(\tilde{x}_{s,\alpha} - x_r, \tilde{u}_{s,\alpha} - u_r) = V_o(\alpha(\tilde{x}_s - x_r), \alpha(\tilde{u}_s - u_r)) \leq \alpha V_o(\tilde{x}_s - x_r, \tilde{u}_s - u_r)$$

and therefore

$$\begin{aligned}
&V_N^{tr}(\tilde{\mathbf{x}}_\alpha, \tilde{\mathbf{u}}_\alpha, \tilde{x}_{s,\alpha}, \tilde{u}_{s,\alpha}, x_r, u_r) - V_N^{tr}(\tilde{\mathbf{x}}, \tilde{\mathbf{u}}, \tilde{x}_s, \tilde{u}_s, x_r, u_r) + (1-\alpha)^2\|\tilde{x}_s - x_r\|_P^2 \\
&= \sum_{i=0}^{N-1} l(\tilde{x}_i - \tilde{x}_{s,\alpha}, \tilde{u}_i - \tilde{u}_{s,\alpha}) - l(\tilde{x}_i - \tilde{x}_s, \tilde{u}_i - \tilde{u}_s) + V_f(\tilde{x}_N - \tilde{x}_{s,\alpha}) - V_f(\tilde{x}_N - \tilde{x}_s) \\
&\quad + V_o(\tilde{x}_{s,\alpha} - x_r, \tilde{u}_{s,\alpha} - u_r) - V_o(\tilde{x}_s - x_r, \tilde{u}_s - u_r) + (1-\alpha)^2\|\tilde{x}_s - x_r\|_P^2 \\
&\leq 3(N+1)(1-\alpha)^2\Delta\tilde{x}_s^2 + 2N(1-\alpha)^2\Delta\tilde{x}_s\Delta\tilde{u}_s + N(1-\alpha)^2\Delta\tilde{u}_s^2 \\
&\quad + (1-\alpha)^2\Delta\tilde{x}_s^2 - (1-\alpha)V_o(\tilde{x}_s - x_r, \tilde{u}_s - u_r) \\
&\leq (1-\alpha)[(3N+4)\Delta\tilde{x}_s^2(1-\alpha) - V_o(\tilde{x}_s - x_r, \tilde{u}_s - u_r) \\
&\quad + 2N\Delta\tilde{x}_s\Delta\tilde{u}_s(1-\alpha) + N\Delta\tilde{u}_s^2(1-\alpha)] \leq 0 \ ,
\end{aligned}$$

which is satisfied for

$$\frac{(3N+4)\Delta\tilde{x}_s^2 + 2N\Delta\tilde{x}_s\Delta\tilde{u}_s + N\Delta\tilde{u}_s^2 - V_o(\tilde{x}_s - x_r, \tilde{u}_s - u_r)}{(3N+4)\Delta\tilde{x}_s^2 + 2N\Delta\tilde{x}_s\Delta\tilde{u}_s + N\Delta\tilde{u}_s^2} \leq \alpha < 1 \ ,$$

proving the result. ■

Lemma 8.19 shows that if the initial tube center $\bar{x}_0$ is very close to the artificial steady-state $\bar{x}_s$, then we can move the artificial steady-state towards x_r, while still providing a decrease in the cost using the auxiliary control law, which allows us to prove feasibility of the warm-start solution provided by Algorithm 5 in the following theorem.

Theorem 8.20 (Feasibility of the warm-start for tracking) *Let $\epsilon_s, \epsilon_f > 0$ be positive constants such that $\mathcal{E}_f \triangleq \{x \mid \|x - \bar{x}_s\|_P \leq \epsilon_s\} \subseteq \bar{\mathcal{X}}_f^{tr}(\bar{x}_s, \bar{u}_s)$ for all $(\bar{x}_s, \bar{u}_s) \in \Theta_s$ and $\mathcal{S} \triangleq \{x \mid \|x - x_r\|_P \leq \epsilon_f\} \subseteq x_r \oplus \mathcal{Z}_\mathcal{W}$. The warm-start solution provided by Algorithm 5 is feasible for $\mathbb{P}_N^{\tau,tr}(x^+, x_r, u_r)$, where $x^+ \in Ax + B\kappa^{\tau,tr}(x) \oplus \mathcal{W}$, i.e. $\bar{x}_0^{ws} \in \mathcal{X}_0^\tau(x^+)$, $(\bar{x}_s^{ws}, \bar{u}_s^{ws}) \in \bar{\Theta}$, $\bar{\mathbf{u}}^{ws} \in \mathcal{U}_N^{\tau,tr}(\bar{x}_0^{ws}, \bar{x}_s^{ws}, \bar{u}_s^{ws})$.*

Proof. Feasibility of $\bar{x}_0^{\text{ws}}$ follows directly from the results in [MSR05]. Since $\bar{\Theta}$ is a convex set, feasibility of (x_r, u_r) and $(\bar{x}_s^{\text{in}}(x), \bar{u}_s^{\text{in}}(x))$ imply $(\bar{x}_s^{\text{ws}}, \bar{u}_s^{\text{ws}}) \in \bar{\Theta}$ for $\alpha \in [0,1]$.

For $\alpha = 1$, $\|\bar{x}_0^{\text{ws}} - \bar{x}_s^{\text{in}}(x(k-1))\|_P \geq \epsilon_s$, feasibility of the warm-start for $\mathbb{P}_N^{\tau,tr}(x^+, x_r, u_r)$ without the Lyapunov constraint (8.8f) is directly provided by feasibility of the input to Algorithm 5 and invariance of the terminal set, see also [LAAC08, ALA+07], for $\|\bar{x}_0^{\text{ws}} - \bar{x}_s^{\text{in}}(x(k-1))\|_P < \epsilon_s$ by feasibility of the auxiliary control law. Since the minimization of α in Step 5 enforces the terminal constraint, satisfaction of the state and input constraints is maintained. The important condition to prove is hence the Lyapunov constraint (8.8f). If $\alpha_{\min} < 1$ in Step 5, the Lyapunov constraint is satisfied by construction with $\epsilon_k = \frac{1}{2}(1 - \alpha_{\min})^2 > 0$ and the critical issue is therefore $\alpha_{\min} = 1$. We first point out why $\alpha_{\min}$=1 may be critical in order to then prove that by the use of Step 5 feasibility of the Lyapunov decrease constraint is provided.

The dependence of the initialization on x and the warm-start on x^+ is neglected for ease of notation, since it is clear from the context. The following facts are used:

$$\begin{aligned} x_{\text{nom}}^{\tau,tr} &= Ax - A\bar{x}_0^{\text{in}} + A\bar{x}_0^{\text{in}} + B\bar{u}_0^{\text{in}} + BK(x - \bar{x}_0^{\text{in}}) \\ &= \bar{x}_1^{\text{in}} + (A + BK)(x - \bar{x}_0^{\text{in}}) = \bar{x}_0^{\text{ws}} + (A + BK)(x - \bar{x}_0^{\text{in}}) \ . \end{aligned} \tag{8.13}$$

By Assumption 8.15 $V_f(\bullet)$ is a Lyapunov function satisfying condition A2 in Assumption 5.3, therefore

$$\begin{aligned} &V_f(x_{\text{nom}}^{\tau,tr} - \bar{x}_0^{\text{ws}}) - V_f(x - \bar{x}_0^{\text{in}}) \\ &= \|(A + BK)(x - \bar{x}_0^{\text{in}})\|_P^2 - \|x - \bar{x}_0^{\text{in}}\|_P^2 \leq -\|x - \bar{x}_0^{\text{in}}\|_Q^2 \ . \end{aligned} \tag{8.14}$$

A relationship that will be used in several places is $\frac{1}{2}\|x + y\|_Q^2 \leq \|x\|_Q^2 + \|y\|_Q^2$, which is provided by convexity of $\|\bullet\|_Q^2$.

Let $\bar{\mathbf{u}}^{\text{shift}}$ denote the shifted sequence given by the initial warm-start in Step 2 of Algorithm 5 and $\bar{\mathbf{x}}^{\text{shift}}$ the corresponding state sequence starting from $\bar{x}_0^{\text{shift}} = \bar{x}_1^{\text{in}}$. If $\alpha_{\min}$=1, the use of the shifted sequence provides the following decrease in the cost, which can be obtained from a direct comparison of the sequences and using standard arguments in MPC (see also [ALA+07]):

$$V_N^{tr}(\bar{\mathbf{x}}^{\text{shift}}, \bar{\mathbf{u}}^{\text{shift}}, \bar{x}_s^{\text{in}}, \bar{u}_s^{\text{in}}, x_r, u_r) \leq V_N^{tr}(\bar{\mathbf{x}}^{\text{in}}, \bar{\mathbf{u}}^{\text{in}}, \bar{x}_s^{\text{in}}, \bar{u}_s^{\text{in}}, x_r, u_r) - l(\bar{x}_0^{\text{in}} - \bar{x}_s^{\text{in}}, \bar{u}_0^{\text{in}} - \bar{u}_s^{\text{in}}) \ .$$

By using (8.14) we then obtain

$$V_N^{tr}(\bar{\mathbf{x}}^{\text{shift}}, \bar{\mathbf{u}}^{\text{shift}}, \bar{x}_s^{\text{in}}, \bar{u}_s^{\text{in}}, x_r, u_r) + V_f(x_{\text{nom}}^{\tau,tr} - \bar{x}_0^{\text{ws}}) \tag{8.15a}$$

$$- V_N^{tr}(\bar{\mathbf{x}}^{\text{in}}, \bar{\mathbf{u}}^{\text{in}}, \bar{x}_s^{\text{in}}, \bar{u}_s^{\text{in}}, x_r, u_r) - V_f(x - \bar{x}_0^{\text{in}}) \tag{8.15b}$$

$$\leq -l(\bar{x}_0^{\text{in}} - \bar{x}_s^{\text{in}}, \bar{u}_0^{\text{in}} - \bar{u}_s^{\text{in}}) - \|x - \bar{x}_0^{\text{in}}\|_Q^2 \tag{8.15c}$$

$$= -\|\bar{x}_0^{\text{in}} - \bar{x}_s^{\text{in}}\|_Q^2 - \|\bar{u}_0^{\text{in}} - \bar{u}_s^{\text{in}}\|_R^2 - \|x - \bar{x}_0^{\text{in}}\|_Q^2 \ . \tag{8.15d}$$

The cost therefore decreases as a function of $\|\bar{x}_0^{\text{in}} - \bar{x}_s^{\text{in}}\|_Q^2 + \|\bar{u}_0^{\text{in}} - \bar{u}_s^{\text{in}}\|_R^2$ instead of $\|\bar{x}_0^{\text{in}} - x_r\|_Q^2$. In order to bound the decrease by $\epsilon_k\|x - x_r\|_Q^2$, as required by the Lyapunov constraint, we would have to choose

$$\epsilon_k = \min\left(\frac{\|\bar{x}_0^{\text{in}} - \bar{x}_s^{\text{in}}\|_Q^2 + \|\bar{u}_0^{\text{in}} - \bar{u}_s^{\text{in}}\|_R^2}{\|\bar{x}_0^{\text{in}} - x_r\|_Q^2}, 1\right) \quad , \tag{8.16}$$

in order to obtain

$$\begin{aligned}
-\|\bar{x}_0^{\text{in}} - \bar{x}_s^{\text{in}}\|_Q^2 - \|\bar{u}_0^{\text{in}} - \bar{u}_s^{\text{in}}\|_R^2 - \|x - \bar{x}_0^{\text{in}}\|_Q^2 &\leq -\epsilon_k\|\bar{x}_0^{\text{in}} - x_r\|_Q^2 - \|x - \bar{x}_0^{\text{in}}\|_Q^2 \\
&\leq -\epsilon_k\|\bar{x}_0^{\text{in}} - x_r\|_Q^2 - \epsilon_k\|x - \bar{x}_0^{\text{in}}\|_Q^2 \\
&\leq -\frac{1}{2}\epsilon_k\|x - x_r\|_Q^2 \ .
\end{aligned}$$

It can, however, be seen from the definition of ϵ_k in (8.16) that if $\|\bar{x}_0^{\text{in}} - \bar{x}_s^{\text{in}}\|_Q^2$ and/or $\|\bar{u}_0^{\text{in}} - \bar{u}_s^{\text{in}}\|_R^2$ go to zero while $\|\bar{x}_0^{\text{in}} - x_r\|_Q^2$ does not, ϵ_k will go to zero. This means that the tube center would converge to $\bar{x}_s^{\text{in}}$ instead of x_r, which is not the desired behavior. In order to ensure convergence to x_r, the Lyapunov constraint therefore requires a strict decrease with $\epsilon_k > 0 \ \forall k \in \mathbb{N}$.

The goal of this proof is to show that there always exists an $\epsilon_k > 0$, such that the warm-start satisfies the Lyapunov constraint in (8.8f). Recall that if $\alpha_{\min} < 1$ is chosen in Step 5, then $\epsilon_k = \frac{1}{2}(1-\alpha_{\min})^2 > 0$. We therefore prove this result for the following two cases: If $\|\bar{x}_0^{\text{ws}} - \bar{x}_s^{\text{in}}\|_P \geq \epsilon_s$ and $\alpha_{\min} = 1$ is the optimal solution in Step 5 it will be shown that $\epsilon_k > 0$ is satisfied by the warm-start. If $\|\bar{x}_0^{\text{ws}} - \bar{x}_s^{\text{in}}\|_P < \epsilon_s$, it will be shown that either $\|\bar{x}_0^{\text{ws}} - \bar{x}_s^{\text{in}}\|_P \geq (1-\alpha)\|\bar{x}_s^{\text{in}} - x_r\|_P$ for some $\alpha < 1$ and $\epsilon_k > 0$ or the optimization of α in Step 5 always provides $\alpha_{\min} < 1$.

Case 1: $\|\bar{x}_0^{\text{ws}} - \bar{x}_s^{\text{in}}\|_P \geq \epsilon_s$

$$\begin{aligned}
\epsilon_s &\leq \|\bar{x}_0^{\text{ws}} - \bar{x}_s^{\text{in}}\|_P \leq \|A(\bar{x}_0^{\text{in}} - \bar{x}_s^{\text{in}}) + B(\bar{u}_0^{\text{in}} - \bar{u}_s^{\text{in}})\|_P \\
&\leq \|A(\bar{x}_0^{\text{in}} - \bar{x}_s^{\text{in}})\|_P + \|B(\bar{u}_0^{\text{in}} - \bar{u}_s^{\text{in}})\|_P \leq \|A\|_P\|\bar{x}_0^{\text{in}} - \bar{x}_s^{\text{in}}\|_2 + \|B\|_P\|\bar{u}_0^{\text{in}} - \bar{u}_s^{\text{in}}\|_2
\end{aligned}$$

and therefore

$$\begin{aligned}
&\|B\|_P\|\bar{u}_0^{\text{in}} - \bar{u}_s^{\text{in}}\|_2 \leq k\epsilon_s \Rightarrow \|A\|_P\|\bar{x}_0^{\text{in}} - \bar{x}_s^{\text{in}}\|_2 \geq (1-k)\epsilon_s \Rightarrow \|\bar{x}_0^{\text{in}} - \bar{x}_s^{\text{in}}\|_Q \geq \frac{(1-k)\epsilon_s}{c_Q\|A\|_P} \\
&\|A\|_P\|\bar{x}_0^{\text{in}} - \bar{x}_s^{\text{in}}\|_2 \leq k\epsilon_s \Rightarrow \|B\|_P\|\bar{u}_0^{\text{in}} - \bar{u}_s^{\text{in}}\|_2 \geq (1-k)\epsilon_s \Rightarrow \|\bar{u}_0^{\text{in}} - \bar{u}_s^{\text{in}}\|_R \geq \frac{(1-k)\epsilon_s}{c_R\|B\|_P} \quad ,
\end{aligned}$$

where $k \in (0,1)$ and $c_Q, c_R \geq 1$ are such that $c_Q^2 Q \succeq I$, $c_R^2 R \succeq I$.

By defining $\epsilon_{s,\min} = \min\left(\frac{(1-k)\epsilon_s}{c_Q\|A\|_P}, \frac{(1-k)\epsilon_s}{c_R\|B\|_P}\right)$, this implies that $\|\bar{x}_0^{\text{in}} - \bar{x}_s^{\text{in}}\|_Q^2 + \|\bar{u}_0^{\text{in}} - \bar{u}_s^{\text{in}}\|_R^2 \geq \epsilon_{s,\min}^2$ and thus

$$\epsilon_k = \frac{\|\bar{x}_0^{\text{in}} - \bar{x}_s^{\text{in}}\|_Q^2 + \|\bar{u}_0^{\text{in}} - \bar{u}_s^{\text{in}}\|_R^2}{\|\bar{x}_0^{\text{in}} - x_r\|_Q^2} \geq \frac{\epsilon_{s,\min}^2}{\max_{\bar{x}_0^{\text{in}} \in \mathbb{X}}\|\bar{x}_0^{\text{in}} - x_r\|_Q^2} > 0 \ .$$

Case 2: $\|\bar{x}_0^{\text{ws}} - \bar{x}_s^{\text{in}}\|_P < \epsilon_s$
In this case, the warm-start sequence is intitialized using the auxiliary control law $\kappa_f^{tr}(x) = \bar{u}_s^{\text{in}} + K(x - \bar{x}_s^{\text{in}})$ starting from $\bar{x}_0^{\text{ws}}$. We will show that either $\|\bar{x}_0^{\text{ws}} - \bar{x}_s^{\text{in}}\|_P \geq (1-\alpha)\|\bar{x}_s^{\text{in}} - x_r\|_P$ for some $\alpha < 1$, in which case $\epsilon_k > 0$, or if $\|\bar{x}_0^{\text{ws}} - \bar{x}_s^{\text{in}}\|_P \leq (1-\alpha)\|\bar{x}_s^{\text{in}} - x_r\|_P$, then $\alpha_{\min} < 1$ in Step 5.
Case 2a: $\|\bar{x}_0^{\text{ws}} - \bar{x}_s^{\text{in}}\|_P \geq (1-\alpha)\|\bar{x}_s^{\text{in}} - x_r\|_P$

$$(1-\alpha)^2\|\bar{x}_s^{\text{in}} - x_r\|_P^2 \leq \|\bar{x}_0^{\text{ws}} - \bar{x}_s^{\text{in}}\|_P^2 \leq c_P\|\bar{x}_0^{\text{ws}} - \bar{x}_s^{\text{in}}\|_Q^2$$

where $c_P \geq 1$ is such that $c_P Q \succeq P$ and therefore

$$\begin{aligned}\|\bar{x}_0^{\text{ws}} - \bar{x}_s^{\text{in}}\|_Q^2 &\geq \frac{1}{2}\|\bar{x}_0^{\text{ws}} - \bar{x}_s^{\text{in}}\|_Q^2 + \frac{1}{2c_P}(1-\alpha)^2\|\bar{x}_s^{\text{in}} - x_r\|_P^2 \\ &\geq \frac{1}{2c_P}(1-\alpha)^2\|\bar{x}_0^{\text{ws}} - \bar{x}_s^{\text{in}}\|_Q^2 + \frac{1}{2c_P}(1-\alpha)^2\|\bar{x}_s^{\text{in}} - x_r\|_Q^2 \\ &\geq \frac{1}{4c_P}(1-\alpha)^2\|\bar{x}_0^{\text{ws}} - x_r\|_Q^2 \ .\end{aligned}$$

From (8.15), (8.16) it then follows that $\epsilon_k \geq \frac{1}{4c_P}(1-\alpha)^2 > 0$.
Case 2b: $\|\bar{x}_0^{\text{ws}} - \bar{x}_s^{\text{in}}\|_P < (1-\alpha)\|\bar{x}_s^{\text{in}} - x_r\|_P$
If the artificial steady-state is $(\bar{x}_s^{\text{in}}, \bar{u}_s^{\text{in}})$, the optimal sequence to regulate the system to this steady-state starting from $\bar{x}_0^{\text{ws}} \in \bar{\mathcal{X}}_f^{tr}(\bar{x}_s^{\text{in}}, \bar{u}_s^{\text{in}})$ is by applying the control law $\kappa_f^{tr}(x) = \bar{u}_s^{\text{in}} + K(x - \bar{x}_s^{\text{in}})$. The corresponding input and state sequences are denoted by $\bar{\mathbf{u}}^{\circ}, \bar{\mathbf{x}}^{\circ}$. By sub-optimality of $\bar{\mathbf{u}}^{\text{shift}}$, we obtain

$$V_N^{tr}(\bar{\mathbf{x}}^{\circ}, \bar{\mathbf{x}}^{\circ}, \bar{x}_s^{\text{in}}, \bar{u}_s^{\text{in}}, x_r, u_r) \leq V_N^{tr}(\bar{\mathbf{x}}^{\text{shift}}, \bar{\mathbf{u}}^{\text{shift}}, \bar{x}_s^{\text{in}}, \bar{u}_s^{\text{in}}, x_r, u_r) \ . \tag{8.17}$$

Since $\|\bar{x}_0^{\text{ws}} - \bar{x}_s^{\text{in}}\|_P < (1-\alpha)\|\bar{x}_s^{\text{in}} - x_r\|_P$, we can use result (8.12) of Lemma 8.19 in order to show that there exists an $\alpha < 1$ such that the warm-start satisfies

$$\begin{aligned}&V_N^{tr}(\bar{\mathbf{x}}^{\text{ws}}, \bar{\mathbf{u}}^{\text{ws}}, \bar{x}_s^{\text{ws}}, \bar{x}_s^{\text{ws}}, x_r, u_r) && \text{(8.18a)}\\ &\leq V_N^{tr}(\bar{\mathbf{x}}^{\circ}, \bar{\mathbf{x}}^{\circ}, \bar{x}_s^{\text{in}}, \bar{u}_s^{\text{in}}, x_r, u_r) - (1-\alpha)^2\|\bar{x}_s^{\text{in}} - x_r\|_P^2 && \text{(8.18b)}\\ &\leq V_N^{tr}(\bar{\mathbf{x}}^{\text{shift}}, \bar{\mathbf{u}}^{\text{shift}}, \bar{x}_s^{\text{in}}, \bar{u}_s^{\text{in}}, x_r, u_r) - (1-\alpha)\|\bar{x}_s^{\text{in}} - x_r\|_P^2 \ , && \text{(8.18c)}\end{aligned}$$

where the last step uses (8.17). Finally, from convexity of $\|\cdot\|_Q^2$, (8.15) and (8.18)

$$\begin{aligned}&V_N^{tr}(\bar{\mathbf{x}}^{\text{ws}}, \bar{\mathbf{u}}^{\text{ws}}, \bar{x}_s^{\text{ws}}, \bar{u}_s^{\text{ws}}, x_r, u_r) + V_f(x_{\text{nom}}^{\tau,tr} - \bar{x}_0^{\text{ws}}) - V_N^{tr}(\bar{\mathbf{x}}^{\text{in}}, \bar{\mathbf{u}}^{\text{in}}, \bar{x}_s^{\text{in}}, \bar{u}_s^{\text{in}}, x_r, u_r) - V_f(x - \bar{x}_0^{\text{in}})\\ &\leq -\|\bar{x}_0^{\text{in}} - \bar{x}_s^{\text{in}}\|_Q^2 - \|x - \bar{x}_0^{\text{in}}\|_Q^2 - (1-\alpha)^2\|\bar{x}_s^{\text{in}} - x_r\|_P^2\\ &\leq -\frac{1}{2}\|x - \bar{x}_s^{\text{in}}\|_Q^2 - (1-\alpha)^2\|\bar{x}_s^{\text{in}} - x_r\|_P^2\\ &\leq -(1-\alpha)^2\frac{1}{2}(\|x - \bar{x}_s^{\text{in}}\|_Q^2 + \|\bar{x}_s^{\text{in}} - x_r\|_Q^2) \leq -(1-\alpha)^2\frac{1}{4}\|x - x_r\|_Q^2 \ .\end{aligned}$$

In order to show feasibility with respect to the terminal constraint, we prove that $\|\bar{x}_N^{\text{ws}} - \bar{x}_s^{\text{ws}}\|_P \leq \epsilon_s$, which implies that $\bar{x}_N^{\text{ws}} \in \bar{\mathcal{X}}_f^{tr}(\bar{x}_{s,\alpha}^{\text{in}}, \bar{u}_{s,\alpha}^{\text{in}})$ by the definition of ϵ_s:

$$\begin{aligned}\|\bar{x}_N^{\text{ws}} - \bar{x}_s^{\text{ws}}\|_P &= \|\bar{x}_N^{\text{ws}} - \bar{x}_s^{\text{in}} + (1-\alpha)(\bar{x}_s^{\text{in}} - x_r)\|_P \\ &\leq \|(A+BK)^N(\bar{x}_0^{\text{ws}} - \bar{x}_s^{\text{in}})\|_P + (1-\alpha)\|\bar{x}_s^{\text{in}} - x_r\|_P \leq \epsilon_s \ .\end{aligned}$$

which is satisfied for some $\alpha < 1$, since $\|(A+BK)^N(\bar{x}_0^{\text{ws}} - \bar{x}_s^{\text{in}})\|_P < \epsilon_s$.

Hence, there exists an $\alpha < 1$ such that the terminal constraint is feasible and the Lyapunov constraint is satisfied with $\epsilon_k = \frac{1}{2}(1-\alpha)^2$, which will therefore be obtained for $\alpha_{\min}$ in Step 5 of Algorithm 5.

As a result, we have shown that the warm-start solution provided by Algorithm 5 is feasible for $\mathbb{P}_N^{\tau,tr}(x^+, x_r, u_r)$ and satisfies the Lyapunov constraint (8.8f) for some $\epsilon_k > 0 \ \forall k \in \mathbb{N}$. ■

We can now show that the results for regulation presented in Section 8.3 directly extend to the tracking case and feasibility again implies stability.

Lemma 8.21 *Consider Problem $\mathbb{P}_N^{\tau,tr}(x, x_r, u_r)$ fulfilling Assumption 8.15, where $(x_r, u_r) \in \bar{\Theta}$ is a feasible steady-state.*
Let $\tilde{x}_0^\tau(x(k)) \in \mathcal{X}_0^\tau(x(k))$ be a feasible tube center, $(\tilde{x}_s^\tau(x(k)), \tilde{u}_s^\tau(x(k))) \in \bar{\Theta}$ a feasible steady-state and $\tilde{\mathbf{u}}^\tau(x(k)) \in \mathcal{U}_N^{\tau,tr}(\tilde{x}_0^\tau(x(k)), \tilde{x}_s^\tau(x(k)), \tilde{u}_s^\tau(x(k)))$ a feasible control sequence for $\mathbb{P}_N^{\tau,tr}(x(k))$ for all $k \in \mathbb{N}$ and $\tilde{\kappa}^\tau(x(k)) = \tilde{u}_0^\tau(x(k)) + K(x - \tilde{x}_0^\tau(x(k)))$ the resulting control law. Let $\mathcal{S} \triangleq \{x \mid \|x - x_r\|_P \leq \epsilon_f\}$, where $\epsilon_f > 0$ is such that $\mathcal{S} \subseteq x_r \oplus \mathcal{Z}_{\mathcal{W}}$ and assume that $V_N^{tr}(\tilde{\mathbf{x}}^\tau(x(k)), \tilde{\mathbf{u}}^\tau(x(k)), \tilde{x}_s^\tau(x(k)), \tilde{u}_s^\tau(x(k)), x_r, u_r) + V_f(x - \tilde{x}_0^\tau(x(k))) \leq V_f(x(k) - x_r)$ if $x(k) \in \mathcal{S}$.

The closed-loop system $x(k+1) = Ax(k) + B\tilde{\kappa}(x(k)) + w(k)$ converges to an RPI set around $x_r \ \forall \ x \in \bar{\mathcal{X}}_N^{tr}$, i.e. the system $x(k+1) - x_r = A(x(k) - x_r) + B(\tilde{\kappa}(x(k)) - u_r)$ is ISS in $\mathcal{X}_N^{\tau,tr}$ with respect to $w(k) \in \mathcal{W}$.

Proof. We define $V_L^{\tau,tr}(x, x_r, u_r) \triangleq V_N^{tr}(\tilde{\mathbf{x}}^\tau(x), \tilde{\mathbf{u}}^\tau(x), \tilde{x}_s^\tau(x), \tilde{u}_s^\tau(x), x_r, u_r) + V_f(x - \tilde{x}^\tau(x))$. Following similar arguments as in the proof of Lemma 8.6 and using convexity of $\|\bullet\|_Q^2$, we can show that there exists a $\mathcal{K}_\infty$-class function $\underline{\alpha}(\bullet)$ such that

$$\begin{aligned}V_L^{\tau,tr}(x, x_r, u_r) &\geq \|\tilde{x}_0^\tau(x) - \tilde{x}_s^\tau(x)\|_Q^2 + \|x - \tilde{x}_0^\tau(x)\|_Q^2 + \|\tilde{x}_s^\tau(x) - x_r\|_T^2 \\ &\geq \frac{1}{2}\|x - \tilde{x}_s^\tau(x)\|_Q^2 + \frac{1}{2}c_T\|\tilde{x}_s^\tau(x) - x_r\|_Q^2 \\ &\geq \frac{1}{4}c_T\|x - x_r\|_Q^2 \geq \underline{\alpha}(\|x - x_r\|) \ \forall x \in \mathcal{X}_N^{\tau,tr} \ ,\end{aligned} \tag{8.19}$$

where $c_T \leq 1$ is such that $T \succeq c_T Q$. By the assumptions in the lemma, there exists a $\mathcal{K}_\infty$-class function $\overline{\alpha}(\bullet)$ such that

$$V_L^{\tau,tr}(x, x_r, u_r) \leq V_f(x - x_r) \leq \overline{\alpha}(\|x - x_r\|) \ \forall x \in \mathcal{S} \ . \tag{8.20}$$

Furthermore, it follows from the Lyapunov constraint (8.8f) that

$$V_N^{tr}(\tilde{\mathbf{x}}^\tau(x^+), \tilde{\mathbf{u}}^\tau(x^+), \tilde{x}_s^\tau(x^+), \tilde{u}_s^\tau(x^+), x_r, u_r) + V_f(x_{\text{nom}}^{\tau,tr} - \tilde{x}_0^\tau(x^+)) - V_L^{\tau,tr}(x) \le -\frac{1}{2}\epsilon_k l(x - x_r, 0)$$

for all $x \in \mathcal{X}_N^{\tau,tr}$ with $x^+ = Ax + B\tilde{\kappa}(x) + w$ and $\epsilon_k > 0$. Let $\epsilon_{\min} = \min_{k\in\mathbb{N}} \epsilon_k > 0$ be the smallest possible value over all ϵ_k. Following the same argument as in the proof of Lemma 8.6 and replacing ϵ with $\epsilon_{\min}$, there hence exists a $\mathcal{K}_\infty$-class function $\beta(\bullet)$ and a $\mathcal{K}$-class function $\gamma(\bullet)$ such that

$$V_L^{\tau,tr}(x^+, x_r) - V_L^{\tau,tr}(x, x_r) \le -\beta(\|x - x_r\|) + \gamma(\|w\|) \tag{8.21}$$

concluding the proof. ■

Theorem 8.22 (Convergence under $\boldsymbol{\kappa^{\tau,tr}(x,k)}$) *Consider Problem $\mathbb{P}_N^{\tau,tr}(x, x_r, u_r)$ fulfilling Assumption 8.15, where $(x_r, u_r) \in \bar{\Theta}$ is a feasible steady-state. The closed-loop system $x(k+1) = Ax(k) + B\kappa^{\tau,tr}(x(k)) + w(k)$ under the τ-RT control law in (8.3) that is obtained from Algorithm 5 converges to an RPI set around x_r $\forall$ $x \in \bar{\mathcal{X}}_N^{tr}$, i.e. the system $x(k+1) - x_r = A(x(k) - x_r) + B(\tilde{\kappa}(x(k)) - u_r)$ is ISS in $\mathcal{X}_N^{\tau,tr}$ with respect to $w(k) \in \mathcal{W}$.*

Proof. The proof is similar to that of Theorem 8.7.
Let $\tilde{\mathbf{x}}^{\tau,tr}(x), \tilde{\mathbf{u}}^{\tau,tr}(x)$ denote the variables obtained at Step 15 of Algorithm 5. We again define $V_L^{\tau,tr}(x, x_r, u_r) \triangleq V_N^{tr}(\tilde{\mathbf{x}}^{\tau,tr}(x), \tilde{\mathbf{u}}^{\tau,tr}(x), \tilde{x}_s^{\tau,tr}(x), \tilde{u}_s^{\tau,tr}(x), x_r, u_r) + V_f(x - \tilde{x}_s^{\tau,tr}(x))$ and $\mathcal{S} \triangleq \{x \mid \|x - x_r\|_P \le \epsilon_f\} \subseteq x_r \oplus \mathcal{Z}_{\mathcal{W}}$. Feasibility of the warm-start was shown in Lemma 8.20, which is maintained by the use of a primal feasible optimization method. If Steps 12-14 are never applied, ISS follows directly from Lemma 8.21. If Steps 12-14 are applied for some $x(k) \in \mathcal{S}$, it is known that $\tilde{\mathbf{u}} = [u_s, \ldots, u_r], \tilde{x}_0 = x_r, \tilde{x}_s = x_r, \tilde{u}_s = u_r$ is feasible for $\mathbb{P}_N^{\tau,tr}(x(k), x_r, u_r)$ without the Lyapunov constraint and $V_L^{\tau,tr}(x(k), x_r, u_r) = V_N(\tilde{\mathbf{u}}, \tilde{\mathbf{x}}, x_r, u_r, x_r, u_r) + V_f(x(k) - x_r) = V_f(x(k) - x_r)$. This provides the upper bound in (8.20). Let $\tilde{\mathbf{u}}^{\text{opt}}, \tilde{\mathbf{x}}^{\text{opt}}, \tilde{x}_s^{\text{opt}}, \tilde{u}_s^{\text{opt}}$ denote the solution that is returned by the real-time optimization. Since it is feasible for $\mathbb{P}_N^{\tau,tr}(x(k), x_r, u_r)$ including the Lyapunov constraint, it follows from (8.21) in the proof of Lemma 8.21 that

$$\begin{aligned}&V_N(\tilde{\mathbf{x}}^{\text{opt}}, \tilde{\mathbf{u}}^{\text{opt}}, \tilde{x}_s^{\text{opt}}, \tilde{u}_s^{\text{opt}}, x_r, u_r) + V_f(x(k) - \tilde{x}_0^{\text{opt}}) \\ &\le V_L^{\tau,tr}(x(k-1)) - \beta(\|x(k-1) - x_r\|) + \gamma(\|w(k-1)\|)\end{aligned}$$

for some $\mathcal{K}$-class function $\gamma(\bullet)$ and $\mathcal{K}_\infty$-class function $\beta(\bullet)$. Finally, since the conditions in Step 12 are fulfilled, it follows that

$$\begin{aligned}V_L^{\tau,tr}(x(k)) = V_f(x(k) - x_r) &\le V_N(\tilde{\mathbf{x}}^{\text{opt}}, \tilde{\mathbf{u}}^{\text{opt}}, \tilde{x}_s^{\text{opt}}, \tilde{u}_s^{\text{opt}}, x_r, u_r) + V_f(x(k) - \tilde{x}_0^{\text{opt}}) \\ &\le V_L^{\tau,tr}(x(k-1)) - \beta(\|x(k-1) - x_r\|) + \gamma(\|w(k-1)\|) \ .\end{aligned}$$

This shows that $V_L^{\tau,tr}(x)$ is a ISS Lyapunov function, proving the result. ■

Remark 8.23. Note that at the first optimization step after a reference change Problem $\mathbb{P}_N^{\tau,tr}(x, x_r, u_r)$ without the Lyapunov constraint (8.8f) has to be considered in the real-time procedure since the reference change may increase the cost value compared to the previous solution before the reference change and a cost decrease cannot be enforced.

Remark 8.24. The use of an artificial reference and corresponding target set enlarges the domain of attraction compared to a standard MPC approach for reference tracking and $\mathcal{X}_N^{\tau,tr} \supseteq \mathcal{X}_N^{\tau}$ [LAAC08].

Remark 8.25. $V_o(\cdot,\cdot)$ is chosen as a quadratic function and does not represent an exact penalty function (see Section 3.1) since this work focuses on a suboptimal method. Local optimality is hence not guaranteed, i.e. the optimal artificial reference resulting from $\mathbb{P}_N^{\tau,tr}(x, x_r, u_r)$ might differ from the desired reference, although $x_s^* = x_r, u_s^* = u_r$ is a feasible solution and could be enforced. This optimality loss can be reduced by choosing large weight matrices T_x and T_u. All the results on real-time MPC for reference tracking presented in the following, however, directly extend to the use of $1-$ or $\infty-$norms in the offset cost, representing an exact penalty function for sufficiently large weights T_x and T_u [Lue84, FLA$^+$09]. Note that the choice of $V_o(\cdot,\cdot)$ only affects the transient behavior and not the optimal steady-state.

Remark 8.26. The method can be extended to achieve zero-offset for constant disturbances using e.g. the approach described in [MBM09].

Having set the theoretical background, the remaining sections focus on the practical aspects and implementation of the proposed real-time robust MPC approach. First, we provide step-by-step implementation details for the robust MPC problem setup that can be applied to large-scale systems. An optimization method for efficiently solving Problem $\mathbb{P}_N^{\tau,tr}(x, x_r, u_r)$ is then introduced and we prove that all guarantees can be provided at high computational speed.

8.5 Problem Setup

In the robust MPC formulation, the set $\mathcal{Z}_{\mathcal{W}}$ in $\mathbb{P}_N^{\tau}(x)$ or $\mathbb{P}_N^{\tau,tr}(x, x_r, u_r)$ would ideally be taken as the minimal RPI (mRPI) set and $\bar{\mathcal{X}}_f/\bar{\mathcal{X}}_f^{tr}(\bar{x}_s, \bar{u}_s)$ as the maximal PI (MPI) set. An explicit representation of these sets can generally not be computed except in special cases [GT91,KG98,Las93]. It is, however, always possible to compute an invariant outer approximation of the mRPI set and an invariant inner approximation of the MPI set of predefined shape. Note that the crucial property of recursive feasibility is not affected by the use of these approximations. The two commonly used approximation types are polyhedral or ellipsoidal invariant sets. In the case that $\mathcal{Z}_{\mathcal{W}}$ and $\bar{\mathcal{X}}_f/\bar{\mathcal{X}}_f^{tr}(\bar{x}_s, \bar{u}_s)$ are polyhedral sets, the robust MPC problem $\mathbb{P}_N^{\tau}(x)$, or $\mathbb{P}_N^{\tau,tr}(x, x_r, u_r)$ respectively, results in a quadratic program (QP). If $\mathcal{Z}_{\mathcal{W}}$ and $\bar{\mathcal{X}}_f/\bar{\mathcal{X}}_f^{tr}(\bar{x}_s, \bar{u}_s)$ are represented by ellipsoidal sets, it can be transformed into a quadratically constrained QP (QCQP), that is a QP with two extra quadratic constraints on the initial and the terminal state.

A key question is then, which type of approximation to use for the invariant set computations. Since the set computations are performed offline the computation times are not crucial. In general, polyhedral approximations can only be computed for smaller systems of approximately $6 - 7$ dimensions. Whereas ellipsoidal approximations might be more conservative in this range, they represent the better, if not the only choice for higher dimensions. If the considered system is in the range where a polyhedral approximation can be computed, an explicit solution of the MPC problem should be considered as well, since it allows for extremely fast computation times in lower dimensions (see Section 6.1). Another advantage of ellipsoidal invariant sets is the fact that the number of constraints that are introduced is fixed, whereas in the polytopic case, the calculated polytopes may add a large number of constraints, leading to slower computation times and excessive memory requirements.

Remark 8.27. Whereas the use of potentially more conservative ellipsoidal approximations may reduce the region of attraction, the tracking formulation $\mathbb{P}_N^{\tau,tr}(x, x_r, u_r)$ has a larger region of attraction than standard MPC approaches (see Remark 8.24), thereby reducing the conservatism introduced by the robust MPC formulation.

Remark 8.28. Computation of the invariant sets and tightened constraints using ellipsoidal approximations is computationally tractable and can be efficiently solved even for large-scale systems because it involves the solution of convex Linear Matrix Inequalities (LMIs). We therefore restrict the description of the implementation details to the use of ellipsoidal approximations. For methods on polytopic invariant set computations see e.g. [GT91, ALBH07, RKKM05, RMKK05].

The details for computing ellipsoidal invariant sets for $\mathcal{Z}_{\mathcal{W}}$ and $\bar{\mathcal{X}}_f/\bar{\mathcal{X}}_f^{tr}(\bar{x}_s, \bar{u}_s)$ as well as the tightened constraints $\bar{\mathbb{X}}$ and $\bar{\mathbb{U}}$ are outlined in the following. For simplicity, the terminal cost is taken as the unconstrained infinite horizon optimal cost $V_f(x) = x^T P x$ and the corresponding optimal infinite horizon linear control law is used for K in (8.3), although there are different ways of choosing a stabilizing affine controller [KBM96, RMKK05]. We denote $A_K \triangleq A + BK$. The polyhedral state and control constraints are defined as $\mathbb{X} = \{x \,|\, G_x x \leq f_x\}$ and $\mathbb{U} = \{u \,|\, G_u u \leq f_u\}$, the tightened constraints as $\bar{\mathbb{X}} = \{x \,|\, G_x x \leq \bar{f}_x\}$ and $\bar{\mathbb{U}} = \{u \,|\, G_u u \leq \bar{f}_u\}$, with $G_x \in \mathbb{R}^{m_x \times n_x}$, $f_x, \bar{f}_x \in \mathbb{R}^{m_x}$, $G_u \in \mathbb{R}^{m_u \times n_u}$ and $f_u, \bar{f}_u \in \mathbb{R}^{m_u}$. For simplicity we assume that $\mathcal{W} =$

$\{w \mid \|w\|_2^2 \leq \delta\}$. The results can, however, be extended to the case where $\mathcal{W}$ is a general ellipse or the intersection of an ellipse and a subspace.

The computations necessary for the problem setup $\mathbb{P}_N^{\tau}(x)$ in the regulation case or $\mathbb{P}_N^{tr}(x, x_r, u_r)$ in the tracking case, respectively, are summarized in Algorithm 6. The remainder of this section explains steps 1-3 of the preprocessing algorithm in more detail.

1) *Ellipsoidal approximation of $\mathcal{Z}_{\mathcal{W}}$*: An RPI ellipsoidal outer approximation $\mathcal{E}_{\mathcal{Z}}$ of the mRPI set can be determined using a level set of $V_f(x) = x^T P x$. Extra constraints are added enforcing that the ellipsoid is in fact an RPI set and all state and input constraints are satisfied. This problem can be transformed into an LMI using the S-procedure and choosing the parameter $\tau_2 \in [0, 1]$ [BEFB94,LÖ1].

2) *Constraint tightening*: The Minkowski differences $\bar{\mathbb{X}} = \mathbb{X} \ominus \mathcal{Z}_{\mathcal{W}}$, $\bar{\mathbb{U}} = \mathbb{U} \ominus K\mathcal{Z}_{\mathcal{W}}$ can be computed by determining the support function of the set $\mathcal{E}_{\mathcal{Z}}$ evaluated at the constraints using the closed-form solution or by solving a series of LPs. The tightened constraints $\bar{\mathbb{X}}$ and $\bar{\mathbb{U}}$ are polytopic and of the same complexity as the original constraints $\mathbb{X}$ and $\mathbb{U}$.

3a) *Ellipsoidal approximation of $\bar{\mathcal{X}}_f$ satisfying Assumption 8.1*: Most approaches use the level set of a quadratic Lyapunov function to derive an invariant ellipsoidal inner approximation of the MPI set [GSdD06, BEFB94, KBM96]. 1.: In the considered case a Lyapunov function is readily available with $V_f(x) = x^T P x$ and an ellipsoidal PI set $\mathcal{E}_{\bar{\mathcal{X}}_f}$ can be computed as the biggest level set fulfilling the state and control constraints, resulting in a simple 1-dimensional LP. 2.: Alternatively, a maximal volume ellipsoidal PI set $\mathcal{E}_{\bar{\mathcal{X}}_f}$ can be employed by computing a quadratic Lyapunov function with the largest level set inside the constraints, which can be formulated as an LMI [BEFB94].

3b) *Ellipsoidal approximation of $\bar{\mathcal{X}}_f^{tr}(x_s, u_s)$ satisfying Assumption 8.15*: The terminal set for tracking introduced in [LAAC08] is the maximal PI set given by the set of feasible states, feasible steady-states and inputs, such that the control law $u = K(x - x_s) + u_s$ is feasible and stabilizes the nominal system in (4.4). Considering the parametrization $x_s = M_x\theta, u_s = M_u\theta$ with $\theta \in \mathbb{R}^{n_\theta}$ described in Section 4.1 and the augmented system $v = \begin{bmatrix} x^T - x_s^T & \theta^T \end{bmatrix}^T$ in (5.16), a quadratic Lyapunov function $V(x) = v^T \begin{bmatrix} P_1 & \\ & P_2 \end{bmatrix} v$ can be computed, such that a level set provides the largest invariant ellipsoid fulfilling the state and control constraints. This can again be formulated as an LMI [BEFB94]. $\mathcal{E}_{\bar{\mathcal{X}}_f^{tr}}(x_s, u_s)$ is then obtained using the relationship $\theta = M^T[x_s^T \; u_s^T]^T$.

The following section is devoted to the main challenge of developing a fast optimization procedure for the presented real-time robust MPC method guaranteeing feasibility

Algorithm 6 Preprocessing Algorithm

```
Input: $G_x, f_x$, $G_u, f_u$, $K$ and $\delta$.
Output: $\mathcal{E}_{\mathcal{Z}}$, $\bar{f}_x$, $\bar{f}_u$ and $\mathcal{E}_{\bar{\mathcal{X}}_f}$ or $\mathcal{E}_{\bar{\mathcal{X}}_f^{tr}}(x_s, u_s)$.
  1: Compute ellipsoidal RPI set $\mathcal{E}_{\mathcal{Z}}$: $\mathcal{E}_{\mathcal{Z}} = \{x \in \mathbb{R}^{n_x} \,|\, x^T P x \leq \gamma_{\min}\}$, where
       $\gamma_{\min} = \underset{\gamma, \tau_1}{\operatorname{argmin}} \Big\{ \gamma \,\Big|\, \Big[\begin{smallmatrix} \tau_2 P - (A+BK)^T P (A+BK) & -(A+BK)^T P \\ -P(A+BK) & \tau_1 I - P \end{smallmatrix}\Big] \succeq 0, \tau_1 \delta + \tau_2 \gamma \leq \gamma, \tau_1 \geq 0,$
           $\|P^{-\frac{1}{2}} G_{x,i}^T\|_2^2 \gamma \leq f_{x,i}^2 \;\forall\; i = 1, \ldots, m_x, \|P^{-\frac{1}{2}} K^T G_{u,j}^T\|_2^2 \gamma \leq f_{u,j}^2 \;\forall\; j = 1, \ldots, m_u\} \;.$
  2: Compute tightened constraints $\bar{\mathbb{X}}$, $\bar{\mathbb{U}}$:
       $\bar{f}_{x,i} = f_{x,i} - h_{\mathcal{E}_{\mathcal{Z}}}(G_{x,i}^T), \forall\; i = 1, \ldots, m_x, \;\; \bar{f}_{u,i} = f_{u,i} - h_{\mathcal{E}_{\mathcal{Z}}}(G_{u,i}^T), \forall\; i = 1, \ldots, m_u \;,$
     where $h_{\mathcal{E}_{\mathcal{Z}}}(a) \triangleq \sup_{x \in \mathcal{E}_{\mathcal{Z}}} a^T x = \sqrt{\gamma_{\min}} \frac{a^T P^{-1} a}{\|P^{-\frac{1}{2}} a\|_2}$ is the support function of $\mathcal{E}_{\mathcal{Z}}$ evaluated at $a$ .
  3a: *Regulation:* Compute ellipsoidal PI set $\mathcal{E}_{\bar{\mathcal{X}}_f}$:
     1: $\mathcal{E}_{\bar{\mathcal{X}}_f} = \{x \in \mathbb{R}^{n_x} \,|\, x^T P x \leq \gamma_{\max}\}$, where
       $\gamma_{\max} = \underset{\gamma}{\operatorname{argmin}} \{-\gamma \,|\, \|P^{-\frac{1}{2}} G_{x,i}^T\|_2^2 \gamma \leq f_{x,i}^2 \;\forall\; i = 1, \ldots, m_x, \|P^{-\frac{1}{2}} K^T G_{u,j}^T\|_2^2 \gamma \leq f_{u,j}^2$
           $\forall\; j = 1, \ldots, m_u\} \;.$
     2: $\mathcal{E}_{\bar{\mathcal{X}}_f} = \{x \in \mathbb{R}^{n_x} \,|\, x^T Q^{-1} x \leq 1\}$, where
       $Q = \underset{Q}{\operatorname{argmin}} \Big\{ -\log\det(Q) \,\Big|\, \Big[\begin{smallmatrix} Q & Q A_K^T \\ A_K Q & Q \end{smallmatrix}\Big] \succeq 0, \|Q^{\frac{1}{2}} G_{x,j}^T\|_2^2 \leq f_{x,j}^2 \forall\; j = 1, \ldots, m_x,$
           $\|Q^{\frac{1}{2}} K^T G_{u,j}^T\|_2^2 \leq f_{u,j}^2 \forall\; j = 1, \ldots, m_u \;\} \;.$
  3b: *Tracking:* Compute ellipsoidal PI set $\mathcal{E}_{\bar{\mathcal{X}}_f^{tr}}(x_s, u_s)$:
     $\mathcal{E}_{\Omega_f} = \{v \in \mathbb{R}^{n_x + n_\theta} \,|\, v^T \Big[\begin{smallmatrix} Q_1^{-1} & \\ & Q_2^{-1} \end{smallmatrix}\Big] v \leq 1\}$, where
       $\Big[\begin{smallmatrix} Q_1 & \\ & Q_2 \end{smallmatrix}\Big] = \underset{Q_1, Q_2}{\operatorname{argmin}} \Big\{ -\log\det\Big(\Big[\begin{smallmatrix} Q_1 & \\ & Q_2 \end{smallmatrix}\Big]\Big) \,\Big|\, \Big[\begin{smallmatrix} Q_1 & Q_1 A_K^T \\ A_K Q_1 & Q_1 \end{smallmatrix}\Big] \succeq 0,$
           $\|Q_1^{\frac{1}{2}} G_{x,i}^T\|_2^2 + \|Q_2^{\frac{1}{2}} M_x^T G_{x,i}^T\|_2^2 \leq f_{x,i}^2 \forall\; i = 1, \ldots, m_x,$
           $\|Q_1^{\frac{1}{2}} K^T G_{u,j}^T\|_2^2 + \|Q_2^{\frac{1}{2}} M_u^T G_{u,j}^T\|_2^2 \leq f_{u,j}^2 \forall\; j = 1, \ldots, m_u \;\} \;.$
     Then $\mathcal{E}_{\bar{\mathcal{X}}_f^{tr}}(x_s, u_s) = \{x \in \mathbb{R}^{n_x} \;|\; (x - x_s)^T Q_1^{-1} (x - x_s) \leq 1 - [x_s^T \;\, u_s^T] M Q_2^{-1} M^T [x_s^T \;\, u_s^T]^T\}$.
```

and stability at high speeds.

8.6 Real-time Optimization Procedure

For the description of the optimization procedure we focus on the real-time robust MPC problem for tracking $\mathbb{P}_N^{\tau,tr}(x, x_r, u_r)$, the results can, however, be equivalently applied to solve the real-time robust problem for regulation $\mathbb{P}_N^{\tau}(x)$. For a given value $x \in \mathcal{X}_N^{\tau,tr}$, problem $\mathbb{P}_N^{\tau,tr}(x, x_r, u_r)$ can be written as a QCQP of the following form:

$$\min_z \; z^T Hz + g^T z \tag{8.22a}$$

$$\text{s.t. } Cz = b \;, \tag{8.22b}$$

$$G(z) \leq 0, \tag{8.22c}$$

where the vector of decision variables $z^T = \begin{bmatrix} \bar{x}_0^T, & \bar{u}_0^T, & \bar{u}_1^T, & \ldots & \bar{x}_N^T, & \bar{x}_s^T, & \bar{u}_s^T \end{bmatrix}$, $C \in \mathbb{R}^{(N+1)n_x \times (N+1)(n_x+n_u)}$ and $b \in \mathbb{R}^{(N+1)n_x}$ contain the equality constraints in (8.2b) and $G(z)$ is formed from the scalar functions $g_i(z)$, $i = 1, \ldots, m$ that contain the affine and quadratic inequality constraints in (8.2c),(8.2f),(8.8d)-(8.8f).

There are several optimization methods that can be used to solve (8.22). Recent work on fast MPC has shown how e.g. active set methods [FBD08], interior-point methods [WB10,Wri97a] or, for a special subclass of problems, gradient methods [RJM09] can be applied to control high speed systems, see also the survey in Section 6.2. An active set approach could, however, only be applied within a sequential quadratic programming framework to solve the QCQP in (8.22), which does not provide feasibility (and thereby stability) at all times due to the linearization of the constraints. In order to solve the QCQP in (8.22), a feasible start primal barrier interior-point method (IPM) is therefore applied in this work, which provides feasibility at all times and can efficiently solve QCQPs [NW06]. The results on structure exploitation in MPC methods presented in the literature, e.g. in [WB10, Wri97a], can, however, not be directly applied in the considered case due to the fact that the Lyapunov constraint and the modified cost of the tracking formulation introduce coupling across the horizon and thereby significantly modify the structure of the resulting optimization problem. A modified method for fast computation of the real-time robust MPC control law for tracking is developed in the following, based on an approximate primal barrier interior-point method.

8.6.1 Approximate Primal Barrier Interior-Point Method

In a barrier method, the inequality constraints of the QCQP in (8.22c) are replaced by a barrier penalty in the cost function resulting in the approximate problem formulation:

$$\min_{z} \; z^T H z + g^T z + \kappa\psi(z) \tag{8.23a}$$

$$\text{s.t. } Cz = b \;\; , \tag{8.23b}$$

where κ is the barrier parameter and ψ is the log barrier function given by

$$\psi(z) = \sum_{i=1}^{m} -\log(-g_i(z)) \;\; . \tag{8.24}$$

The method starts from a strictly primal feasible (interior) point and then solves a sequence of linearly constrained minimization problems (8.23) for decreasing values of the barrier parameter κ starting from the previous iterate (see Algorithm 7) using Newton's method (e.g. [BV04,NW06]). As $\kappa \to 0$ the solution converges to the optimal solution of (8.22). See Section 3.1 and [BV04, NW06] for a detailed description of the method and the choice of parameters involved in the procedure.

The warm-start approximate interior-point method is outlined in Algorithm 7. The procedure is initialized with the warm-start solution defined in Algorithm 5 (Step 1). We can start with any ν, e.g. $\nu = 0$, where ν are the Lagrange multipliers associated with the equality constraints $Cz = d$. This initial solution is then improved by taking feasible Newton iterations (Steps $2-8$). At each iteration, the primal and dual search directions Δz and $\Delta\nu$ from the current iterate z and ν are obtained by solving the following linear system (Step 3):

$$\begin{bmatrix} \nabla^2_{zz}\mathcal{L}(z,\nu) + \kappa\Delta G(z)^T S^{-2}\Delta G(z) & C^T \\ C & 0 \end{bmatrix} \begin{bmatrix} \Delta z \\ \Delta\nu \end{bmatrix} = -\begin{bmatrix} r_d \\ 0 \end{bmatrix} \tag{8.25}$$

where $\mathcal{L}(z,\nu) = z^T H z + g^T z + (Cz-b)^T\nu + (G(z)+s)^T\lambda$ with $G(z)+s = 0$ and $\Delta G(z)$ is the Jacobian matrix of $G(z)$. The dual residual is $r_d = 2Hz + g + \kappa\Delta G(z)^T S^{-1} + C^T\nu$, where $S = \text{diag}(s)$. See e.g. [BV04] for details on Steps 3-5. The τ-RT control sequence and tube center are then obtained from the solution z after time τ (Step 9).

Remark 8.29. Note that the warm-start defined in Algorithm 5 provides a strictly interior point for the Lyapunov constraint (8.8f) to warm-start the barrier interior-point method.

Remark 8.30. One can employ a variant of the barrier interior-point method described in [WB10], where the barrier parameter is fixed to a predefined value instead of gradually decreasing it to zero. This reduces the procedure to a Newton method, which solves the approximate problem (8.23) for one particular value of κ and therefore requires lower computation times. While a fixed barrier parameter introduces a steady-state offset using the method in [WB10], the proposed approach provides convergence to the desired steady-state by means of the Lyapunov decrease constraint.

Algorithm 7 Approximate primal barrier method, [BV04, NW06]

```
Input: feasible control sequence $\bar{\mathbf{u}}^{\text{ws}}(x)$, initial tube center $\bar{x}_0^{\text{ws}}(x)$ and steady-state $\bar{x}_s^{\text{ws}}(x), \bar{u}_s^{\text{ws}}(x)$ for $\mathbb{P}_N^{\tau,tr}(x, x_r, u_r)$, initial barrier parameter $\kappa$, $\beta \in (0,1)$, tolerance $\epsilon_{\text{tol}}$
Output: $\tau$-RT optimizer $\bar{\mathbf{u}}^{\tau,tr}(x)$, $\bar{x}_0^{\tau,tr}(x)$, $\bar{x}_s^{\tau,tr}(x)$, $\bar{u}_s^{\tau,tr}(x)$
Build $z$ from $\bar{\mathbf{u}}^{\text{ws}}(x)$, $\bar{x}_0^{\text{ws}}(x)$ $\bar{x}_s^{\text{ws}}(x)$ and $\bar{u}_s^{\text{ws}}(x)$, choose $\nu$
while clock $< \tau$ do
    Newton step direction: Compute $\Delta z$, $\Delta\nu$ in (8.25)
    Line search: Choose step size $\alpha$
    Update: $z = z + \alpha\Delta z$, $\nu = \nu + \alpha\Delta\nu$, $r_d$ in (8.25)
    if $r_d < \epsilon_{\text{tol}}$ then $z$ is optimal solution to (8.23) for $\kappa$, $\kappa = \beta\kappa$
    end if
end while
Extract $\bar{\mathbf{u}}^{\tau,tr}(x)$, $\bar{x}_0^{\tau,tr}(x)$, $\bar{x}_s^{\tau,tr}(x)$, $\bar{u}_s^{\tau,tr}(x)$ from $z$
```

8.6.2 Fast Newton Step Computation

The main computational effort when solving an optimization problem using the approximate primal barrier interior-point method in Algorithm 7 is the Newton step computation in (8.25). While it has been shown in the literature how this linear system can be solved efficiently for a standard MPC formulation using factorizations or Riccati recursions (e.g. [Wri97a, Han00, WB10]), these results can not directly be applied to the considered case due to the fact that the Lyapunov constraint as well as the tracking formulation significantly modify the structure of the Newton step computation in (8.25). $\nabla^2_{zz}\mathcal{L}(z,\nu)$ has an arrow shape, $\Delta G(z)$ is block-diagonal with a dense row at the bottom and C is banded, which causes the term $\nabla^2_{zz}\mathcal{L}(z,\nu) + \kappa\Delta G^T(z)^T S^{-2}\Delta G(z)$ in (8.25) to be dense. We will show in the following how the particular structure can be exploited and fast computation of the Newton step direction can still be achieved with a complexity that is of the same order as that of standard MPC approaches.

During each iteration, the primal and dual search directions Δz and $\Delta\nu$ from the current iterate z and ν are obtained efficiently by solving a modified Newton step computation instead of (8.25) that distinguishes between parts involving the sequence of states and inputs $\bar{\mathbf{x}}, \bar{\mathbf{u}}$ and the ones involving the artificial steady state and input $\bar{x}_s, \bar{u}_s$. Thereby, the dense and the sparse parts are separated using the following parametrization: $\Delta z^T = [\Delta\bar{z}^T \;\Delta z_s^T]$, $\Delta\nu^T = [\Delta\bar{\nu}^T \;\Delta\nu_s^T]$,

$$\nabla^2_{zz}\mathcal{L}(z,\nu) = \begin{bmatrix} \bar{L} & l_1 \\ l_1^T & l_2 \end{bmatrix}, \Delta G(z) = \begin{bmatrix} \bar{G} & f_1 \\ f_2 & f_3 \end{bmatrix}, C = \begin{bmatrix} \bar{C} & 0 \\ 0 & a \end{bmatrix}, S = \begin{bmatrix} \bar{S} & 0 \\ 0 & s_J \end{bmatrix}, \tag{8.26a}$$

$$H = \begin{bmatrix} \bar{H} & h_1 \\ h_1^T & h_2 \end{bmatrix}, g = \begin{bmatrix} \bar{g} \\ g_1 \end{bmatrix}. \tag{8.26b}$$

Note that the dependence of $\bar{L}, l_1, l_2, \bar{G}, f_1, f_1, f_2, f_3$ on z is omitted for ease of notation.

Structure of the Matrices in the Modified Newton Step Computation

The matrices and vectors $\bar{H}, h_1, h_2, \bar{g}, g_1, \bar{L}, l_1, l_2, \bar{G}, f_1, f_2, f_3, \bar{C}, a, \bar{S}$ and s_J are described in the following. It is assumed that the quadratic initial state constraint is given by $G_0(\bar{x}_0) \leq 0$ and the quadratic terminal state constraint is $G_N(\bar{x}_N, \bar{x}_s, \bar{u}_s) \leq 0$. The tightened polyhedral state and input constraints are again $\bar{\mathbb{X}} = \{x \,|\, G_x x \leq \bar{f}_x\}$ and $\bar{\mathbb{U}} = \{u \,|\, G_u u \leq \bar{f}_u\}$.

The matrices $\bar{L}, l_1, l_2, \bar{G}, f_1, f_2, f_3, \bar{C}$ and a in (8.26) are then given by:

$$\bar{L} = \begin{bmatrix} (2Q{+}2P)(1{+}\lambda_J){+}\frac{\partial^2}{\partial \bar{x}_0^2} G_0(\bar{x}_0)\lambda_0 & 0 & & & & \\ 0 & 2R(1{+}\lambda_J) & & & & \\ & & \ddots & & & \\ & & & 2Q(1{+}\lambda_J) & 0 & \\ & & & 0 & 2R(1{+}\lambda_J) & \\ & & & & & \begin{array}{c} 2P(1{+}\lambda_J) \\ +\frac{\partial^2}{\partial \bar{x}_N^2} G_N(\bar{x}_N,\bar{x}_s,\bar{u}_s)\lambda_N \end{array} \end{bmatrix}, \tag{8.27}$$

$$l_1 = \begin{bmatrix} -2Q(1{+}\lambda_J) & 0 \\ 0 & -2R(1{+}\lambda_J) \\ \vdots & \vdots \\ -2Q(1{+}\lambda_J) & 0 \\ 0 & -2R(1{+}\lambda_J) \\ -2P(1{+}\lambda_J){+}\frac{\partial^2}{\partial \bar{x}_N \partial \bar{x}_s} G_N(\bar{x}_N,\bar{x}_s,\bar{u}_s)\lambda_N & \frac{\partial^2}{\partial \bar{x}_N \partial \bar{u}_s} G_N(\bar{x}_N,\bar{x}_s,\bar{u}_s)\lambda_N \end{bmatrix}, \tag{8.28}$$

$$l_2 = \begin{bmatrix} (N2Q{+}2P{+}2T_x)(1{+}\lambda_J){+}\frac{\partial^2}{\partial \bar{x}_s^2} G_N(\bar{x}_N,\bar{x}_s,\bar{u}_s)\lambda_N & \frac{\partial^2}{\partial \bar{x}_s \partial \bar{u}_s} G_N(\bar{x}_N,\bar{x}_s,\bar{u}_s)\lambda_N \\ \frac{\partial^2}{\partial \bar{u}_s \partial \bar{x}_s} G_N(\bar{x}_N,\bar{x}_s,\bar{u}_s)\lambda_N & (N2R{+}2T_u)(1{+}\lambda_J){+}\frac{\partial^2}{\partial \bar{u}_s^2} G_N(\bar{x}_N,\bar{x}_s,\bar{u}_s)\lambda_N \end{bmatrix}, \tag{8.29}$$

$$\bar{G} = \begin{bmatrix} \frac{\partial}{\partial \bar{x}_0} G_0(\bar{x}_0) & 0 & & & & \\ 0 & \bar{G}_u & & & & \\ & & \ddots & & & \\ & & & \bar{G}_x & 0 & \\ & & & 0 & \bar{G}_u & \\ & & & & & \frac{\partial}{\partial \bar{x}_N} G_N(\bar{x}_N,\bar{x}_s,\bar{u}_s) \end{bmatrix}, f_1 = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ \vdots & \vdots \\ 0 & 0 \\ 0 & 0 \\ \frac{\partial}{\partial \bar{x}_s} G_N(\bar{x}_N,\bar{x}_s,\bar{u}_s) & \frac{\partial}{\partial \bar{u}_s} G_N(\bar{x}_N,\bar{x}_s,\bar{u}_s) \end{bmatrix}, \tag{8.30}$$

$$f_2 = \left[\, 2(\bar{x}_0{-}\bar{x}_s)^T Q{-}2(x_{\mathrm{nom}}^{\tau,tr}{-}\bar{x}_0)^T Q \quad 2(\bar{u}_0{-}\bar{u}_s)^T R \quad \cdots \quad 2(\bar{x}_{N-1}{-}\bar{x}_s)^T Q \quad 2(\bar{u}_{N-1}{-}\bar{u}_s)^T R \quad 2(\bar{x}_N{-}\bar{x}_s)^T P \,\right], \tag{8.31}$$

$$f_3 = \left[\, -\textstyle\sum_{k=0}^{N-1} 2(\bar{x}_k{-}\bar{x}_s)^T Q{-}2(\bar{x}_N{-}\bar{x}_s)^T P{+}2(\bar{x}_s{-}x_r)^T T_x \quad \sum_{k=0}^{N-1} 2(\bar{u}_k{-}\bar{u}_r)^T R{+}2(\bar{u}_s{-}u_s)^T T_u \,\right], \tag{8.32}$$

$$\bar{C} = \begin{bmatrix} A & B & -I & & & \\ & & A & & & \\ & & & \ddots & & \\ & & & A & B & -I \end{bmatrix}, a = \left[\, A{-}I \;\; B \,\right], \tag{8.33}$$

where s_J is the slack variable, λ_J the Lagrange multiplier associated with the Lyapunov constraint in (8.8f) and $\bar{S}$ is a diagonal matrix whose entries are given by the remaining slack variables. $\bar{H}, h_1, h_2$ is the partition of H and similarly $\bar{g}, g_1$ is the partition of g corresponding to the parametrization of z.

Modified Newton Step Computation

The modified Newton step is obtained from the following linear system:

$$\left[\begin{array}{cc|ccc} \bar{L}+\kappa\bar{G}^T\bar{S}^{-2}\bar{G} & \bar{C}^T & 0 & f_2^T & l_1+\kappa\bar{G}^T\bar{S}^{-2}f_1 \\ \bar{C} & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & a \\ f_2 & 0 & 0 & -\frac{1}{\kappa}s_J^2 & f_3 \\ l_1^T+\kappa f_1^T\bar{S}^{-2}\bar{G} & 0 & a^T & f_3^T & l_2+\kappa f_1^T\bar{S}^{-2}f_1 \end{array}\right] \left[\begin{array}{c} \Delta\bar{z} \\ \Delta\bar{\nu} \\ \hline \Delta\nu_s \\ \Delta\lambda_J \\ \Delta z_s \end{array}\right] = - \left[\begin{array}{c} r_d \\ 0 \\ \hline 0 \\ 0 \\ r_{d,s} \end{array}\right] \tag{8.34}$$

where

$$r_d = 2\bar{H}\bar{z} + 2h_1 z_s + \bar{C}^T\bar{\nu} + \kappa\bar{G}^T\bar{S}^{-1}\mathbf{1} + \kappa f_2^T s_J^{-1}, \tag{8.35}$$

$$r_{d,s} = 2h_1^T\bar{z} + 2h_2 z_s + a^T\nu_s + \kappa f_1^T\bar{S}^{-1}\mathbf{1} + \kappa f_3^T s_J^{-1} \tag{8.36}$$

and $\mathbf{1} = [1, \dots, 1]^T$ denotes a vector of ones of appropriate dimension.

We show in the following that the modified Newton step computation in (8.34) is equivalent to the standard Newton step in (8.25). Equation (8.34) can be directly derived from the full primal-dual Newton step [NW06]:

$$\begin{bmatrix} \nabla_{zz}^2\mathcal{L}(z,\nu) & 0 & C^T & \Delta G^T(z) \\ 0 & \Sigma & 0 & I \\ C & 0 & 0 & 0 \\ \Delta G(z) & I & 0 & 0 \end{bmatrix} \begin{bmatrix} \Delta z \\ \Delta s \\ \Delta\nu \\ \Delta\lambda \end{bmatrix} = - \begin{bmatrix} r_d \\ \lambda - \kappa S^{-1}\mathbf{1} \\ r_p \\ 0 \end{bmatrix}, \tag{8.37}$$

with $\Sigma = S^{-1}\Lambda$, where λ are the Lagrange multipliers associated with the inequality constraints in (8.22c) and $\Lambda = \text{diag}(\lambda)$. Note that the primal residual is zero, i.e. $r_p = 0$, since a primal feasible IPM is considered. Using the parametrization in (8.26) yields

$$\begin{bmatrix} \bar{L} & l_1 & & & \bar{C}^T & & \bar{G}^T & f_2^T \\ l_1^T & l_2 & & & & a^T & f_1^T & f_3^T \\ & & \bar{\Sigma} & & & & I & \\ & & & \sigma_J & & & & I \\ \bar{C} & & & & & & & \\ & a & & & & & & \\ \bar{G} & f_1 & I & & & & & \\ f_2 & f_3 & & I & & & & \end{bmatrix} \begin{bmatrix} \Delta\bar{z} \\ \Delta z_s \\ \Delta\bar{s} \\ \Delta s_J \\ \Delta\bar{\nu} \\ \Delta\nu_s \\ \Delta\bar{\lambda} \\ \Delta\lambda_J \end{bmatrix} = - \begin{bmatrix} r_d \\ r_{d,s} \\ \bar{\lambda} - \kappa\bar{S}^{-1}\mathbf{1} \\ \lambda_J - \kappa s_J^{-1}\mathbf{1} \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{8.38}$$

with $\bar{\Sigma} = \bar{S}^{-1}\bar{\Lambda}$, $\sigma_J = s_J^{-1}\lambda_J$. In the barrier method we use $\Lambda = \kappa S^{-1}$ [NW06], hence $\bar{\Sigma} = \kappa \bar{S}^{-2}$ and $\sigma_J = \kappa s_J^{-2}$. Eliminating $\Delta\bar{s} = -\bar{\Sigma}^{-1}\Delta\bar{\lambda}$, $\Delta s_J = -\sigma_J^{-1}\Delta\lambda_J$, $\Delta\bar{\lambda} = \kappa\bar{S}^{-2}(\bar{G}\Delta\bar{z} + f_1\Delta z_s)$ and reordering yields (8.34).

The complexity for solving the modified Newton step equation in (8.34) is analyzed in the following. For simplicity we rewrite (8.34) according to the separating lines into

$$\begin{bmatrix} \Phi & \vartheta \\ \vartheta^T & \varphi \end{bmatrix} \begin{bmatrix} y \\ y_s \end{bmatrix} = \begin{bmatrix} r \\ r_s \end{bmatrix} \; . \tag{8.39}$$

After reordering, Φ is a block diagonal or banded matrix [Wri97a] and the matrix in (8.39) results in an arrow structure with a dense band ϑ that is only of size $2n_x + n_u + 1$, where n_x is the state and n_u is the input dimension. Equation (8.39) can then be solved for y and y_s using the following three steps:

1. Solve $\Phi\, y_A = r$ for y_A, $\Phi\, Y_B = \vartheta$ for Y_B.
2. Solve $(\varphi - \vartheta^T Y_B) y_s = r_s - \vartheta^T y_A$ for y_s.
3. Solve $\Phi\, y = r - \vartheta y_s$ for y.

Remark 8.31. Note that Φ corresponds to the left hand side of (8.25) for a standard MPC setup for regulation without the Lyapunov decrease constraint and Step 1 and 3 can therefore be solved with complexity $\mathcal{O}(N(n_x + n_u)^3)$ using existing results in the literature [Wri97a, Han00, WB10].

Solution of Steps 1 and 3

We outline the procedure described in [WB10] that is based on block elimination [BV04]. A system of the form $\Phi \left[\begin{smallmatrix} y_1 \\ y_2 \end{smallmatrix}\right] = \left[\begin{smallmatrix} \Psi & \bar{C}^T \\ \bar{C} & 0 \end{smallmatrix}\right] \left[\begin{smallmatrix} y_1 \\ y_2 \end{smallmatrix}\right] = \left[\begin{smallmatrix} r_1 \\ r_2 \end{smallmatrix}\right]$ with the block diagonal matrix $\Psi = \bar{L} + \kappa \bar{G}^T \bar{S}^{-1} \bar{G}$ is solved by means of the following operations:

a) Build $Y = \bar{C}^T \Psi^{-1} \bar{C}$ and $\beta = -r_2 + \bar{C}^T \Psi^{-1} r_1$ using a Cholesky factorization of the block diagonal matrix Ψ.

b) Solve $Y y_2 = \beta$ for y_2 using a Cholesky factorization of the block tridiagonal matrix Y.

c) Solve $\Psi y_1 = r_1 - \bar{C}^T y_2$ for y_1 using the Cholesky factorization of Ψ computed in a).

As was shown in [WB10] Steps a)-c) require a complexity in the order of $\mathcal{O}(N(n_x+n_u)^3)$.

Solution of Step 2

Matrix $\varphi - \vartheta^T Y_B$ is dense but only of the order $\mathbb{R}^{2n_x+n_u+1 \times 2n_x+n_u+1}$. Step 2 can therefore be computed with the lower complexity $\mathcal{O}((n_x+n_u)^3)$ using a factorization of $\varphi - \vartheta^T Y_B$.

The overall effort for Steps 1-3 and the entire Newton step computation in (8.39) is hence of the order $\mathcal{O}(N(n_x+n_u)^3)$ and the real-time robust MPC problem with stability guarantees can be solved with the same complexity that was shown for standard MPC approaches.

8.6.3 Implementation

A custom solver written in C++ was developed, extending the results given in [WB10], for the real-time method proposed in this chapter that yields computation times in the range of milliseconds (see results in Section 8.7). This offers the possibility to apply real-time robust MPC to high-speed systems with the significant advantage that stability and constraint satisfaction are always guaranteed and the available computation time is used to improve the solution and increase the performance.

8.7 Results & Examples

The presented results are demonstrated in the following sections using three numerical examples. The offline set computations were carried out using the YALMIP toolbox [Löf04] and the solver 'SeDuMi' [SED]. The simulations were executed on a 2.8GHz AMD Opteron running Linux using a single core.

8.7.1 Illustrative Example

We first illustrate the method and its components using the following 2D system:

$$x(k+1) = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} x(k) + \begin{bmatrix} 1 \\ 0.5 \end{bmatrix} u(k) + w(k) \ , \tag{8.40}$$

with a prediction horizon $N = 5$ and the constraints $\|x\|_\infty \leq 10$ and $\|u\|_\infty \leq 1$ on the states and control inputs, $Q = I$ and $R = 1$. The disturbance is assumed to be bounded in $\mathcal{W} = \{w \mid \|w\|_2^2 \leq 0.04\}$. The terminal cost function $V_f(x)$ is taken as the unconstrained infinite horizon optimal value function for the nominal system with $P = [\begin{smallmatrix} 1.8085 & 0.2310 \\ 0.2310 & 2.6489 \end{smallmatrix}]$ and $\kappa_f(x) = Kx$ is the corresponding optimal LQ controller. Ellipsoidal approximations of $\mathcal{Z}_\mathcal{W}$ and $\bar{\mathcal{X}}_f^{tr}(x_s, u_s)$ were calculated as described in Section 8.5. Figure 8.1 illustrates a state trajectory $x(k)$ that is steered to the reference $x_r = [1, 0]^T$ starting from $x(0) = [-2, -2]^T$ under a sequence of extreme disturbances together with the corresponding trajectory of tube centers $\bar{x}_0(x(k))$, sets $\bar{x}_0(x(k)) \oplus \mathcal{Z}_\mathcal{W}$ and terminal sets $\bar{\mathcal{X}}_f^{tr}(\bar{x}_s(x(k)), \bar{u}_s(x(k))$. At $x(0)$ the reference x_r is infeasible and in-

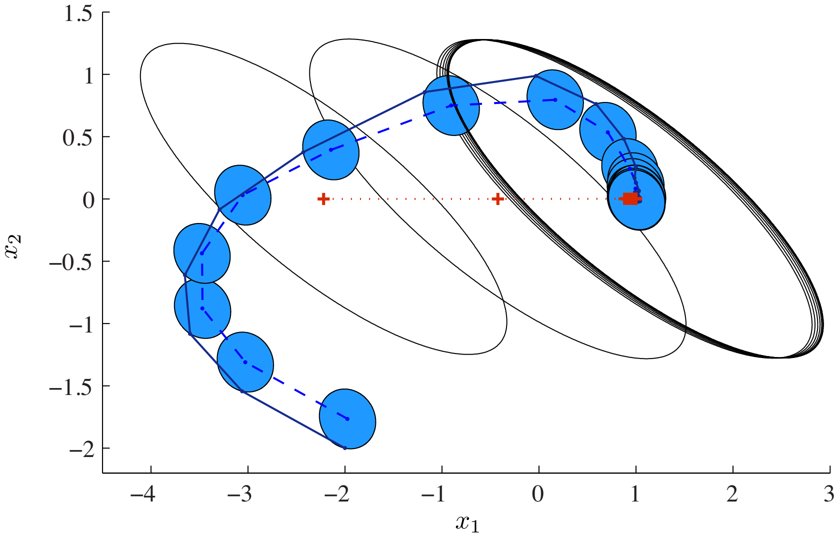


Figure 8.1: State trajectories for Example (8.7.1). The solid line is the actual trajectory $x(i)$ and the dashed line represents the trajectory of tube centers $\bar{x}_0(x(i))$. Crosses indicate the artificial reference $(\bar{x}_s(x(i)), \bar{u}_s(x(i)))$. The terminal sets $\bar{\mathcal{X}}_f^{tr}(\bar{x}_s(x(i)), \bar{u}_s(x(i))$ are shown as well as the sets $\bar{x}_0(x(i)) \oplus \mathcal{Z}_{\mathcal{W}}$.

termediate artificial references $\bar{x}_s(x(k))$ are computed, which are also the centers of the terminal sets and converge to x_r, indicated by the crosses in Figure 8.1.

8.7.2 Oscillating Masses Example

The oscillating masses example described in [WB10] is chosen to examine the proposed real-time method and evaluate it against that proposed in [WB10]. The considered model has $n_x = 12$ states and $n_u = 3$ inputs. Ellipsoidal invariant sets were computed for $\bar{\mathcal{X}}_f$ and $\mathcal{Z}_{\mathcal{W}}$, polytopic approximations cannot be computed for this problem size. For a horizon of $N = 30$ this results in a QCQP with 462 optimization variables and 1238 constraints. A random disturbance sequence with $\|w\|_2 \leq 0.25$ is acting on the system, which corresponds to 20% of the actuator's control range. We consider the regulation case described in Section 8.3, where the method was run with the same optimization parameters given in [WB10] and a fixed number of optimization steps $k_{\max} = 5$ in order to have a direct comparison with the reported results. The solver proposed in this work was able to compute 5 Newton steps in 6msec (averaged over 100 runs) and hereby achieves timings that are essentially equal to those reported in [WB10]. We can hence achieve the same fast sampling rates using the robust MPC

Table 8.1: Closed-loop performance deterioration in %

$k_{\max}$	1	2	3	4	5	6	7	8
ΔJ_{cl}	1.39	1.32	1.10	0.88	0.70	0.55	0.44	0.33

design and achieve guaranteed feasibility and stability. Both methods provide a closed-loop performance deterioration $\Delta J_{cl} < 1\%$ taken over a large number of sample points, where $\Delta J_{\mathrm{cl}} = \frac{\sum_{i=0}^{\infty}(l(x_i,\hat{\kappa}(x_i))-l(x_i,\kappa(x_i)))}{\sum_{i=0}^{\infty} l(x_i,\kappa(x_i))}$, $\hat{\kappa}(x)$ denotes the suboptimal controller obtained after $k_{\max}$ iterations and $\kappa(x)$ the optimal controller of the considered method. ΔJ_{cl} is estimated by simulating the trajectory for a long time period.

After establishing that the proposed approach performs equally well for the particular example it is important to note that one would choose the optimization parameters differently for the proposed method. A long horizon was taken in [WB10] since no stability guarantee is provided by the problem setup. This is however not necessary using the presented approach due to its a-priori stability guarantee. We therefore repeat the simulation with a horizon of $N = 10$ and investigate the effect of the number of allowed iterations on the closed-loop performance deterioration, reported in Table 8.1. It is important to note that the performance as well as the region of attraction are not affected by the reduction of the horizon to $N = 10$. One Newton step can now be computed in 0.3msec. Consequently, the real-time MPC method with $k_{\max} = 5$ iterations can be implemented with a sampling time of 2msec resulting in a controller rate of 500Hz. It is remarkable that the one step solution still shows considerably low performance loss. Since stability is guaranteed at all times one could therefore choose $k_{\max} = 1$ in order to achieve extremely low computation times of 0.3msec in trade for lower performance.

8.7.3 Large-Scale Example

A random example with $n_x = 30, n_u = 8$ and $N = 10$ was generated resulting in an optimization problem with 410 optimization variables and 1002 constraints. Ellipsoidal invariant sets were computed for $\bar{\mathcal{X}}_f$ and $\mathcal{Z}_{\mathcal{W}}$. We recorded the computation time for the invariant sets and tightened constraints, which were computed offline in only 19 seconds. The invariant set for tracking $\bar{\mathcal{X}}_f^{tr}(x_s, u_s)$ is computed by solving a large dimensional LMI in 35 seconds. The robust MPC problem with $k_{\max} = 5$ Newton iterations was solved in 10msec.

8.8 Conclusions

A new approach for real-time MPC was presented that provides guarantees on feasibility, stability and convergence for all time constraints using a robust MPC problem setup with a Lyapunov constraint that ensures the Lyapunov decrease property of the MPC cost. The computational details for an approximate warm-start interior-point method were described. The described real-time procedure represents a practical method that can be applied to systems of higher dimension and can be extended to provide reference tracking. The presented numerical examples illustrate the approach and show that a tailored custom solver achieves computation times in the range of milliseconds that are equal, or faster than those reported for methods without guarantees.

Part III

Soft Constrained MPC for Linear Systems

9 Soft Constrained MPC with Robust Stability Guarantees

9.1 Introduction

In control systems, there are generally two types of constraints: those originating from physical limitations of the actuators or the system itself and those that represent desired or critical bounds related to, for example, safety or particular system specifications. While input constraints can therefore never be exceeded and are considered as hard constraints, state or output constraints can either be hard if they fall under the first category or they are soft and may in practice be briefly violated if necessary, e.g. because of disturbances that are acting on the system.

In this chapter we propose a soft constrained MPC approach for linear systems that provides stability even for unstable systems. Soft constrained MPC approaches are based on the idea that, due to the nature of the state constraints, violation can often be tolerated for short time periods. Several methods for the development of controllers that enforce state constraints when they are feasible and allow for possible relaxation when they are not have been studied in the literature, see e.g. [Mac00] for an overview. In [RM93] a simple stabilizing strategy for infinite horizon MPC is proposed that can be applied to both stable and unstable systems. The authors in [ZM95] prove stability of MPC with hard input and soft state constraints for systems with eigenvalues in the closed unit disk. In [dOB94] the use of l_1-, l_∞-norm and quadratic penalties for constraint violation is compared and it is shown that l_1-norm penalties with a finite penalty parameter preserve the stability characteristics of the corresponding hard-constrained problem wherever the state constraints can be enforced. A comparison between soft constrained and minimum-time approaches is provided in [SMR99]. [KM00] propose the use of exact penalty functions in a soft constrained MPC approach in order to guarantee constraint satisfaction whenever possible. A soft constrained method for stochastic MPC is developed in [Pri07].

In contrast to soft constrained MPC, robust MPC methods design the control problem for an expected worst-case bound on the disturbance in order to ensure constraint satisfaction and robust stability, see e.g. [BM99, MRRS00, MS07, LAR+09] for an overview. The results can, however, be conservative due to the fact that the guarantees are only valid if the disturbance never exceeds the expected bound, which often requires a conservative choice of the considered disturbance set since worst-case bounds are difficult to obtain in practice.

The proposed method is based on a finite horizon MPC setup and uses a terminal weight as well as a terminal constraint. All input constraints are hard constraints and state constraints are softened in two ways. The terminal constraint is relaxed by allowing the origin to move to any steady-state satisfying only the input constraints. All other state constraints are softened by the introduction of two types of slack variables, which is a crucial item for proving stability. Quadratic and l_1- or l_∞-norm penalties for the constraint violations are introduced in the cost in order to allow for more flexibility in the problem formulation. The use of l_1 or l_∞ penalties allows for exact penalty functions which preserve the optimal MPC behavior whenever the state constraints can be enforced. The proposed problem setup results in a convex second-order cone program (SOCP) and can therefore be solved efficiently using standard methods for convex optimization, e.g. interior-point methods.

We show that, in contrast to existing soft constrained MPC schemes, asymptotic stability of the nominal system in the absence of disturbances is guaranteed even for unstable systems. The presented approach offers an enlarged region of attraction due to the constraint relaxation that, by choosing the prediction horizon accordingly, can cover any region of interest up to the maximum stabilizable set for the input-constrained system, i.e. all initial states for which there exists a feasible input at all times such that the state converges to the origin without considering the state constraints. The robust stability properties of the proposed soft constrained scheme are analyzed and input-to-state stability under additive disturbances is proven. A key advantage of the presented method over robust MPC approaches is that, while stability is formally guaranteed in a robust invariant set that depends on the considered disturbance size, the control law is defined everywhere in a large feasible set. In contrast, when using a robust MPC method the control law is only defined for a set of tightened constraints that is determined by the considered disturbance size.

It is shown how the presented soft constrained method can be directly combined with a robust MPC framework. The combined approach can be beneficial if the nature of the disturbance is such that a certain disturbance magnitude is constantly influencing the system, which is, however, exceeded from time to time. This can be exploited by accounting for one part of the disturbance by means of a robust MPC design. Input-to-state stability of the uncertain system under the combined robust soft constrained approach is proven.

A numerical example demonstrates the presented procedures and shows that the constraint relaxation enlarges the robust invariant set where stability can be guaranteed for various disturbance sizes and significant disturbances can be tolerated. The soft-constrained MPC method is applied to a large-scale example problem showing that the corresponding optimization problem can be solved with reasonably small computation times even for significant problem dimensions.

The outline of the chapter is as follows: After stating the problem addressed in this work in Section 9.2, Section 9.3 introduces the soft constrained MPC problem and its properties. Section 9.4 shows that the proposed control law is optimal wherever the state constraints can be enforced. Asymptotic stability of the nominal system under the proposed control law is then proven in Section 9.5 and Section 9.6 analyzes the robustness properties of the proposed scheme and proves input-to-state stability of the uncertain system under the nominal control law. Section 9.7 shows that all theoretical results extend to the combination of a soft constrained and robust MPC method. The properties and advantages of the presented soft constrained MPC approach are illustrated in Section 9.8 using numerical examples.

9.2 Problem Statement

In order to resolve the feasibility issues described in the introduction, state constraints are generally relaxed in practice. While a relaxation of only the state constraints in (5.1c) would preserve the stability properties of the nominal MPC setup, a hard terminal constraint represents a significant limitation and could render the optimization problem infeasible in the presence of disturbances. A long prediction horizon would have to be chosen in order to guarantee feasibility in a sufficiently large region of interest. A standard soft constrained approach that is frequently applied in practice is therefore to relax all state constraints in (5.1c) and (5.1d) by the introduction of slack variables ϵ_i, $i = 0, \ldots, N$. The amount of constraint violation is then minimized by including penalty functions $l_\epsilon(\epsilon_i)$ for all ϵ_i, $0 = 1, \ldots, N$ in the MPC cost.

Consider the discrete-time system in (4.4) that is subject to the state and input constraints $\mathbb{X} \triangleq \{x \mid G_x x \leq f_x\}$ and $\mathbb{U} \triangleq \{u \mid G_u u \leq f_u\}$, where $G_x \in \mathbb{R}^{m_x \times n_x}, f_x \in \mathbb{R}^{m_x}$, $G_u \in \mathbb{R}^{m_u \times n_u}, f_u \in \mathbb{R}^{m_u}$, and an ellipsoidal terminal constraint. Relaxing all state constraints including the terminal constraint, we obtain the following standard soft-constrained MPC problem:

$$\min_{\mathbf{x},\mathbf{u},\epsilon_1,\ldots,\epsilon_N} \sum_{i=0}^{N-1} l(x_i,u_i) + l_\epsilon(\epsilon_i) + V_f(x_N) + l_\epsilon(\epsilon_N) \tag{9.1a}$$

$$\text{s.t.} \quad G_u u_i \leq f_u \ , \tag{9.1b}$$

$$\begin{aligned} G_x x_i &\leq f_x + \epsilon_i \ , && (9.1c)\\ x_N^T T x_N &\leq 1 + \epsilon_N \ , && (9.1d)\\ (5.1b),\ \ &(5.1e)\ , && (9.1e) \end{aligned}$$

for $i = 0, \dots, N-1$, where T is a symmetric positive definite matrix. This soft constrained method does, however, not guarantee stability or constraint satisfaction even in the nominal case. The standard stability proof for the considered finite horizon MPC scheme shows that the optimal MPC cost is a Lyapunov function by using the solution from the previous sampling time together with a locally stable and feasible control law, see the outline of the proof of Theorem 5.4 and [MRRS00]. This stability proof fails for the soft constrained problem in (9.1) for two possible reasons. If the terminal state is outside of the region where the local control law is feasible, no feasible input sequence is available for proving a decrease in the cost. If the local control law is feasible but the state constraints are violated, a decrease in the cost can no longer be guaranteed due to the introduction of the penalties on ϵ_i into the cost function. In both cases it can no longer be shown that the optimal cost is a Lyapunov function.

This is illustrated in the following for the two-dimensional unstable example system:

Example 9.1.

$$x(k+1) = \begin{bmatrix} 1.2 & 1 \\ 0 & 1 \end{bmatrix} x(k) + \begin{bmatrix} 1 \\ 0.5 \end{bmatrix} u(k) + w(k)\ . \tag{9.2}$$

The prediction horizon was chosen to be $N = 5$, the constraints on the states and control inputs are $|x_1| \leq 5$,$|x_2| \leq 2$ and $|u| \leq 1$. A quadratic stage cost $l(x,u) = \|x\|_Q^2 + \|u\|_R^2$ with $Q = I$, $R = 1$ is chosen, $l_\epsilon(\epsilon) = \|\epsilon\|_S^2$ and $S = 100I$. The terminal cost function $V_f(x)$ is taken as the unconstrained infinite horizon optimal value function for the nominal system. The previously described nominal soft-constrained MPC problem in (9.1) was solved in a receding horizon fashion starting from the initial state $x(0) = [-13\ 5]^T$ outside of the feasible set of the corresponding hard constrained MPC problem under a sequence of small disturbances with $\|w(k)\|_\infty \leq 0.08\ \forall k \in \mathbb{N}$. The closed-loop simulation results in Figure 9.1 show that the state x_1 grows unbounded while the input saturates, demonstrating that the standard soft-constrained MPC scheme in (9.1) does not provide stability of the closed-loop system.

We introduce a new soft constrained MPC formulation in the next section, which provides optimality and constraint satisfaction wherever the state constraints can be enforced (Section 9.4), asymptotic stability in the nominal case (Section 9.5) and input-to-state stability in the presence of additive disturbances (Section 9.6).

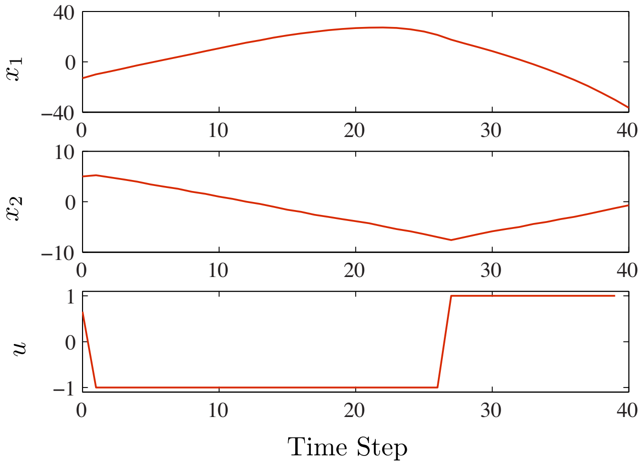


Figure 9.1: Closed-loop state and input trajectories using the standard soft constrained MPC approach in (9.1), where all state constraints are relaxed.

9.3 Soft Constrained MPC - Problem Setup

As discussed in Section 9.2, a stability guarantee by means of the standard stability proof in MPC has to be sacrificed in exchange for a complete relaxation of the terminal constraint. We therefore use a restricted relaxation by means of an enlarged terminal set in this work that is obtained by relaxing the origin as a regulation point to any other steady-state considering input constraints only. In addition, two different types of slack variables are employed, which will be key in proving (input-to-state) stability in a large feasible set.

We propose the following soft constrained MPC problem $\mathbb{P}_N^s(x)$:

Problem $\mathbb{P}_N^s(x)$ (Soft constrained MPC problem)

$$V_N^s(\mathbf{x},\mathbf{u},x_s,u_s,\boldsymbol{\epsilon}) \triangleq \sum_{i=0}^{N-1} l(x_i - x_s, u_i - u_s) + l_\epsilon(\epsilon_i) + V_f(x_N - x_s) + l_\epsilon(\epsilon_s) + V_o(x_s,u_s)$$

$$V_N^{s*}(x) = \min_{\mathbf{x},\mathbf{u},x_s,u_s,\boldsymbol{\epsilon}} V_N^s(\mathbf{x},\mathbf{u},x_s,u_s,\boldsymbol{\epsilon}) \tag{9.3a}$$

$$\text{subject to} \quad x_0 = x \ , \tag{9.3b}$$

$$x_{i+1} = Ax_i + Bu_i \ , \tag{9.3c}$$

$$G_u u_i \leq f_u \ , \tag{9.3d}$$

$$G_x x_i \leq f_x + \epsilon_s + \epsilon_i \ , \tag{9.3e}$$

$$\|T^{\frac{1}{2}}(x_N - x_s)\|_2^2 \leq 1 - r(x_s,u_s) \ , \tag{9.3f}$$

$$(I - A)x_s = Bu_s \ , \tag{9.3g}$$

$$G_u u_s \leq f_u \ , \tag{9.3h}$$

$$\epsilon_i \geq 0 \ , \tag{9.3i}$$

$$\epsilon_s \geq 0 \ , \tag{9.3j}$$

$$(1+\xi)G_x x_s \leq f_x + \epsilon_s \ , \tag{9.3k}$$

$$c\|T^{\frac{1}{2}}(x_N - x_s)\|_2 \leq f_x + \epsilon_s - G_x x_s \tag{9.3l}$$

for $i = [0, \ldots, N-1]$, where $\boldsymbol{\epsilon} = [\epsilon_0, \ldots, \epsilon_{N-1}, \epsilon_s]$ are the slack variables corresponding to the state sequence $\mathbf{x}$, $l(x,u) = \|x\|_Q^2 + \|u\|_R^2$ is the stage cost and $V_f(x) = \|x\|_P^2$ is a terminal cost function. The penalty function on the slack variables is taken as $l_\epsilon(\epsilon) = \|\epsilon\|_S^2 + \rho_1\|\epsilon\|_p$ and $V_o(x_s, u_s) = \rho_2\|x_s\|_p + \rho_2\|u_s\|_p$ is the offset cost function penalizing the deviation of (x_s, u_s) from the origin, where S is a symmetric positive definite matrix, $p \in \ \{1, \infty\}$ and $\rho_1, \rho_2 \in \mathbb{R}_+$ are positive constant weights. $r : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \rightarrow \mathbb{R}_+$ is assumed to be a quadratic function with $r(0,0) = 0$, $c_i = \|T^{-\frac{1}{2}} G_{x,i}^T\|_2 \, \forall \ i = 1, \ldots, m_x$ and T is defined in the description below. The set of feasible steady-states in the soft constrained case is given by $\Theta^s \triangleq \{(x_s, u_s) \mid u_s \in \mathbb{U}, (A-I)x_s + Bu_s = 0\}$.

The proposed soft constrained MPC setup is a modification of $\mathbb{P}_N(x)$ introducing the following three components:

- In (9.3f) the terminal constraint is relaxed by allowing the origin to move to any other feasible steady-state $(x_s, u_s) \in \Theta^s$, which can be considered as an artificial regulation point, using the approach presented in [LAAC08]. The terminal state then has to lie in an invariant set that is parameterized by the steady-state given by $\mathcal{X}_f^s(x_s, u_s)$, see also Section 5.4 for details on the method. In this work an invariant ellipsoidal target set is employed, given by

$$\mathcal{E}_{\mathcal{X}_f^s}(x_s, u_s) \triangleq \left\{x \ \middle| \ \|x - x_s\|_T^2 \leq 1 - r(x_s, u_s)\right\} \ . \tag{9.4}$$

- In (9.3e) all state constraints from 0 to $N-1$ are softened by means of the slack variables ϵ_s and ϵ_i:
ϵ_s captures the amount of constraint expansion that is necessary in order to include the ellipsoid $\mathcal{E}_T^s(x_N, x_s)$ in the softened state constraints, where

$$\mathcal{E}_T^s(x_N, x_s) \triangleq \left\{x \ \middle| \ \|x - x_s\|_T^2 \leq \|x_N - x_s\|_T^2\right\} \tag{9.5}$$

is a scaling of the ellipsoidal terminal set $\mathcal{E}_{\mathcal{X}_f^s}(x_s, u_s)$ containing x_N on its boundary. ϵ_i represents the additional constraint violation of each state x_i, for $i = 1, \ldots, N-1$, with respect to the state constraints relaxed by ϵ_s. Constraint (9.3k) additionally enforces that the steady-state x_s always has to lie in the interior of the constraints expanded by ϵ_s by an amount ξ, which is a user-specified small, positive parameter.

- Quadratic and l_1 or l_∞ penalties on the slack variables are included in the cost in (9.3), in order to minimize the constraint violation and ensure satisfaction of the state constraints whenever possible. An l_1 or l_∞ penalty on the steady-state is used, minimizing the deviation from the origin. In addition the cost is now designed for regulation around the artificial steady-state (x_s, u_s) and penalizes the deviation from the steady-state instead of the origin [LAAC08].

Assumption 9.1. It is assumed that for a given $(x_s, u_s) \in \Theta^s$, $V_f(x - x_s)$ is a Lyapunov function in $\mathcal{E}_{\mathcal{X}_f^s}(x_s, u_s)$ satisfying condition A2 in Assumption 5.3. Furthermore, $\mathcal{E}_{\mathcal{X}_f^s}(x_s, u_s)$ is a PI set under the local control law $\kappa_f^s(x) = K(x - x_s) + u_s$ satisfying the following assumptions:

A3: $\kappa_f^s(x) \in \mathbb{U} \; \forall x \in \mathcal{E}_{\mathcal{X}_f^s}(x_s, u_s)$

A4: $\|Ax + B\kappa_f^s(x) - x_s\|_T^2 = \|(A + BK)(x - x_s)\|_T^2 \leq \|x - x_s\|_T^2$.

Remark 9.2 (Terminal target set). Note that Assumption 9.1 only requires satisfaction of the input constraints and not the state constraints in the terminal target set $\mathcal{E}_{\mathcal{X}_f^s}(x_s, u_s)$, i.e. $\mathcal{E}_{\mathcal{X}_f^s}(x_s, u_s) \nsubseteq \mathbb{X}$.

Remark 9.3 (Hard state constraints). For ease of notation we assume the relaxation of all state constraints except the terminal constraint in the problem formulation $\mathbb{P}_N^s(x)$. The results presented in this chapter however directly extend to the case where some of the state constraints are considered as hard constraints with only minor notational changes.

Problem $\mathbb{P}_N^s(x)$ implicitly defines the set of feasible control sequences $\mathcal{U}_N^s(x, x_s, u_s) = \{\mathbf{u} \,|\, \exists\, \mathbf{x} \text{ s.t. } (9.3\text{b}) - (9.3\text{d}), (9.3\text{f}) \text{ hold}\}$ and feasible initial states $\mathcal{X}_N^s \triangleq \{x \,|\, \exists (x_s, u_s) \in \Theta^s \text{ s.t. } \mathcal{U}_N^s(x, x_s, u_s) \neq \emptyset\}$. For a given state $x \in \mathcal{X}_N^s$, Problem $\mathbb{P}_N^s(x)$ results in a convex Second Order Cone Program (SOCP) and its solution yields the optimal control sequence $\mathbf{u}^{s*}(x)$. Note that SOCPs can be efficiently solved using e.g. interior-point methods [BV04]. The implicit optimal soft constrained MPC control law is then given in a receding horizon fashion by

$$\kappa^s(x) \triangleq u_0^{s*}(x). \tag{9.6}$$

The components employed in the soft constrained problem setup $\mathbb{P}_N^s(x)$ are illustrated in Figure 9.2. The set of states, for which there exists an (x_s, u_s) such that $x \in \mathcal{E}_{\mathcal{X}_f^s}(x_s, u_s)$ is denoted as $\mathcal{C}_f^s$ and can be considered as an enlarged terminal set for the MPC problem, this is further explained in Section 9.3.1. By relaxing the terminal constraint, the soft constrained MPC regulates the state to an artificial steady-state that is simultaneously steered to the origin while minimizing the violation of the state

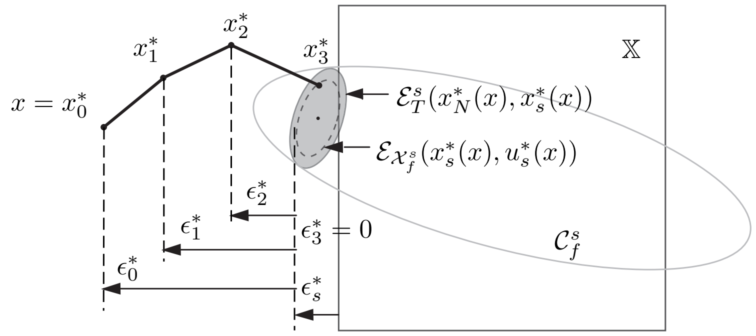


Figure 9.2: Illustration of the optimal slack variables $\epsilon_s^*(x)$, $\epsilon_i^*(x)$, $i = 0, 1, 2, 3$, the terminal set $\mathcal{E}_{\mathcal{X}_f^s}(x_s^*(x), u_s^*(x))$, $\mathcal{E}_T^s(x_N^*(x), x_s^*(x))$ and $\mathcal{C}_f^s$ for an initial state x outside $\mathbb{X}$.

constraints. The use of the slack variable ϵ_s ensures that the terminal state, which is contained in $\mathcal{E}_T^s(x_N, x_s)$, will lie inside the state constraints relaxed by the amount ϵ_s and will not require a further relaxation of the state constraints. This provides feasibility of the shifted sequence using the shifted slack variables with the last slack variable being zero. If the terminal state is close to x_s, a minimum amount of relaxation ξ with respect to the steady-state is ensured such that x_s does not lie too close to the boundary of the constraints expanded by ϵ_s. For a state that is close to the artificial steady-state, this ensures that the steady-state can always be shifted towards the origin without increasing the slack variables. As will be shown in Section 9.5, these items allow us to show that the optimal cost function is still a Lyapunov function and are hence crucial for proving stability of the proposed soft constrained MPC scheme.

Remark 9.4. While a strictly positive value of ξ in constraint (9.3k) is required to prove stability of the closed-loop system (Theorem 9.10), the particular choice of the parameter ξ is not crucial. Note that the size of ξ increases the value of ϵ_s, this effect will, however, be negligible for small values of ξ.

The remainder of this section is devoted to a detailed analysis of the two types of soft constraints. In the following sections we will then demonstrate how the introduction of the previously described components allows us to show that:

1. $\kappa^s(x) = \kappa(x)$ wherever the state constraints can be satisfied, i.e. for all $x \in \mathcal{X}_N$ (Section 9.4).
2. The optimal cost function $V_N^{s*}(x)$ is a Lyapunov function and the controlled nominal system is asymptotically stable with an enlarged region of attraction compared to a standard nominal MPC method (Section 9.5).

3. The system under additive disturbances is ISS (Section 9.6) with an enlarged region of attraction compared to a robust MPC approach considering the same disturbance size, which is demonstrated using numerical examples (Section 9.8).
4. The system under a combined robust and soft constrained approach is ISS (Section 9.7).

9.3.1 Relaxation of the Terminal Constraint

For the relaxation of the terminal set we make use of the formulation proposed in [LAAC08], which was introduced in the framework of a tracking approach but is employed for regulation in the proposed method. An artificial steady-state (x_s, u_s) is introduced as a decision variable into the optimization problem and the cost then penalizes the distance to the non-zero steady-state instead of the origin as well as the offset from the non-zero steady-state to the origin. The invariant set $\mathcal{E}_{\mathcal{X}_f^s}(x_s, u_s)$ in (9.5) is obtained from the terminal set for tracking introduced in Section 5.4.1 similar to the description in Section 8.5 and Algorithm 6. It is defined as the maximal PI set given by the set of states and steady-states (x_s, u_s), such that the control law $u = K(x - x_s) + u_s$ satisfies the input constraints and stabilizes the nominal system in (4.4).

Lemma 9.5 *The invariant ellipsoidal target set* $\mathcal{E}_{\mathcal{X}_f^s}(x_s, u_s)$ *in* (9.4) *can be computed by solving a convex linear matrix inequality (LMI).*

Proof. Consider a parametrization of the steady-state by the parameter θ, i.e. $x_s = M_x\theta, u_s = M_u\theta$, as described in Section 4.1. Using the augmented system $v^T = [x^T - x_s^T \quad \theta^T]$ and corresponding dynamics in (5.16), an ellipsoidal invariant set of the form $\Omega_f = \{v \in \mathbb{R}^n \,|\, v^T \left[\begin{smallmatrix} Q_1^{-1} & \\ & Q_2^{-1} \end{smallmatrix}\right] v \leq 1\}$ can be computed by solving a convex LMI [BV04], where only the input constraints are considered and the state constraints are neglected:

$$\left[\begin{smallmatrix} Q_1 & \\ & Q_2 \end{smallmatrix}\right] = \underset{Q_1,Q_2}{\operatorname{argmin}} \Big\{ -\log\det\left(\left[\begin{smallmatrix} Q_1 & \\ & Q_2 \end{smallmatrix}\right]\right) \;\Big|\; \left[\begin{smallmatrix} Q_1 & Q_1 A_K^T \\ A_K Q_1 & Q_1 \end{smallmatrix}\right] \succeq 0,$$
$$\|Q_1^{\frac{1}{2}} K^T G_{u,j}^T\|_2^2 + \|Q_2^{\frac{1}{2}} M_u^T G_{u,j}^T\|_2^2 \leq f_{u,j}^2 \forall\, j = 1, \dots, m_u \,\Big\} \;\;,$$

with $A_K = A + BK$. Using the relationship $\theta = M_x^T x_s + M_u^T u_s$ this can be directly transformed into the ellipsoidal terminal set $\mathcal{E}_{\mathcal{X}_f^s}(x_s, u_s)$ in (9.4) with $T = Q_1^{-1}$ and $r(x_s, u_s) = (M_x^T x_s + M_u^T u_s)^T Q_2^{-1} (M_x^T x_s + M_u^T u_s)$. ■

This also characterizes the set of all values x for which there exists a steady-state satisfying the terminal constraint, given by $\mathcal{C}_f^s = \{x \mid \exists\, (x_s, u_s) \in \Theta^s : x \in \mathcal{E}_{\mathcal{X}_f^s}(x_s, u_s)\}$, which can be seen as an enlarged terminal set providing an enlarged feasible set $\mathcal{X}_N^s \supseteq \mathcal{X}_N$ of the soft constrained MPC problem.

9.3.2 Slack Variables

We now explain the crucial item in the proposed soft constrained scheme, the use of the slack variables ϵ_s and ϵ_i.

Remark 9.6. By Assumption 9.1, the set $\mathcal{E}_T^s(x_N, x_s)$ is a positively invariant set under the local control law $\kappa_f^s(x)$.

ϵ_s represents the amount of constraint relaxation that is necessary in order to include the set $\mathcal{E}_T^s(x_N, x_s)$ for a particular value of (x_s, u_s) into the relaxed state constraints, i.e.

$$\max\left\{G_{x,i}x \;\middle|\; \|x - x_s\|_T^2 \leq \|x_N - x_s\|_T^2\right\} \leq f_{x,i} + \epsilon_{s,i} \ ,$$

for all $i = 1, \ldots, m_x$. Using the variable transformation $y = T^{\frac{1}{2}}(x - x_s)$ this can be written as

$$\max\{G_{x,i}T^{-\frac{1}{2}}y + G_{x,i}x_s \mid y^T y \leq \|x_N - x_s\|_T^2\} \leq f_{x,i} + \epsilon_{s,i} \ ,$$

for all $i = 1, \ldots, m_x$, which can be expressed by the following condition [BEFB94]:

$$\|T^{-\frac{1}{2}}G_{x,j}^T\|_2 \|T^{\frac{1}{2}}(x_N - x_s)\|_2 \leq f_{x,j} + \epsilon_{s,j} - G_{x,j}x_s, \ \forall\, j = 1, \ldots, m_x$$

and therefore corresponds to (9.3l), which is a collection of m_x convex second order cone constraints.

ϵ_i in (9.3f) represents the additional constraint violation of each state x_i with respect to the state constraints relaxed by ϵ_s. Let ϵ_N be the slack variable of the terminal state defined by $G_x x_N \leq f_x + \epsilon_s + \epsilon_N$. Since $x_N \in \mathcal{E}_T^s(x_N, x_s)$ it follows from this and (9.3l), that $\epsilon_N = 0$, which will be necessary for proving that the cost function is a Lyapunov function in Section 9.5.

9.4 Optimality in $\mathcal{X}_N$

In this section we show that the behavior of the soft constrained control law corresponds to the hard constrained one, wherever the state constraints can be satisfied. The constraint violations ϵ_s and ϵ_i are penalized in the cost. Two types of penalty functions are included, quadratic and l_1 or l_∞-norm penalties, in order to allow for flexibility in modeling the soft constraints. While the quadratic penalty may be preferred for tuning purposes, the l_1 or l_∞-norm is included in order to allow for exact penalty functions. As described in Section 3.1.3, it is well-known that, when the weights on the l_1 or l_∞-norms are sufficiently large and there exists a feasible solution to the hard constrained problem $\mathbb{P}_N(x)$, then the solution to the soft constrained problem $\mathbb{P}_N^s(x)$ corresponds to that of the hard constrained problem [Fle87, Lue84]. An l_1 or l_∞-norm is also used for penalizing the deviation of the artificial steady-state from the origin in order to enforce the origin as the regulation point if it is feasible [FLA+09].

Consider the following optimization problem $\mathbb{P}_N^h(x)$, enforcing all state constraints as hard constraints:

$$\min_{\mathbf{x},\mathbf{u},x_s,u_s,\boldsymbol{\epsilon}} \quad V_N^s(\mathbf{x},\mathbf{u},x_s,u_s,\boldsymbol{\epsilon}) \tag{9.7a}$$

$$\text{s.t. } (9.3\text{b}) - (9.3\text{l}) \ , \tag{9.7b}$$

$$\epsilon_s = 0 \tag{9.7c}$$

$$\epsilon_i = 0, \ i = 0, \dots, N-1 \ , \tag{9.7d}$$

$$x_s = 0, u_s = 0 \ . \tag{9.7e}$$

Note that the optimizer of $\mathbb{P}_N^h(x)$ corresponds to the optimizer of $\mathbb{P}_N(x)$.

Theorem 9.7 (Optimality in $\mathcal{X}_N$ [Fle87]) *Consider problem $\mathbb{P}_N^h(x)$. Let $\lambda_{ss}^*(x)$ denote the optimal Lagrange multipliers corresponding to the constraints in* (9.7e) *and $\lambda_{\boldsymbol{\epsilon}}^*(x)$ the optimal Lagrange multipliers corresponding to the equality constraints* (9.7d) *and* (9.7c) *satisfying the KKT conditions in* (3.4) *at a given state $x \in \mathcal{X}_N$. Let $\kappa^s(x)$ be the optimal soft constrained control law in* (9.6) *and $\kappa^h(x)$ the optimal hard constrained control law in* (5.4)*. Let $\|\cdot\|_D$ denote the corresponding dual norm to $\|\cdot\|_p$ for $p \in \{1, \infty\}$. If $\lambda_{ss}^*(x), \lambda_{\boldsymbol{\epsilon}}^*(x)$ are bounded (i.e. the primal optimal solution is feasible), $\rho_1 \geq \|\lambda_{ss}^*(x)\|_D$ and $\rho_2 \geq \|\lambda_{\boldsymbol{\epsilon}}^*(x)\|_D$ for all $x \in \mathcal{X}_N$, then $\kappa^s(x) = \kappa^h(x)$ for all $x \in \mathcal{X}_N$.*

Theoretically, the existence of a conservative state-dependent bound can be shown based on Lipschitz continuity of the convex optimization problem in (9.7) [Hag79]. A lower bound for ρ_1 and ρ_2 can e.g. be obtained by computing the optimal Lagrange multipliers parametrically for all $x \in \mathcal{X}_N$ [KM00], which can, however, be computationally intractable for large systems. A lower bound providing optimality in the terminal set for tracking $\mathcal{E}_{\mathcal{X}_f^s}(x_s, u_s)$ that can be obtained by means of a linear program was presented in [FLA+08]. The efficient computation of a bound on the exact multiplier is, however, still an active topic of research. In practice, approximate values for the parameters ρ_1 and ρ_2 are often chosen by simulation.

9.5 Nominal Stability

After having shown that the proposed soft constrained scheme preserves optimality whenever possible, we now prove that the resulting optimal soft constrained control law $\kappa^s(x)$ in (9.6) asymptotically stabilizes the nominal system in (4.4) in an enlarged positively invariant set $\mathcal{X}_N^s$. For this, we show that the optimal cost function of the soft-constrained MPC problem V_N^{s*} is a Lyapunov function in two stages.

Lemma 9.8 *Consider Problem $\mathbb{P}_N^s(x)$ under Assumption 9.1. Let $\mathbf{u}^{s*}(x)$, $\mathbf{x}^{s*}(x)$, $x_s^{s*}(x)$, $u_s^{s*}(x)$, $\boldsymbol{\epsilon}^{s*}(x)$ be the optimizer of $\mathbb{P}_N^s(x)$ for some $x \in \mathcal{X}_N^s$. The shifted control sequence*

$$\mathbf{u}^{shift}(x) = [u_1^{s*}(x), \ldots, u_{N-1}^{s*}(x), \tilde{u}(x)] \ , \tag{9.8}$$

with $\tilde{u}(x) = K(x_N^{s*}(x) - x_s^{s*}(x)) + u_s^{s*}(x)$ is feasible for $\mathbb{P}_N^s(x^+)$ with $x^+ = Ax + B\kappa^s(x)$ and

$$V_N^{s*}(x^+) - V_N^{s*}(x) \leq -l(x - x_s^{s*}(x), u_0^{s*}(x) - u_s^{s*}(x)) \ . \tag{9.9}$$

Proof. For ease of notation we drop the dependence of the sequences on x in the following proof. The states and slack variables associated with $\mathbf{u}^{\text{shift}}$ in (9.8) are: $\mathbf{x}^{\text{shift}}$, $\boldsymbol{\epsilon}^{\text{shift}} = [\epsilon_1^{s*}, \ldots, \epsilon_{N-1}^{s*}, 0, \epsilon_s^{s*}]$, where $\epsilon_N^{s*} = 0$ follows from the fact that $x_N^{s*} \in \mathcal{E}_T^s(x_N^{s*}, x_s^{s*})$ and the definition of the slack variables in (9.7d) and (9.7c). Feasibility of $\mathbf{u}^{\text{shift}}$ for $\mathbb{P}_N^s(x^+)$ then follows from feasibility of $\mathbf{u}^{s*}, x_s^{s*}, u_s^{s*}, \boldsymbol{\epsilon}^{s*}$ at x and positive invariance of $\mathcal{E}_T^s(x_N^{s*}, x_s^{s*})$. Together with the definition of the sequences and Assumption 9.1 this provides that $V_N^s(\mathbf{x}^{\text{shift}}, \mathbf{u}^{\text{shift}}, x_s^{s*}, u_s^{s*}, \boldsymbol{\epsilon}^{\text{shift}}) - V_N^{s*}(x) \leq -l(x - x_s^{s*}, u_0^{s*} - u_s^{s*})$ using standard arguments and (9.9) follows from $V_N^{s*}(x^+) \leq V_N^s(\mathbf{x}^{\text{shift}}, \mathbf{u}^{\text{shift}}, x_s^{s*}, u_s^{s*}, \boldsymbol{\epsilon}^{\text{shift}})$. ■

Lemma 9.8 shows that the closed-loop system converges to x_s^{s*}. In order to achieve asymptotic convergence to the origin and not to a non-zero steady-state, we now need to show that x_s^{s*} simultaneously converges to zero, i.e. that $l(x-x_s^{s*}(x), u_0^{s*}(x)-u_s^{s*}(x)) = 0$ only if $x = 0$.

Lemma 9.9 *If at a given state x the optimal solution to $\mathbb{P}_N^s(x)$ is such that $\|x - x_s^{s*}(x)\| = 0$, then $x = 0$.*

Proof. We first sketch a modification of the proof of Lemma 3 in [LAAC08], which is proven by contradiction and then extend it to the case considered here.

In the first part of the proof it is shown that for every feasible steady-state $(x_s, u_s) \neq (0, 0)$ there exists a steady-state $(\alpha x_s, \alpha u_s)$ with $\alpha \in [0, 1)$ such that $x_s \in \mathcal{E}_{\mathcal{X}_f^s}(\alpha x_s, \alpha u_s)$. Therefore the control sequence $\mathbf{u}_\alpha(x_s)$ generated by $u_i = K(x_i - \alpha x_s) + \alpha u_s$ with $x_0 = x_s$ is feasible at x_s. Let $\mathbf{x}_\alpha(x_s)$ be the state sequence corresponding to $\mathbf{u}_\alpha(x_s)$.

In the second part of the proof it is shown that, if the current state is x_s, then the cost to move to αx_s applying $\mathbf{u}_\alpha(x_s)$ is in fact smaller than the cost of staying at x_s

by applying u_s over the entire horizon and therefore staying at x_s cannot be the optimal solution. In [LAAC08, FLA$^+$09] it is shown that $V_N(x_s, \mathbf{u}_\alpha(x_s)) + V_o(\alpha x_s, \alpha u_s) < V_N(x_s, \mathbf{u}_s) + V_o(x_s, u_s)$, where V_N is the MPC cost without slack variables in 5.1a and $\mathbf{u}_s = [u_s, \dots, u_s]$.
In order to prove that this extends to the soft constrained approach we need to show that also $V_N^s(x_s, \mathbf{u}_\alpha(x_s), \alpha x_s, \alpha u_s, \boldsymbol{\epsilon}_\alpha) < V_N^s(x_s, \mathbf{u}_s, x_s, u_s)$ or

$$\sum_{i=0}^{N-1} l_\epsilon(\epsilon_{\alpha,i}) + l_\epsilon(\epsilon_{\alpha,s}) \leq \sum_{i=0}^{N-1} l_\epsilon(\epsilon_i) + l_\epsilon(\epsilon_s) \ , \tag{9.10}$$

where ϵ_s and ϵ_i are the minimum slacks corresponding to the state sequence $\mathbf{x}_s = [x_s, \dots, x_s]$ and $\epsilon_{\alpha,s}$ and $\epsilon_{\alpha,i}$ are the minimum slacks corresponding to the state sequence $\mathbf{x}_\alpha(x_s)$ defined by the constraints (9.3e) and (9.3l). This is achieved by the fact that ϵ_s and $\epsilon_{\alpha,s}$ characterize the state relaxation necessary in order to include the sets $\mathcal{E}_T^s(x_s, x_s)$ and $\mathcal{E}_T^s(x_{\alpha,N}(x_s), \alpha x_s)$, respectively, into the softened state constraints, where $x_{\alpha,N}(x_s)$ denotes the terminal state of the sequence $\mathbf{x}_\alpha(x_s)$. (9.3k) provides a minimum amount of distance between x_s and the state constraints expanded by the amount ϵ_s allowing to move the center to a point αx_s without increasing the slacks.
Condition (9.10) will be proven in the following by showing that $\epsilon_{\alpha,s} \leq \epsilon_s$ and $\epsilon_{\alpha,i} \leq \epsilon_i$.

We recall that we now have two constraints on ϵ_s:

$$\epsilon_s \geq -f_x + (1+\xi)G_x x_s, \quad \epsilon_s \geq c\|T^{\frac{1}{2}}(x_N - x_s)\|_2 - f_x + G_x x_s \ .$$

For $x = x_s$, we have $\epsilon_s = [-f_x + (1+\xi)G_x x_s]_+$. We then choose $\alpha \in [0, 1)$ big enough such that

$$\alpha \xi G_x x_s \geq (1-\alpha)c\|T^{\frac{1}{2}} x_s\|_2 \ . \tag{9.11}$$

Using $\beta \geq 1$ such that $c\|T^{\frac{1}{2}} x_s\|_2 \leq \beta G_x x_s$ this condition is fulfilled if $\alpha \geq \frac{\beta}{\beta+\xi}$. Note that $f_x > 0$ since the origin is included in the interior of the state constraints. Therefore, only the case when $G_x x_s \geq 0$ is relevant, since otherwise $\epsilon_s = 0$, i.e. the constraint is not violated, and $(1-\alpha)c\|T^{\frac{1}{2}} x_s\|_2 \leq f_x$ can be satisfied for a small value of α. Since by Assumption 9.1

$$\|T^{\frac{1}{2}}(x_{\alpha,N} - \alpha x_s)\|_2 = \|T^{\frac{1}{2}}((A+BK)^N(x_s - \alpha x_s)\| \leq (1-\alpha)\|T^{\frac{1}{2}} x_s\|_2 \ ,$$

we have from this and (9.11) that $\epsilon_{\alpha,s} = [\alpha(1+\xi)G_x x_s - f_x]_+$ and $\epsilon_{\alpha,s} \leq \epsilon_s$ follows directly. $\epsilon_{\alpha,i} = 0$ for $i = 1, \dots, m_x$ then follows from the fact that $(1-\alpha)\|T^{\frac{1}{2}} x_s\|_2 \geq \|T^{\frac{1}{2}}(x_{\alpha,i} - \alpha x_s)\|_2$, showing that the ellipse with center αx_s and radius $\|T^{\frac{1}{2}} x_{i,\alpha} - \alpha x_s\|_2$ is contained in the constraints relaxed by $\epsilon_{\alpha,s}$ and therefore also $x_{i,\alpha}$.

We therefore have $\epsilon_{\alpha,s} \leq \epsilon_s, \epsilon_{\alpha,i} = 0$, which concludes the proof. ■

We can now state one of the main results of this work and prove asymptotic stability of the closed loop system under the soft constrained MPC control law:

Theorem 9.10 (Asymptotic Stability under $\boldsymbol{\kappa^s(x)}$) *The closed loop system $x(k+1) = Ax(k) + B\kappa^s(x(k))$ is asymptotically stable with region of attraction $\mathcal{X}_N^s$.*

Proof. We have that $V_N^{s*}(x) \geq l(x, u_0^{s*}(x)) \geq \underline{\alpha}(\|x\|)\ \forall x \in \mathcal{X}_N^s$ and by optimality of $V_N^{s*}(x)$ in $\mathcal{X}_N$ (Theorem 9.7) and the fact that $V_N^*(x)$ is a Lyapunov function (Theorem 5.4) we obtain $V_N^{s*}(x) \leq V_f(x) \leq \overline{\alpha}(\|x\|)\ \forall x \in \mathcal{E}_{\mathcal{X}_f^s}(0,0)$, where $\underline{\alpha}(\bullet), \overline{\alpha}(\bullet)$ are suitable $\mathcal{K}_\infty$-class functions. From Lemma 9.8 we obtain

$$V_N^{s*}(x^+) \leq V_N^{s*}(x) - \|x - x_s^{s*}(x)\|_Q^2\ \forall x \in \mathcal{X}_N^s \text{ with } x^+ = Ax + B\kappa^s(x)\ .$$

By optimality and Lemma 9.9 the optimal regulation point is $x_s^{s*}(x) = 0$ if $x \in \mathcal{X}_N$. If $x \notin \mathcal{X}_N$ we can assume $\|x - x_s^{s*}(x)\|_Q \geq \hat{\delta}$ for some small positive constant $\hat{\delta} \in (0,1]$. In this case we can choose

$$\delta(x) = \frac{\|x - x_s^{s*}(x)\|_Q^2}{\|x\|_Q^2} \geq \frac{\hat{\delta}^2}{\|x\|_Q^2} \text{ and } \delta_{\min} = \frac{\hat{\delta}^2}{\max_{x \in \mathcal{X}_N^s} \|x\|_Q^2}\ ,$$

which provides that $V_N^{s*}(x^+) - V_N^{s*}(x) \leq -\delta_{\min}\|x\|_Q^2\ \forall x \in \mathcal{X}_N^s$, with $\delta_{\min} > 0$. Therefore, there exists a $\mathcal{K}_\infty$-class function $\beta(\bullet)$ such that $V_N^{s*}(x^+) - V_N^{s*}(x) \leq -\beta(\|x\|)$ in $\mathcal{X}_N^s$ and $V_N^{s*}(x)$ is a Lyapunov function in $\mathcal{X}_N^s$ proving asymptotic stability with region of attraction $\mathcal{X}_N^s$ by Theorem 4.11. ■

The soft constrained formulation $\mathbb{P}_N^s(x)$ enlarges the feasible set compared to $\mathbb{P}_N(x)$ since $\mathcal{X}_N \subseteq \mathcal{X}_N^s$ and by choosing the prediction horizon accordingly it can be chosen to cover any polytopic region of interest, e.g. a known upper bound on the state values, up to the maximum stabilizable set for the input-constrained system.

Corollary 9.11 *Let $\beta\mathbb{X}$, with $\beta \geq 1$, be a scaling of the state constraints and $\mathcal{X}_\infty := \{x \mid \exists u(k) \in \mathbb{U} : \lim_{j\to\infty}, x(j) = 0, x(0) = x, x(k+1) = Ax(k) + Bu(k), \forall k \geq 0\}$ the set of all stabilizable states for the input-constrained system. There exists a finite prediction horizon $\bar{N}$ such that $\mathcal{X}_{\bar{N}}^s \supseteq \beta\mathbb{X} \cap \mathcal{X}_\infty$.*

9.6 Robust Stability

In practice, model uncertainties or external disturbances cause a deviation from the nominal system dynamics in (4.4). The question is then if the control law that was designed for the nominal system model is still stabilizing the uncertain system in (4.1), or so-called robustly stable. This issue has been studied in the literature, see e.g. [LAR+09, GMTT04] and the references therein. While in general, nominal MPC control laws can have zero robust stability margin, it was shown that particularly for linear systems the nominal MPC control law offers robust stability in an RPI set under certain assumptions on the MPC problem setup and for a sufficiently small bound on the disturbance size [LAR+09, GMTT04].

Another possibility to handle uncertainties is to take them explicitly into account using robust MPC schemes that provide robust stability by changing the problem formulation and tightening the constraints, e.g. min-max MPC or tube-based approaches (see the discussion in Section 5.3 or [MRRS00, LAR+09, MSR05, RM09] and the references therein). The disadvantage of robust MPC methods is, however, that a bound on the disturbance size is assumed and the problem is designed for the worst-case disturbance. If the disturbance is then significantly smaller than the worst-case bound for most of the time, the solutions can be highly conservative. If, on the other hand, the disturbance exceeds the expected bound, then the robust MPC problem may be infeasible and cannot provide a control input.

When using a nominal MPC scheme, no knowledge on the size of the uncertainty is required for the computation of the controller and it may even be able to take advantage of a disturbance in the right direction. Conservatism is however introduced when analyzing the RPI set, where robust stability of a nominal MPC scheme can be guaranteed, since it is based on a particular bound on the disturbance size. In the presence of hard constraints, the RPI set may be prohibitively small for the considered disturbance size. The idea is therefore that by using the proposed soft constrained approach robust stability can be guaranteed in a much larger RPI set due to the fact that state constraints can be relaxed. It is however important to note that, while stability is formally only guaranteed within the RPI set, the control law is defined everywhere in a large feasible set (see Corollary 9.11) and may still stabilize the uncertain system for a variety of disturbance signals. Note that no RPI sets need to be computed in order to apply the proposed method.

The robust stability properties of the proposed soft constrained MPC scheme are analyzed in the following using the framework of input-to-state stability introduced in Section 4.2. Assume that the system is subject to an additive uncertainty as given in (4.1). Because of the disturbance, the shifted sequence $\mathbf{u}^{\text{shift}}(x)$ in (9.8) may no longer be feasible for $\mathbb{P}_N^s(x(k+1))$. For all $x(k+1) \in \mathcal{X}_N^s$ there does however exist a feasible solution to $\mathbb{P}_N^s(x(k+1))$ and input-to-state stability can be shown in an RPI set $\mathcal{X}_\mathcal{W}^s \subset \mathcal{X}_N^s$. It is given by the robust positively invariant set for the controlled uncertain system $x(k+1) = Ax(k) + B\kappa^s(x(k)) + w(k)$, where $\kappa^s(x)$ is the optimal soft constrained MPC control law in (9.6):

$$Ax + B\kappa^s(x) + w \in \mathcal{X}_\mathcal{W}^s \;\forall\; x \in \mathcal{X}_\mathcal{W}^s,\; w \in \mathcal{W}. \tag{9.12}$$

In order to show that the uncertain system in (4.1) is input-to-state stable with respect to the (unspecified) disturbance set $\mathcal{W}$ under the nominal control law, we make use of the following result:

Lemma 9.12 (Continuity of $\boldsymbol{V_N^{s*}(x)}$) *Consider the optimization problem $\mathbb{P}_N^s(x)$. The optimal value function $V_N^{s*}(x)$ is continuous on $\mathcal{X}_N^s$.*

Proof. Follows directly from continuity and convexity of the cost function and the constraints in (9.3) as well as compactness of the constraint set for all $x \in \mathcal{X}_N$ (Theorem 4.3.3 in [BGK$^+$82]). ∎

This allows us to extend the results on asymptotic stability for the nominal system in Section 9.5 and prove input-to-state stability of the uncertain system controlled by $\kappa^s(x)$ in (9.6).

Theorem 9.13 (ISS under $\boldsymbol{\kappa^s(x)}$) *The closed loop system $x(k+1) = Ax(k)+B\kappa^s(x(k))+w(k)$ is ISS w.r.t to $w(k) \in \mathcal{W}$ with region of attraction $\mathcal{X}_{\mathcal{W}}^s$.*

Proof. From the proof of Theorem 9.10 and Lemma 9.12 it follows that $V_N^{s*}(x)$ is a continuous Lyapunov function and hence there exists a $\mathcal{K}$-class function $\gamma(\boldsymbol{\cdot})$, such that $|V_N^{s*}(y) - V_N^{s*}(x)| \leq \gamma(\|y-x\|)$ as well as a $\mathcal{K}_\infty$-class function $\beta(\boldsymbol{\cdot})$ such that $V_N^{s*}(Ax(k) + B\kappa^s(x(k))) - V_N^{s*}(x(k)) \leq -\beta(\|x(k)\|)$. It follows from these facts that

$$\begin{aligned}
&V_N^{s*}(x(k+1)) - V_N^{s*}(x(k)) \\
&= V_N^{s*}(Ax(k) + B\kappa^s(x(k)) + w) - V_N^{s*}(Ax(k) + B\kappa^s(x(k))) \\
&\quad + V_N^{s*}(Ax(k) + B\kappa^s(x(k))) - V_N^{s*}(x(k)) \\
&\leq -\beta(\|x(k)\|) + |V_N^{s*}(Ax(k) + B\kappa^s(x(k)) + w) - V_N^{s*}(Ax(k) + B\kappa^s(x(k)))| \\
&\leq -\beta(\|x(k)\|) + \gamma(\|w(k)\|) \ .
\end{aligned}$$

Then $V_N^{s*}(x(k))$ is an ISS-Lyapunov function with respect to $w(k) \in \mathcal{W}$ and by Theorem 4.14 the closed-loop system is ISS. ∎

The uncertain system controlled by the control law resulting from the soft constrained MPC problem $\mathbb{P}_N^s(x)$ is hence input-to-state stable against sufficiently small disturbances. Since the RPI set $\mathcal{X}_{\mathcal{W}}^s$ depends on $\mathcal{W}$, the size of the disturbances and the corresponding region for which stability can be formally guaranteed depend on the particular system of interest.

We now revisit our introductory Example 9.1 and show that the proposed soft-constrained MPC scheme provides input-to-state stability of the closed-loop system. The results are shown in Figure 9.3 for the same disturbance bound $\|w\|_\infty \leq 0.08$.

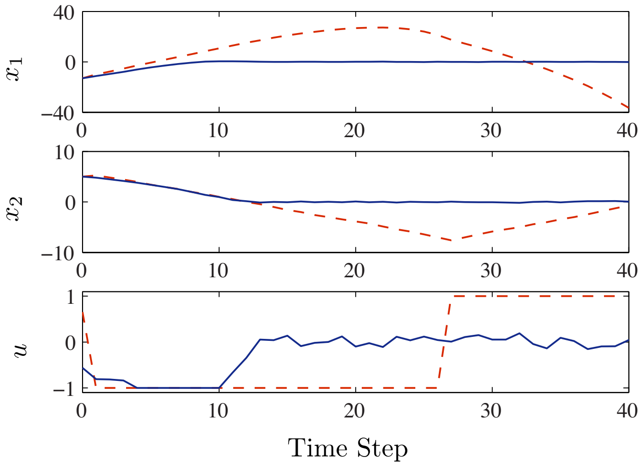


Figure 9.3: Closed-loop state and input trajectories using the standard soft constrained MPC approach in (9.1), where all state constraints are relaxed, (dashed line) and the proposed soft constrained MPC method in (9.3) (solid line).

In the following section we will show that the previously presented results can be directly extended to the combination of a robust and soft constrained MPC framework allowing for a more flexible disturbance handling.

9.7 Combination with Robust MPC

The soft constrained MPC approach introduced in Section 9.3 is well suited for disturbances that are irregular or of varying magnitude. If it is, however, known that the disturbance has the particular characteristic that a certain disturbance size is constantly affecting the system while a larger disturbance size only occurs irregularly, it can be beneficial to combine a robust and soft constrained MPC procedure. Whereas designing a robust MPC approach for the worst case disturbance size would result in a highly conservative solution in this case, it can be used to take into account a disturbance up to a certain size. If the disturbance exceeds this bound, the idea is again that the use of relaxed state constraints will allow us to guarantee eventual recovery of feasibility and input-to-state stability of the uncertain system in a large RPI set covering a region of interest by using a combination of a robust and soft-constrained approach.

We assume in the following that the system is subject to two types of uncertainty:

$$x(k+1) = Ax(k) + Bu(k) + w_1(k) + w_2(k) \ , \tag{9.13}$$

where $w_1(k) \in \mathcal{W}_1 \subset \mathbb{R}^n$ is a bounded disturbance that is expected to occur frequently with maximum size and $w_2(k) \in \mathcal{W}_2 \subset \mathbb{R}^n$ captures the additional part of the disturbance with varying magnitude and occurrence. $\mathcal{W}_1, \mathcal{W}_2$ are convex and compact sets that each contain the origin.

Since the disturbance w_1 will constantly affect the system, it is explicitly taken into account using a robust MPC framework providing constraint satisfaction and stability in the presence of w_1. In this work, we consider the tube-based robust MPC approach for linear systems [MSR05] described in Section 5.3. The method is based on the use of a feedback policy of the form $u = \bar{u} + K(x - \bar{x})$ that bounds the effect of the disturbance w_1 and keeps the states x of the uncertain system under w_1 with $w_2 = 0$ close to the states of the nominal system in (4.4). The use of tightened state and input constraints $\bar{\mathbb{X}} = \mathbb{X} \ominus \mathcal{Z} \triangleq \{x \mid G_x x \le \bar{f}_x\}, \bar{\mathbb{U}} = \mathbb{U} \ominus K\mathcal{Z} \triangleq \{u \mid G_u u \le \bar{f}_u\}$, $\mathcal{E}_{\bar{\mathcal{X}}_f^s}(x_s, u_s) = \mathcal{E}_{\mathcal{X}_f^s}(x_s, u_s) \ominus \mathcal{Z} \triangleq \{x \mid \|x - x_s\|_T^2 \le 1 - \bar{r}(x_s, u_s)\}$ ensures feasibility of the uncertain system in (9.13) despite the disturbance w_1, where $\mathcal{Z}$ is an RPI set for the controlled system $x(k+1) = (A + BK)x(k) + w_1(k)$. See [MSR05] for a detailed description of the method.

The robust soft constrained problem is directly obtained from $\mathbb{P}_N^s$ by replacing (9.3b) with $x \in x_0 \oplus \mathcal{Z}$, f_x, f_u in (9.3e),(9.3d) with $\bar{f}_x, \bar{f}_u$ and $r(\bullet, \bullet)$ with $\bar{r}(\bullet, \bullet)$:

Problem $\mathbb{P}_N^{rs}(x)$ (Robust soft constrained MPC problem)

$$V_N^{rs*}(x) = \min_{\bar{\mathbf{x}}, \bar{\mathbf{u}}, \bar{x}_s, \bar{u}_s, \bar{\boldsymbol{\epsilon}}} V_N^s(\bar{\mathbf{x}}, \bar{\mathbf{u}}, \bar{x}_s, \bar{u}_s, \bar{\boldsymbol{\epsilon}}) + V_f(x - \bar{x}_0) \tag{9.14a}$$

$$\begin{aligned}
\text{subject to} \quad & x && \in \bar{x}_0 \oplus \mathcal{Z} \ , && (9.14\text{b}) \\
& \|T^{\frac{1}{2}}(\bar{x}_N - \bar{x}_s)\|_2^2 && \le 1 - \bar{r}(\bar{x}_s, \bar{u}_s) \ , && (9.14\text{c}) \\
& G_x \bar{x}_i && \le \bar{f}_x + \bar{\epsilon}_s + \bar{\epsilon}_i \ , && (9.14\text{d}) \\
& G_u \bar{u}_i && \le \bar{f}_u \ , && (9.14\text{e}) \\
& (1 + \xi) G_x \bar{x}_s && \le \bar{f}_x + \bar{\epsilon}_s \ , && (9.14\text{f}) \\
& c\|T^{1/2}(\bar{x}_N - \bar{x}_s)\|_2 && \le \bar{f}_x + \bar{\epsilon}_s - G_x \bar{x}_s \ , && (9.14\text{g}) \\
& (9.3\text{c}), (9.3\text{g}), (9.3\text{i}), (9.3\text{j}) && && (9.14\text{h})
\end{aligned}$$

The robust soft constrained problem implicitly defines the set of feasible initial states $\mathcal{X}_N^{rs}$. For a given state $x \in \mathcal{X}_N^{rs}$, Problem $\mathbb{P}_N^{rs}$ yields the optimal control sequence $\mathbf{u}^{rs*}(x)$ and the optimal first tube center $\bar{x}_0^{rs*}(x)$. Note that the first tube center $\bar{x}_0$ is not necessarily equal to the current state measurement x but is an optimization variable. Ideally, $\mathcal{Z}$ would be taken as the minimal RPI set, an explicit representation can however not be computed except in special cases, see Section 8.5 for more details. In this work we employ a robust positively invariant ellipsoid for $\mathcal{Z}$, which is readily obtained by taking the smallest invariant level set of $V_f(x)$: $\mathcal{Z} \triangleq \{x \mid x^T P x \leq r\}$. For details on the computation please refer to Algorithm 6 in Section 8.5. The robust formulation does not change the problem structure and Problem $\mathbb{P}_N^{rs}(x)$ again results in a convex SOCP. The robust soft constrained control law is then given in a receding horizon fashion by

$$\kappa^{rs}(x) \triangleq \bar{u}_0^{rs*}(x) + K(x - \bar{x}_0^{rs*}(x)). \tag{9.15}$$

Input-to-state stability will in the following be shown for the robust invariant set $\mathcal{X}_{\mathcal{W}}^{rs} \subseteq \mathcal{X}_N^{rs}$ given by the robust positively invariant set for the controlled uncertain system $x(k+1) = Ax(k) + B\kappa^{rs}(x(k)) + w_1(k) + w_2(k)$:

$$Ax + B\kappa^{rs} + w_1 + w_2 \in \mathcal{X}_{\mathcal{W}}^{rs} \ \forall \ x \in \mathcal{X}_{\mathcal{W}}^{rs}, \ w_1 \in \mathcal{W}_1 \ w_2 \in \mathcal{W}_2. \tag{9.16}$$

Theorem 9.14 (ISS under $\boldsymbol{\kappa^{rs}(x)}$) *The closed loop system* $x(k+1) = Ax(k) + B\kappa^{rs}(x(k)) + w_1(k) + w_2(k)$ *is ISS w.r.t to* $w_1(k) \in \mathcal{W}_1$ *and* $w_2(k) \in \mathcal{W}_2$ *with region of attraction* $\mathcal{X}_{\mathcal{W}}^{rs}$*.*

Proof. Following similar arguments as in the proof of Lemma 9.10 and using optimality of $V_N^{rs*}(x)$ it can be shown that there exist $\mathcal{K}_\infty$-class functions $\underline{\alpha}(\cdot), \overline{\alpha}(\cdot)$ such that $V_N^{rs*}(x) \geq \underline{\alpha}(\|x\|) \ \forall x \in \mathcal{X}_N^{rs}$ and $V_N^{rs*}(x) \leq \overline{\alpha}(\|x\|) \ \forall x \in \mathcal{E}_{\bar{\mathcal{X}}_f^s}(0,0) \oplus \mathcal{Z}$. Let $\mathbf{u}^{\text{shift}}$ be the shifted sequence of $\bar{\mathbf{u}}^{rs*}(x)$ defined in (9.8) and $\mathbf{x}^{\text{shift}}$ the corresponding state sequence starting from $\bar{x}_1^{rs*}(x)$. By extending the arguments in the proof of Lemma 9.8 to the robust case, feasibility of the shifted solution can be shown with $\bar{\boldsymbol{\epsilon}}^{\text{shift}} = [\bar{\epsilon}_1^{rs*}, \dots, \bar{\epsilon}_{N-1}^{rs*}, 0, \bar{\epsilon}_s^{rs*}]$. Let $w_2 = 0$, i.e. $x^+ = Ax + B\kappa^{rs}(x) + w_1$. Using the same arguments as in the proof of Theorem 5.8, we obtain $V_f(x^+ - \bar{x}_1^{rs*}(x)) - V_f(x - \bar{x}_0^{rs*}(x)) \leq -\|x - \bar{x}_0^{rs*}(x)\|_Q^2 + \gamma_1(\|w_1\|)$. Using standard arguments and convexity of $\|\cdot\|_Q^2$, it can then be shown that there exists a $\mathcal{K}$-class function $\gamma_1(\cdot)$ such that

$$\begin{aligned}
&V_N^{rs*}(Ax + B\kappa^{rs}(x) + w_1) - V_N^{rs*}(x) \\
&\leq V_N^s(\bar{\mathbf{x}}^{\text{shift}}, \bar{\mathbf{u}}^{\text{shift}}, \bar{x}_s^{rs*}(x), \bar{u}_s^{rs*}(x), \bar{\boldsymbol{\epsilon}}^{\text{shift}}) + V_f(x^+ - \bar{x}_1^{rs*}(x)) \\
&\quad - V_N^s(\bar{\mathbf{x}}^{rs*}(x), \bar{\mathbf{u}}^{rs*}(x), \bar{x}_s^{rs*}(x), \bar{u}_s^{rs*}(x), \bar{\boldsymbol{\epsilon}}^{rs*}(x)) - V_f(x - \bar{x}_0^{rs*}(x)) \\
&\leq -\|\bar{x}_0^{rs*}(x) - \bar{x}_s^{rs*}(x)\|_Q^2 - \|x - \bar{x}_0^{rs*}(x)\|_Q^2 + \gamma_1(\|w_1\|) \\
&\leq -\frac{1}{2}\|x - \bar{x}_s^{rs*}(x)\|_Q^2 + \gamma_1(\|w_1\|) \ .
\end{aligned}$$

Note that Lemma 9.9 can be directly extended to the robustified problem formulation, since the term $V_f(x - \bar{x}_0)$ in the cost (9.14a) is irrelevant for the argument. $\bar{x}_s^{rs*}(x) = \bar{x}_0^{rs*}(x)$ can hence only be the optimal solution if $\bar{x}_0^{rs*}(x) = 0$. If $\bar{x}_s^{rs*}(x) = x$ for $x \neq 0$ was the optimal choice then also $\bar{x}_0^{rs*}(x) = x$, which is a contradiction and

$\|x - \bar{x}_s^{rs*}(x)\|$ can only be zero at the origin. Following a similar argument as in the proof of Theorem 9.10, it then follows that there exists a $\mathcal{K}_\infty$-class function $\beta(\bullet)$ such that

$$V_N^{rs*}(Ax + B\kappa^{rs}(x) + w_1) - V_N^{rs*}(x) \leq -\beta(\|x\|) + \gamma_1(\|w_1\|).$$

Continuity of the optimal value function V_N^{rs*} follows from the proof of Lemma 9.12, therefore there exists a $\mathcal{K}$-class function $\gamma_2(\bullet)$, such that $|V_N^{rs*}(y) - V_N^{rs*}(x)| \leq \gamma_2(\|y - x\|)$ and we obtain

$$\begin{aligned}
&V_N^{rs*}(Ax + B\kappa^{rs}(x) + w_1 + w_2) - V_N^{rs*}(x) \\
&\leq |V_N^{rs*}(Ax + B\kappa^{rs}(x) + w_1 + w_2) - V_N^{rs*}(Ax + B\kappa^{rs}(x) + w_1)| \\
&\quad + |V_N^{rs*}(Ax + B\kappa^{rs}(x) + w_1) - V_N^{rs*}(x)| \\
&\leq -\beta(\|x\|) + \gamma_1(\|w_1\|) + \gamma_2(\|w_2\|) \qquad \blacksquare
\end{aligned}$$

proving the result.

Theorem 9.14 proves ISS of the uncertain system in (9.13) controlled by $\kappa^{rs}(x)$ in (9.15). The results presented for the soft constrained MPC method can hence directly be extended to the combination of a robust and soft constrained approach.

9.8 Numerical Examples

In this section the proposed methods for soft constrained and robust soft constrained MPC are illustrated using numerical examples. All set computations were carried out using the YALMIP toolbox [Löf04] and the MPT toolbox [KGBM04].

9.8.1 Illustrative Example

Consider the following system:

$$x(k+1) = \begin{bmatrix} 1.05 & 1 \\ 0 & 1 \end{bmatrix} x(k) + \begin{bmatrix} 1 \\ 0.5 \end{bmatrix} u(k) + w(k) \ . \tag{9.17}$$

The system has eigenvalues at $s_1 = 1.05,\ s_2 = 1$ and is hence unstable. The prediction horizon was chosen to be $N = 5$, the constraints on the states and control inputs to $\|x\|_\infty \leq 5$ and $\|u\|_\infty \leq 1$, $Q = I$, $R = 1$ and $S = 100I$. The terminal cost function $V_f(x)$ is taken as the unconstrained infinite horizon optimal value function for the nominal system with $P = \left[\begin{smallmatrix} 1.9119 & 0.2499 \\ 0.2499 & 2.6510 \end{smallmatrix}\right]$ and $\kappa_f(x) = Kx$ is the corresponding optimal LQR controller. The exact penalty multipliers were chosen as $\rho_1 = \rho_2 = 100$ which was observed to provide optimality in $\mathcal{X}_N$ as defined in Theorem 9.7.

Soft Constrained MPC

First, the feasible set $\mathcal{X}_5^s$ and the enlarged terminal set $\mathcal{C}_f^s$ for the soft constrained approach $\mathbb{P}_N^s(x)$ are illustrated and compared with the feasible set $\mathcal{X}_5$ for the hard constraint problem $\mathbb{P}_N(x)$ in Figure 9.4, which demonstrates that the soft constrained approach significantly enlarges the feasible set and thereby the region of attraction for the nominal closed-loop system. In addition we plot a state set of interest, where asymptotic stability should be guaranteed, taken as $2\mathbb{X} \cap \mathcal{X}_\infty$. It can be seen that for a horizon of $N = 5$ the set is not included in the feasible set $\mathcal{X}_5^s$. If we however lengthen the horizon to $N = 8$, then the set of interest is included in $\mathcal{X}_8^s$.

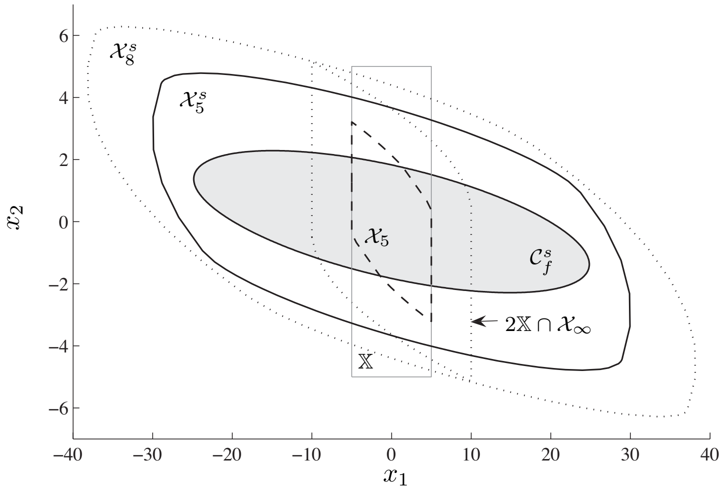
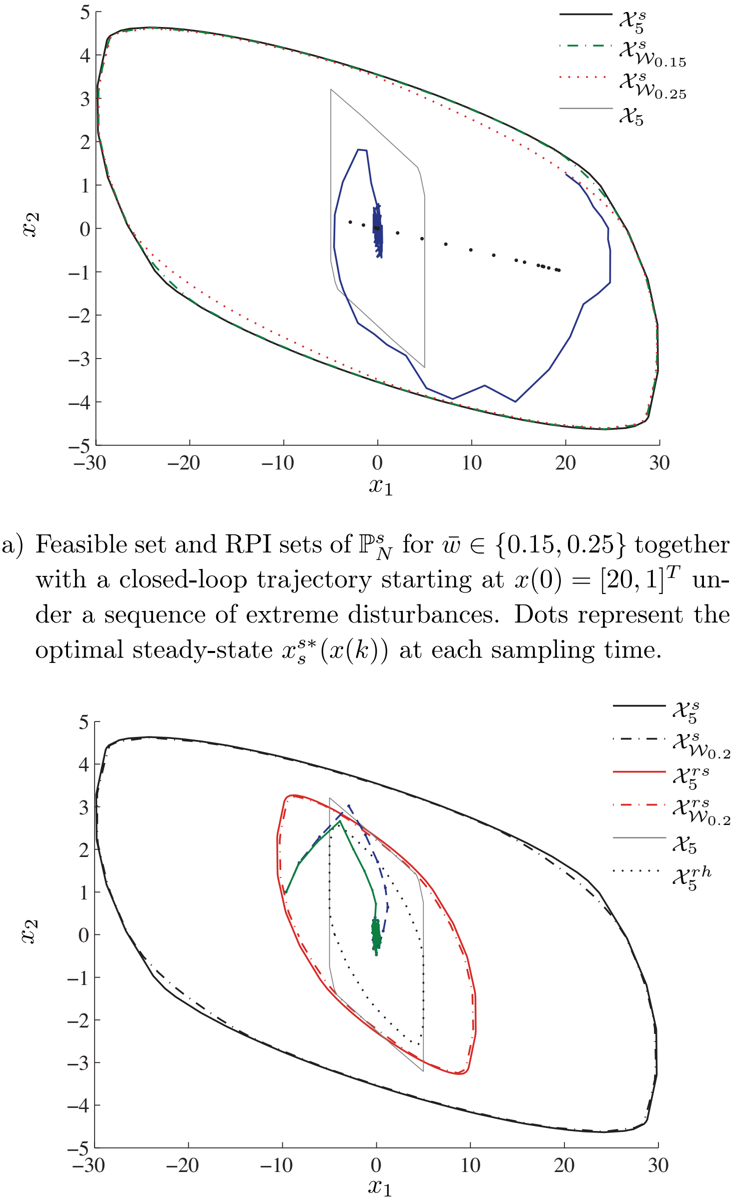


Figure 9.4: Feasible and terminal set for the soft constrained approach for $N = 5$ in comparison with the feasible set of the hard constraint problem. The state set of interest $2\mathbb{X} \cap \mathcal{X}_\infty$ can be covered by $\mathcal{X}_8^s$.

We now analyze the robust stability properties of the example system (9.17) under the soft constrained control law $\kappa^s(x)$ in (9.6). Figure 9.5 shows the size of the RPI sets $\mathcal{X}_{\mathcal{W}_{\bar{w}}}$ for two bounds $\mathcal{W}_{\bar{w}} \triangleq \{w \mid \|w\|_\infty \leq \bar{w}\}$, with $\bar{w} \in \{0.15, 0.25\}$. Note that for a robust tube-based approach the feasible set is always a subset of $\mathcal{X}_N$. This demonstrates the advantage of the soft constrained approach, where input-to-state stability in the presence of a comparably large disturbance $w \in \mathcal{W}_{0.25}$ can be guaranteed in the RPI set $\mathcal{X}_{\mathcal{W}_{0.25}} \supseteq \mathcal{X}_N$. In addition, a closed-loop trajectory starting at $x(0) = [20\ 1]^T$ under a sequence of extreme disturbances with $w(k) = \pm 0.25,\ k \geq 0$ is shown as well as the corresponding optimal steady-states at each sampling time, demonstrating that the closed-loop system is stable and does not leave the RPI set $\mathcal{X}_{\mathcal{W}_{0.25}}$.



(a) Feasible set and RPI sets of $\mathbb{P}_N^s$ for $\bar{w} \in \{0.15, 0.25\}$ together with a closed-loop trajectory starting at $x(0) = [20, 1]^T$ under a sequence of extreme disturbances. Dots represent the optimal steady-state $x_s^{s*}(x(k))$ at each sampling time.

(b) Feasible set and RPI sets of $\mathbb{P}_N^s$ and $\mathbb{P}_N^{rs}$ for $\bar{w}_1 = 0.1, \bar{w}_2 = 0.1$ together with closed-loop trajectories starting at $x(0) = [-9.6, 1]^T$, where the dashed line solves $\mathbb{P}_N^s$ and the solid line $\mathbb{P}_N^{rs}$. $\mathcal{X}_5^{rh}$ denotes the feasible set for the robust hard constrained problem.

Figure 9.5: Illustration of the soft constrained and the robust soft constrained MPC method

Robust Soft Constrained MPC

In the following the properties of the robust soft constrained MPC approach described in Section 9.7 are illustrated. Consider again system (9.17) with $w(k) = w_1(k)+w_2(k)$ that is now subject to two types of disturbances. Figure 9.5(b) shows the comparison of the feasible set $\mathcal{X}_5^{rs}$ and the RPI set $\mathcal{X}_{\mathcal{W}_{0.2}}^{rs}$ for $w_1 \in \mathcal{W}_{\bar{w}_1}, w_2 \in \mathcal{W}_{\bar{w}_2}$ with $\bar{w}_1 = 0.1, \bar{w}_2 = 0.1$ in comparison with the feasible set $\mathcal{X}_N^s$ and the RPI set $\mathcal{X}_{\mathcal{W}_{0.2}}^s$ of the pure soft constrained approach. The feasible set of the hard constrained robust MPC problem, i.e. Problem $\mathbb{P}_N^{rs}(x)$ with $x_s = u_s = 0, \epsilon_s = 0, \epsilon_i = 0$, $w = w_1 + w_2 \in \mathcal{W}_{0.2}$ is denoted by $\mathcal{X}_5^{rh}$. Due to the tightening of the input constraints, the robust soft constrained approach has a significantly smaller feasible set when compared to the pure soft constrained method. However, in comparison with the hard constrained robust MPC method, the feasible set for the combined approach is significantly larger while still guaranteeing ISS with respect to w_1 in $\mathcal{X}_N^{rs}$, which is almost as large as the nominal feasible set $\mathcal{X}_5$. The RPI set $\mathcal{X}_{\mathcal{W}_{0.2}}^{rs}$ is only slightly smaller than $\mathcal{X}_5^{rs}$ and input-to-state stability with respect to the combined disturbance $w = w_1 + w_2 \in \mathcal{W}_{0.2}$ is still provided in a comparably large set. Closed-loop trajectories starting from $x(0) = [-9.6 \; 1]^T$ are shown for both approaches under a sequence of extreme disturbances $w_1(k) = \pm 0.1$ and a disturbance $w_2(k) \in \mathcal{W}_{0.1}$, $k \geq 0$, of varying size that additionally affects the system at every third sampling time. Figure 9.5(b) demonstrates that both approaches provide input-to-state stability of the closed-loop system, however the robust soft constrained approach provides a better performance, since it is designed for the disturbance w_1 that constantly affects the system.

9.8.2 Large-Scale Example

We now apply the soft constrained MPC approach to a large scale example and estimate the computational effort required to solve the corresponding SOCP. Consider the problem of regulating a system of 12 oscillating masses which are interconnected by spring-damper systems and connected to walls on the side, as shown in Fig. 9.6. The six actuators exert tension between different masses. The masses are 1, the spring constants are 1, the damping constants are 0.1 and $F = 1.05x_1$. The state and input constraints are $\|u\|_\infty \leq 1$, $\|x\|_\infty \leq 4$, the horizon is chosen as $N = 5$ and the weight

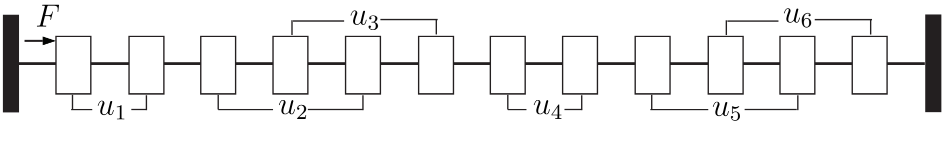


Figure 9.6: System of oscillating masses.

matrices Q and R are taken as identity matrices. The MPC problem has 24 states and 6 inputs, resulting in an SOCP with 300 optimization variables. The invariant ellipsoidal target set $\mathcal{E}_{\mathcal{X}_f^s}(x_s, u_s)$ was computed by solving an LMI in 11s. The soft-constrained MPC problem is solved in only 1.8s using SeDuMi [SED], averaged over 100 initial states, where the simulations were executed on a 2.8GHz AMD Opteron running Linux. This demonstrates that the soft constrained MPC problem $\mathbb{P}_N^s(x)$ can be solved with reasonably low computation times even for large system dimensions.

9.9 Conclusions

In this chapter a new soft constrained MPC method based on a finite horizon MPC scheme was introduced that provides closed-loop stability even for unstable systems. The proposed control law preserves optimality and constraint satisfaction whenever the state constraints can be enforced. Asymptotic stability of the nominal system under the soft constrained control law was shown as well as input-to-state stability in the presence of additive disturbances and the robust stability properties were analyzed. The results on input-to-state stability were extended to the combination of a robust and soft constrained approach.

10 Conclusions and Outlook

The main focus of this thesis was the development of real-time and soft-constrained MPC methods that guarantee the essential properties of closed-loop feasibility and stability. Providing a control law that meets the computational requirements in particular in terms of available online computation time is crucial for the application of MPC to high-speed systems. In addition, practical implementations of MPC often consider the relaxation of state and output constraints. The development of methods providing theoretical properties for these approaches are therefore highly relevant. A detailed summary of the presented results can be found at the end of each chapter. In the following, we provide a brief overview of the work that was presented in this thesis, as well as an outlook to possible directions for future research on these topics.

In Chapter 7, a real-time MPC method based on a combination of online and explicit MPC was developed, which offers the possibility to trade off solution properties in order to satisfy constraints in terms of online computation time, storage and performance. By means of a preprocessing analysis hard real-time, stability and performance guarantees for the proposed controller are provided. As the best solution method is dependent on the particular system as well as the given hardware and performance restrictions, it was shown how the offline analysis can be utilized to identify the best approach for a given set of requirements. The presented results show that for many systems there exists a point where the combination of an online and explicit approximation method outperforms a pure explicit or online approximation.

The use of an explicit approximation, however, still limits the applicability to small or medium size problems. A real-time approach based solely on online optimization was therefore developed in Chapter 8, which can be applied to all problem dimensions. The limitations of fast online MPC methods in the literature were addressed and feasibility, stability and convergence for all time constraints was provided by using a robust MPC problem setup with a Lyapunov constraint. It was shown that by exploiting the new structure of the resulting optimization problem, a tailored custom solver achieves computation times in the range of milliseconds that are equal, or faster than those reported for methods without guarantees.

The last part of the thesis was concerned with soft constrained MPC. A new soft-constrained MPC method was presented in Chapter 9, that provides closed-loop stability even for unstable systems. Asymptotic stability of the nominal system with an enlarged region of attraction under the soft constrained control law was shown as well as input-to-state stability in the presence of additive disturbances. The robust stability properties were analyzed and the results demonstrate that input-to-state stability can be provided in an enlarged region of attraction compared to a robust MPC approach considering the same disturbance size. A combination of this approach with a robust MPC scheme allows for flexibility in the disturbance handling and the stability results extend to the combined approach.

The results presented in this thesis successfully show that theoretical properties can be maintained in a practical MPC implementation and a real-time or soft constrained MPC method with stability guarantees can be provided. The following topics would be interesting for future developments in this area.

One of the main limitations in real-time MPC methods is the initialization of the MPC problem with a feasible solution. Feasibility is required in order to provide closed-loop stability using standard results. Since it cannot be guaranteed that feasibility is recovered in a real-time setting with a given online computation time that generally only allows for suboptimal solutions, the optimization has to be started from an initial feasible point. In a nominal approach a feasible initialization is readily available using the solution computed at the previous sampling time together with a local stable control law. This initialization may, however, no longer be feasible in the presence of disturbances.

In the real-time methods presented in Chapter 7 and 8 this issue was resolved by using a feasible explicit approximation to warm-start the optimization or by applying a robust MPC scheme that recovers recursive feasibility and thereby provides feasibility of the shifted previous sequence. Explicit approximations are, however, limited to small problem dimensions and robust MPC methods are designed for a pre-specified bound on the disturbance size, in which case the guarantees are only valid if the disturbance never exceeds the expected bound. One way to circumvent the necessity of a feasible initialization is the relaxation of state constraints by means of a soft constrained approach with stability guarantees. This motivates the extension of the presented soft-constrained method in Chapter 9 relying on optimal solutions to a suboptimal and real-time approach. The challenge would be to extend the stability proof for a real-time control law and different stability results would need to be investigated.

The main disadvantage of the presented soft constrained MPC approach is the need for second-order cone constraints. It was shown that SOCPs can, however, still be solved efficiently by means of interior-point methods (see e.g. [ART03]), in particular since the resulting SOC constraints are of small dimension. A fast solver exploiting the structure of an MPC problem with SOC constraints would prove the practicality of the approach and would also be relevant for SOCPs resulting from different problem setups such as robust programming [LVBL98].

Interesting aspects on the topic of soft constrained MPC include the incorporation of the possibility to prioritize the relaxation of constraints, building on the results in [Ker00], which would improve the performance of the soft constrained scheme. The efficient computation of an upper bound on the exact penalty multiplier represents a relevant topic beyond the application to soft constrained MPC. Exact penalty functions are often applied in control but a corresponding multiplier can only be obtained by simulation if the parametric solution is intractable. Any improvement on this topic would therefore have an important impact.

Since the results in this thesis focus on a linear system model, a more general direction of future work would be the development of real-time methods for nonlinear systems. In the nonlinear case, the MPC problem results in a nonlinear optimization problem that has to be solved in real-time. While several methods for fast solution of these problems have been proposed, see for example the overview in [DFH09], the challenge would be to extend the main theme of this thesis and provide theoretical guarantees on the resulting real-time control laws.

Bibliography

[AB09] A. Alessio and A. Bemporad, *A Survey on Explicit Model Predictive Control*, Lecture Notes in Control and Information Sciences, vol. 384, pp. 345–369, Springer, 2009.

[AH08] D. Axehill and A. Hansson, *A dual gradient projection quadratic programming algorithm tailored for model predictive control*, Proc. 47th IEEE Conf. on Decision and Control, Dec. 2008, pp. 3057–3064.

[ALA$^+$07] I. Alvarado, D. Limon, T. Alamo, M. Fiacchini, and E.F. Camacho, *Robust tube based MPC for tracking of piece-wise constant references*, Proc. 46th IEEE Conf. on Decision and Control, Dec. 2007, pp. 1820–1825.

[ALBH07] A. Alessio, M. Lazar, A. Bemporad, and W. P. M. H. Heemels, *Squaring the circle: An algorithm for generating polyhedral invariant sets from ellipsoidal ones*, Automatica **43** (2007), 2096–2103.

[ART03] E. D. Andersen, C. Roos, and T. Terlaky, *On implementing a primal-dual interior–point method for conic quadratic optimization*, Mathematical Programming **95** (2003), 249–277.

[ATW94] 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** (1994), 221–248.

[Bao02] M. Baotić, *An efficient algorithm for multi-parametric quadratic programming*, Tech. Report AUT02-05, Automatic Control Laboratory, ETH, Zurich, Switzerland, 2002.

[BB06] R. A. Bartlett and L. T. Biegler, *QPSchur: a dual, active set, Schur complement method for large-scale and structured convex quadratic programming algorithm*, Optimization and Engineering **7** (2006), 5–32.

[BBBG02] R. A. Bartlett, L. T. Biegler, J. Backstrom, and V. Gopal, *Quadratic programming algorithms for large-scale model predictive control*, Journal of Process Control **12** (2002), 775–795.

[BBBM08] M. Baotić, F. Borrelli, A. Bemporad, and M. Morari, *Efficient On-Line*

Computation of Constrained Optimal Control, SIAM Journal on Control and Optimization **47** (2008), no. 5, 2470–2489.

[BBM02] A. Bemporad, F. Borrelli, and M. Morari, *Model Predictive Control Based on Linear Programming - The Explicit Solution*, IEEE Transactions on Automatic Control **47** (2002), no. 12, 1974–1985.

[BBM03] F. Borrelli, A. Bemporad, and M. Morari, *A Geometric Algorithm for Multi-Parametric Linear Programming*, Journal of Optimization Theory and Applications **118** (2003), no. 3, 515–540.

[BCM97] A. Bemporad, A. Casavola, and E. Mosca, *Nonlinear Control of Constrained Linear Systems via Predictive Reference Managment*, IEEE Transactions on Automatic Control **42** (1997), 340–349.

[BDH96] C. B. Barber, D. P. Dobkin, and H. Huhdanpaa, *The Quickhull Algorithm for Convex Hulls*, ACM Transactions on Mathematical Software **22** (1996), no. 4, 469–483.

[BEFB94] S. Boyd, L. El Ghaoui, E. Feron, and V. Balakrishnan, *Linear Matrix Inequalities in System and Control Theory*, Studies in Applied Mathematics, vol. 15, SIAM, Philadelphia, PA, June 1994.

[BF03] A. Bemporad and C. Filippi, *Suboptimal Explicit Receding Horizon Control via Approximate Multiparametric Quadratic Programming*, Journal of Optimization Theory and Applications **117** (2003), 9–38.

[BF06] A. Bemporad and C. Filippi, *An Algorithm for Approximate Multiparametric Convex Programming*, Computational Optimization and Applications **35** (2006), no. 1, 87–108.

[BGK$^+$82] B. Bank, J. Guddat, D. Klatte, B. Kummer, and K. Tammer, *Non-linear parametric optimization*, Akademie-Verlag, Berlin, 1982.

[BI75] E. M. Bronshteyn and L. D. Ivanov, *The approximation of convex sets by polyhedra*, Siberian Mathematical Journal **16** (1975), no. 5, 852–853.

[Bla77] R. G. Bland, *New finite pivoting rules for the simplex method*, Mathematics of Operations Research **2** (1977), 103–107.

[Bla99] F. Blanchini, *Set invariance in control*, Automatica **35** (1999), 1747–1767.

[BM99] A. Bemporad and M. Morari, *Robust Model Predictive Control: A Survey*, Robustness in Identification and Control **245** (1999), 207–226.

[BMDP02] A. Bemporad, M. Morari, V. Dua, and E. N. Pistikopoulos, *The Explicit Linear Quadratic Regulator for Constrained Systems*, Automatica **38** (2002), 3–20.

[Bor03] F. Borrelli, *Constrained Optimal Control of Linear and Hybrid Systems*, Lecture Notes in Control and Information Science, vol. 290, Springer Verlag, 2003.

[BR85] M. J. Best and K. Ritter, *Linear Programming: Active Set Analysis and Computer Programs*, Prentice-Hall, Inc., 1985.

[BTN01] A. Ben-Tal and A. Nemirovski, *Lectures on modern convex optimization: Analysis, algorithms, and engineering applications*, Society for Industrial Mathematics, 2001.

[BV04] S. Boyd and L. Vandenberghe, *Convex Optimization*, Cambridge University Press, 2004.

[BWB00] R. A. Bartlett, A. Wachter, and L. T. Biegler, *Active set vs. interior point strategies for model predictive control*, Proc. American Control Conference, 2000.

[CB04] E. F. Camacho and C. Bordons, *Model Predictive Control*, 2nd ed., Advanced Textbooks in Control and Signal Processing, Springer Verlag, London, 2004.

[CDD] *A C++ implementation of the double description method*, `http://www.ifor.math.ethz.ch/~fukuda/cdd_home/cdd.html`.

[Chr07] F. J. Christophersen, *Optimal contol of constrained piecewise affine systems*, Lecture Notes in Control and Information Sciences, vol. 359, Springer Verlag, 2007.

[CLK08] M. Cannon, W. Liao, and B. Kouvaritakis, *Efficient mpc optimization using pontryagin's minimum principle*, Int. J. of Robust and Nonlinear Control **18** (2008), 831–844.

[CLP] *Coin-or linear programming*, `https://projects.coin-or.org/Clp`.

[CM96] D. Chmielewski and V. Manousiouthakis, *On Constrained Infinite-Time Linear Quadratic Optimal Control*, Proc. 35th Conf. on Decision and Control (Kobe, Japan), Dec. 1996.

[CPL] *ILOG CPLEX Optimizer*, `http://www-01.ibm.com/software/integration/optimization/cplex-optimizer/`.

[CRZ01] L. Chisci, J. A. Rossiter, and G. Zappa., *Systems with Persistent Disturbances: Predictive Control with Restricted Constraints*, Automatica **37** (2001), 1019–1028.

[CZ03] L. Chisci and G. Zappa, *Dual mode predictive tracking of piecewise constant references for constrained linear systems*, International Journal of Control

76 (2003), 61–72.

[CZJM07] F. J. Christophersen, M. N. Zeilinger, C. N. Jones, and M. Morari, *Controller complexity reduction for piecewise affine systems through safe region elimination*, Proc. 46th IEEE Conf. on Decision and Control, 2007, pp. 4773–4778.

[Dan49] G. B. Dantzig, *Programming in a linear structure*, Econometrica **17** (1949), 73–74.

[Dan63] ______, *Linear programming and extensions*, Princeton University Press, 1963.

[DB89] J. C. Dunn and D. P. Bertsekas, *Efficient dynamic programming implementations of newton's method for unconstrained optimal control problems*, Journal of Optimization Theory and Applications **63** (1989), 23–38.

[DFH09] M. Diehl, H. J. Ferreau, and N. Haverbeke, *Efficient Numerical Methods for Nonlinear MPC and Moving Horizon Estimation*, Nonlinear Model Predictive Control: Towards New Challenging Applications (L. Magni, D. M. Raimondo, and F. Allgöwer, eds.), Springer-Verlag, 2009, pp. 391–417.

[dOB94] N. M. C de Oliveira and L. T. Biegler, *Constraint Handling and Stability Properties of Model-Predictive Control*, AIChE Journal **40** (1994), 1138–1155.

[FBD08] H. J. Ferreau, H. G. Bock, and M. Diehl, *An online active set strategy to overcome the limitations of explicit* MPC, International Journal of Robust and Nonlinear Control **18** (2008), 816–830.

[FBD09] ______, *qpOASES – Online Active Set Strategy* , `http://www.kuleuven.be/optec/software/qpOASES`, 2009.

[FLA+08] A. Ferramosca, D. Limon, I. Alvarado, T. Alamo, and E. F. Camacho, *On the characterization of the conditions for local optimality for MPC for tracking*, Tech. report, Universidad de Sevilla, 2008.

[FLA+09] A. Ferramosca, D. Limon, I. Alvarado, T. Alamo, and E. F. Camacho, *MPC for tracking with optimal closed-loop performance*, Automatica **45** (2009), 1975–1978.

[Fle71] R. Fletcher, *A general quadratic programming algorithm*, IMA Journal of Applied Mathematics **7** (1971), no. 1, 76–91.

[Fle87] ______, *Practical Methods of Optimization*, second ed., John Wiley and Sons, New York, 1987.

[FT72] J. J. H. Forrest and J. A. Tomlin, *Updated triangular factors of the basis to maintain sparsity in the product form simplex method*, Mathematical

Programming **2** (1972), 263–278.

[Gal95] T. Gal, *Postoptimal analysis, parametric programming and related topics*, de Gruyter, Berlin, 1995.

[GB98] V. Gopal and L. T. Biegler, *Large scale inequality constrained optimization and control*, IEEE Control Systems Magazine **18** (1998), 59 – 68.

[GJ84] T. Glad and H. Johnson, *A Method for State and Control Constrained Linear-Quadratic Control Problems*, Proc. 9th IFAC World Congress (Budapest, Hungary), 1984.

[GK99] E. G. Gilbert and I. Kolmanovsky, *Fast reference governors for systems with state and control constraints and disturbance inputs*, Int. Journal of Robust and Nonlinear Control **9** (1999), 1117–1142.

[GKBM04] P. Grieder, M. Kvasnica, M. Baotić, and M. Morari, *Low complexity control of piecewise affine systems with stability guarantee*, Proc. American Control Conference (Boston, MA), 2004.

[GKR08] Paul Goulart, Eric Kerrigan, and Daniel Ralph, *Efficient robust optimization for robust control with constraints*, Mathematical Programming **114** (2008), no. 1, 115–147.

[GM03] P. Grieder and M. Morari, *Complexity reduction of receding horizon control*, Proc. 42th IEEE Conf. on Decision and Control (Maui, Hawaii, USA), Dec. 2003, pp. 3179 – 3184.

[GMSW89] P. E. Gill, W. Murray, M. A. Saunders, and M. H. Wright, *A practical anti-cycling procedure for linearly constrained optimization*, Mathematical Programming **45** (1989), no. 1-3, 437–474.

[GMTT04] G. Grimm, M. J. Messina, S. E. Tuna, and A. R. Teel, *Examples when nonlinear model predictive control is nonrobust*, Automatica **40** (2004), 1729–1738.

[GMW82] P. E. Gill, W. Murray, and M. H. Wright, *Practical Optimization*, Academic Press, 1982.

[GN72] T. Gal and J. Nedoma, *Multiparametric Linear Programming*, Management Science **18** (1972), no. 7, 406–422.

[GPM89] C. E. Garcia, D. M. Prett, and M. Morari, *Model Predictive Control: Theory and Practice – A Survey*, Automatica **22** (1989), 335–348.

[Grü03] B. Grünbaum, *Convex polytopes*, 2nd ed., Springer, 2003.

[Grü61] B. Grünbaum, *Measures of symmetry for convex sets*, Proc. 7th Symp. in Pure Mathematics of the American Mathematical Society, Symposium on

Convexity (1961), 233–270.

[GSdD06] G. C. Goodwin, M. M. Seron, and J. A. de Doná, *Constrained Control and Estimation, An Optimisation Approach*, Springer, 2006.

[GT91] E.G. Gilbert and K. T. Tan, *Linear systems with State and Control Constraints: The Theory and Application of Maximal Output Admissible Sets*, IEEE Transactions on Automatic Control **36** (1991), 1008–1020.

[GTM08] T. Geyer, F. D. Torrisi, and M. Morari, *Optimal complexity reduction of polyhedral piecewise affine systems*, Automatica **44** (2008), 1728–1740.

[Gup09] V. Gupta, *On an anytime algorithm for control*, Proc. IEEE Conf. on Decision and Control (Shanghai, China), Dec. 2009, pp. 6218 – 6223.

[Hag79] William W. Hager, *Lipschitz continuity for constrained processes*, SIAM Journal on Control and Optimization **17** (1979), 321–338.

[Han00] A. Hansson, *A primal-dual interior-point method for robust optimal control of linear discrete-time systems*, IEEE Transactions on Automatic Control **45** (2000), no. 9, 1639–1655.

[HL02] B. Hu and A. Linnemann, *Toward infinite-horizon optimality in nonlinear model predictive control*, IEEE Transactions on Automatic Control **47** (2002), no. 4, 679–682.

[HNW08] L. Hei, J. Nocedal, and R. A. Waltz, *A Numerical Study of Active-Set and Interior-Point Methods for Bound Constrained Optimization*, Modeling, Simulation and Optimization of Complex Processes, 2008, pp. 273–292.

[JBM07] C. N. Jones, M. Barić, and M. Morari, *Multiparametric Linear Programming with Applications to Control*, European Journal of Control **13** (2007), no. 2-3, 152–170.

[JG03] T. A. Johansen and A. Grancharova, *Approximate explicit constrained linear model predictive control via orthogonal search tree*, IEEE Transactions on Automatic Control **48** (2003), 810–815.

[JGR06] C. N. Jones, P. Grieder, and S. V. Raković, *A logarithmic-time solution to the point location problem for parametric linear programming*, Automatica **42** (2006), 2215–2218.

[JKM07] C. N. Jones, E. C. Kerrigan, and E. C. Maciejowski, *Lexicographic perturbation for multiparametric linear programming with applications to control.*, Automatica **43** (2007), 1808–1816.

[JM06] C. N. Jones and M. Morari, *Multiparametric linear complementarity problems*, Proc. 45th IEEE Conf. on Decision and Control (San Diego, CA), 2006, pp. 5687–5692.

[JM09] ______, *Approximate Explicit MPC using Bilevel Optimization*, Proc. European Control Conference (Budapest, Hungary), August 2009.

[JM10] ______, *Polytopic Approximation of Explicit Model Predictive Controllers*, IEEE Transactions on Automatic Control **55** (2010), 2542 – 2553.

[Joh04] T. A. Johansen, *Approximate explicit receding horizon control of constrained nonlinear systems*, Automatica **40** (2004), no. 2, 293–300.

[JW01] Z.-P. Jiang and Y. Wang, *Input-to-state stability for discrete-time systems*, Automatica **37** (2001), 857–869.

[Kar84] N. Karmarkar, *A new polynomial-time algorithm for linear programming*, Combinatorics **4** (1984), 373–395.

[KBM96] M. Kothare, V. Balakrishnan, and M. Morari, *Robust constrained model predictive control using linear matrix inequalities*, Automatica **32** (1996), no. 10, 1361–1379.

[KC01] B. Kouvaritakis and M. Cannon, *Non-linear Predictive Control: Theory and Practice*, The Institution of Engineering and Technology, London, UK, 2001.

[KCR02] B. Kouvaritakis, M. Cannon, and J. A. Rossiter, *Who needs QP for linear MPC anyway?*, Automatica **38** (2002), 879–884.

[Ker00] E. Kerrigan, *Robust Constraint Satisfaction: Invariant Sets and Predictive Control*, Ph.D. thesis, University of Cambridge, 2000.

[KG98] I. Kolmanovsky and E. C. Gilbert, *Theory and computation of disturbance invariant sets for discrete-time linear systems*, Mathematical Problems in Engineering **4** (1998), 317–367.

[KGBM04] M. Kvasnica, P. Grieder, M. Baotic, and M. Morari, *Multi-Parametric Toolbox (MPT)*, pp. 448–462, March 2004.

[Kha02] H. Khalil, *Nonlinear systems*, 3rd ed., Prentice Hall, 2002.

[KM82] V. Klee and G. J. Minty, *How good is the simplex algorithm?*, Inequalities, ch. III, pp. 159–175, Academic Press, New York, 1982.

[KM00] E. C Kerrigan and J. M. Maciejowski, *Soft Constraints and Exact Penalty Functions in Model Predictive Control*, Proc. UKACC International Conference (Control 2000) (Cambridge, UK), Sept. 2000.

[KRS00] B. Kouvaritakis, J. A. Rossiter, and J. Schuurmans, *Efficient robust predictive control*, IEEE Transactions on Automatic Control **45** (2000), no. 8, 1545–1549.

[Kur61] K. Kuratowski, *Introduction to set theory and topology*, Pergamon Press, 1961.

[LÖ1] J. Löfberg, *Linear Model Predictive Control: Stability and Robustness*, Ph.D. thesis, Linköping University, 2001.

[LAAC08] D. Limon, I. Alvarado, T. Alamo, and E.F. Camacho, *MPC for tracking piecewise constant references for constrained linear systems*, Automatica **44** (2008), 2382–2387.

[LAC02] D. Limon, T. Alamo, and E. F. Camacho, *Input-to-state stable MPC for constrained discrete-time nonlinear systems with bounded additive uncertainties*, Proc. 41st IEEE Conf. on Decision and Control, vol. 4, 2002, pp. 4619– 4624.

[LAC03] D. Limon, T. Alamo, and E. F. Camacho, *Stable constrained MPC without terminal constraint*, Proc. American Control Conf., vol. 6, 2003, pp. 4893–4898.

[LAR$^+$09] D. Limon, T. Alamo, D. M. Raimondo, D. Muñez de la Peña, J. M. Bravo, A. Ferramosca, and E. F. Camacho, *Input-to-State Stability: A Unifying Framework for Robust Model Predictive Control*, Lecture Notes in Control and Information Sciences **384** (2009), 1–26.

[LaS76] J. P. LaSalle, *The stability of dynamical systems*, Regional Conference Series in Applied Mathematics No. 25 Philadelphia, SIAM, 1976.

[Las93] J. B. Lasserre, *Reachable, controllable sets and stabilizing control of constrained linear systems*, Automatica **29** (1993), 531–536.

[LH09] M. Lazar and W. P. H. M. Heemels, *Predictive control of hybrid systems: Input-to-state stability results for sub-optimal solutions*, Automatica **45** (2009), 180–185.

[LKC10] S. Li, B. Kouvaritakis, and M. Cannon, *Improvements in the efficiency of linear MPC*, Automatica **46** (2010), no. 1, 226–229.

[LMF96] A. Lim, J. Moore, and L. Faybusovich, *Linearly constrained LQ and LQG Optimal Control*, Proc. 13th IFAC World Congress (San Francisco, CA), 1996.

[LR06] B. Lincoln and A. Rantzer, *Relaxing dynamic programming*, IEEE Transactions on Automatic Control **51** (2006), no. 8, 1249–1260.

[LRH$^+$08] M. Lazar, B. J. P. Roset, W. P. M. H. Heemels, H. Nijmeijer, and P. P. J. van den Bosch, *Input-to-state stabilizing sub-optimal nonlinear MPC algorithms with an application to DC-DC converters*, International Journal of Robust and Nonlinear Control **18** (2008), 890–904.

[Lue84] D. G. Luenberger, *Linear and Nonlinear Programming*, Addison-Wesley, 1984.

[LVBL98] M. S. Lobo, L. Vandenberghe, S. Boyd, and H. Lebret, *Applications of second-order cone programming*, Linear Algebra Appl. **284** (1998), 193–228.

[Löf04] J. Löfberg, *YALMIP : A Toolbox for Modeling and Optimization in MATLAB*, Proceedings of the CACSD Conference (Taipei, Taiwan), 2004.

[Mac00] J. Maciejowski, *Predictive Control with Constraints*, Prentice Hall, 2000.

[May01] D. Q. Mayne, *Control of Constrained Dynamic Systems*, European Journal of Control **7** (2001), 87–99.

[MBM09] U. Mäder, F. Borrelli, and M. Morari, *Linear offset-free Model Predictive Control*, Automatica **45** (2009), no. 10, 2214 – 2222.

[MD08] R. Milman and E. J. Davison, *A Fast MPC Algorithm Using Nonfeasible Active Set Methods*, Journal of Optimization Theory and Applications **139** (2008), 591–616.

[Meh92] S. Mehrotra, *On the Implementation of a Primal-Dual Interior Point Method* , SIAM Journal on Optimization **2** (1992), 575–601.

[ML99] M. Morari and J. H. Lee, *Model Predictive Control: past present and future*, Computers and Chemical Enineering **23** (1999), no. 4-5, 667–682.

[MRRS00] D. Q. Mayne, J. B. Rawlings, C. V. Rao, and P. O. M. Scokaert, *Constrained model predictive control: Stability and optimality*, Automatica **36(6)** (2000), 789–814.

[MRS06] L. Magni, D. M. Raimondo, and R. Scattolini, *Regional Input-to-State Stability for Nonlinear Model Predictive Control*, IEEE Transactions on Automatic Control **51** (2006), no. 9, 1548–1553.

[MS07] L. Magni and R. Scattolini, *Robustness and Robust Design of MPC for Nonlinear Discrete-Time Systems*, Assessment and Future Directions of Nonlinear Model Predictive Control, Springer, 2007.

[MSR05] D. Q. Mayne, M. M. Seron, and S. V. Rakovic, *Robust model predictive control of constrained linear systems with bounded disturbances*, Automatica **41** (2005), 219–234.

[Mur83] K. G. Murty, *Linear Programming*, Wiley, 1983.

[NAG] *Numerical Algorithms Group*, `http://www.nag.co.uk/`.

[Nes83] Yu. E. Nesterov, *A method for solving a convex programming problem with convergence rate* $\mathcal{O}(1/k^2)$, Soviet. Math. Dokl. **27** (1983), 372–376.

[Nes03] ———, *Introductory lectures on convex optimization: A basic course*, Springer, 2003.

[NFJM10] A. Nölle-Fuchs, C. N. Jones, and M. Morari, *Optimal decision trees for point location in polytopic data sets - application to explicit MPC*, Proc. American Control Conference (Baltimore, Maryland, USA), June 2010.

[NN94] Yu. E. Nesterov and A. Nemirovsky, *Interior-point polynomial methods in convex programming*, Studies in Applied Mathematics, vol. 13, SIAM, Philadelphia, 1994.

[NW06] J. Nocedal and S. J. Wright, *Numerical Optimization*, Springer Series in Operations Research, Springer New York, 2006.

[Pri07] J. A. Primbs, *A Soft Constraint Approach to Stochastic Receding Horizon Control*, Proc. 48th Conf. on Decision and Control, 2007.

[PRW07] G. Pannocchia, J. B.. Rawlings, and S. J. Wright, *Fast, large-scale model predictive control by partial enumeration*, Automatica **43** (2007), 852–860.

[PSS10] P. Patrinos, P. Sopasakis, and H. Sarimveis, *A global piecewise smooth newton method for fast large-scale model predictive control*, Tech. Report TR2010-02, National Technical University of Athens, 2010.

[QPC] *QPC – Quadratic Programming in C*, `http://sigpromu.org/quadprog/`.

[Raw00] J. B. Rawlings, *Tutorial overview of model predictive control*, IEEE Control Systems Magazine **20** (2000), no. 3, 38–52.

[RJM09] S. Richter, C. N. Jones, and M. Morari, *Real-Time Input-Constrained MPC Using Fast Gradient Methods*, Proc. 47th IEEE Conf. on Decision and Control, Dec. 2009, pp. 7387–7393.

[RKKM05] S. V. Rakovic, E. C. Kerrigan, K. I. Kouramas, and D. Q. Mayne, *Invariant Approximations of the Minimal Robust Positively Invariant Set*, IEEE Transactions on Automatic Control **50** (2005), 406–410.

[RM93] J.B. Rawlings and K.R. Muske, *The stability of Constrained Receding Horizon Control*, IEEE Transactions on Automatic Control **38** (1993), 1512–1516.

[RM09] J. B. Rawlings and D. Q. Mayne, *Model predictive control: Theory and design*, Nob Hill Publishing, 2009.

[RMKK05] S. V. Rakovic, D. Q. Mayne, E. C. Kerrigan, and K. I. Kouramas, *Optimized robust control invariant sets for constrained linear discrete-time systems*, Proc. 16th IFAC world congress, 2005.

[Roc70] R. T. Rockafellar, *Convex analysis*, Princeton University Press, 1970.

[Ros03] J. A. Rossiter, *Model-Based Predictive Control: A Practical Approach*, CRC Press Inc., 2003.

[RTMA06] S. V. Rakovic, A. R. Teel, D. Q. Mayne, and A. Astolfi, *Simple robust control invariant tubes for some classes of nonlinear discrete time systems*, Proc. 45th IEEE Conference on Decision and Control, 13-15 Dec. 2006, pp. 6397–6402.

[RW98] R. T. Rockafellar and R. J-B. Wets, *Variational analysis*, Springer, 1998.

[RWR98] C. V. Rao, S. J. Wright, and J. B. Rawling, *Application of interior–point methods to model predictive control*, Journal of Optimization Theory and Applications **99** (1998), 723–757.

[Sch87] M. Schechter, *Polyhedral functions and multiparametric linear programming*, Journal of Optimization Theory and Applications **53** (1987), 269Ű280.

[SED] *SeDuMi: a Matlab toolbox for optimization over symmetric cones.*, `http://sedumi.ie.lehigh.edu/`.

[SJLM09] S. Summers, C. N. Jones, J. Lygeros, and M. Morari, *A Multiscale Approximation Scheme for Explicit Model Predictive Control with Stability, Feasibility, and Performance Guarantees*, Proc. 48th IEEE Conference on Decision and Control (Shanghai, China), Dec. 2009, pp. 6327 – 6332.

[SMR99] P. O. M. Scokaert, D. Q. Mayne, and J. B. Rawlings, *Suboptimal model predictive control (feasibility implies stability)*, IEEE Transactions on Automatic Control **44** (1999), no. 3, 648–654.

[Son89] E. D. Sontag, *Smooth stabilization implies coprime factorization*, IEEE Transactions on Automatic Control **34** (1989), no. 4, 435–443.

[SR98] P. O. M. Scokaert and J. B. Rawlings, *Constrained linear quadratic regulation*, IEEE Transactions on Automatic Control **43** (1998), no. 8, 1163–1169.

[Ste94] M. C. Steinbach, *A structured interior point SQP method for nonlinear optimal control problems*, pp. 213–222, Birkhauser Verlag Basel, Switzerland, 1994.

[SW81] R. Schneider and J. A. Wieacker, *Approximation of Convex Bodies by Polytopes*, Bull. of the London Math. Soc. **13** (1981), no. 2, 149–156.

[SW99] E. D. Sontag and Y. Wang, *New characterizations of input-to-state stability*, IEEE Transactions on Automatic Control **44** (1999), 648–654.

[TJB03a] P. Tøndel, T. A. Johansen, and A. Bemporad, *An algorithm for multiparametric quadratic programming and explicit MPC solutions*, Automatica

39 (2003), 489–497.

[TJB03b] P. Tøndel, T.A. Johansen, and A. Bemporad, *Evaluation of Piecewise Affine Control via Binary Search Tree*, Automatica **39** (2003), no. 5, 945–950.

[TTT] K. C. Toh, R. H. Tütüncü, and M. J. Todd, *SDPT3 – a Matlab software package for semidefinite-quadratic-linear programming, version 3.0*, `http://www.math.cmu.edu/~reha/sdpt3.html`.

[VBN02] L. Vandenberghe, S. Boyd, and M. Nouralishahi, *Robust linear programming and optimal control*, Proc. 15th IFAC World Congress (Barcelona, Spain), 2002.

[Vid93] M. Vidyasagar, *Nonlinear Systems Analysis*, 2nd ed., Prentice Hall, 1993.

[WB08] Y. Wang and S. Boyd, *fast_mpc: code for fast model predictive control*, `http://www.stanford.edu/~boyd/fast_mpc`, Sept 2008.

[WB10] ______, *Fast Model Predictive Control Using Online Optimization*, IEEE Transactions on Control Systems Technology **18** (2010), 267–278.

[Wil70] J. L. Willems, *Stability theory of dynamical systems*, Thomas Nelson and Sons, London, England, 1970.

[WJM07] Y. Wang, C. N. Jones, and J. M. Maciejowski, *Efficient point location via subdivision walking with application to explicit MPC*, European Control Conference (Kos, Greece), July 2007.

[Wol59] P. Wolfe, *The simplex method for quadratic programming*, Econometrica **27** (1959), 382–398.

[Wol63] ______, *A technique for resolving degeneracy in linear programming*, SIAM Journal **11** (1963), 205–211.

[Wri93] S. J. Wright, *Interior-point methods for optimal control of discrete-time systems*, Journal of Optimization Theory and Applications **77** (1993), 161–187.

[Wri97a] ______, *Applying New Optimization Algorithms To Model Predictive Control*, Chemical Process Control-V, CACHE, vol. 93, 1997, pp. 147–155.

[Wri97b] ______, *Primal-dual interior-point methods*, Society for Industrial Mathematics, 1997.

[YW02] E. A. Yildirim and S. J. Wright, *Warm-Start Strategies In Interior-Point Methods For Linear Programming*, SIAM Journal on Optimization **12** (2002), 782–810.

[Zie94] G. M. Ziegler, *Lectures on polytopes*, Springer, 1994.

[ZJM08] M. N. Zeilinger, C. N. Jones, and M. Morari, *Real-time suboptimal Model Predictive Control using a combination of Explicit MPC and Online Computation*, Proc. of the 47th IEEE Conf. on Decision and Control (Cancun, Mexico), Dec. 2008, pp. 4718 – 4723.

[ZJM10] ———, *Robust stability properties of soft constrained MPC*, Proc. 49th IEEE Conf. on Decision and Control (Atlanta, USA), Dec. 2010.

[ZJM11] ———, *Real-time suboptimal model predictive control using a combination of explicit MPC and online optimization*, IEEE Transactions on Automatic Control (2011), to appear.

[ZJRM09] M. N. Zeilinger, C. N. Jones, D. M. Raimondo, and M. Morari, *Real-time MPC – Stability through Robust MPC design*, Proc. of the 48th IEEE Conf. on Decision and Control (Shanghai, China), Dec. 2009, pp. 3980 – 3986.

[ZJRM11] M. N. Zeilinger, C. N. Jones, D. Raimondo, and M. Morari, *On real-time robust MPC*, IEEE Transactions on Automatic Control (2011), submitted.

[ZM95] A. Zheng and M. Morari, *Stability of Model Predictive Control with Mixed Constraints*, IEEE Transactions on Automatic Control **40** (1995), 1818–1823.

Curriculum Vitae

Melanie Zeilinger
born May 9th, 1982 in Stuttgart, Germany.

2006 – 2011	Doctorate at the Automatic Control Laboratory, ETH Zürich, Switzerland (Dr. sc. ETH Zürich)
2006	Internship at the DaimlerChrysler research department for Truck Product Creation, Stuttgart, Germany
2005 – 2006	Diploma Project at the Department of Chemical Engineering, University of California, Santa Barbara, USA
2001 – 2006	Undergraduate studies in Engineering Cybernetics, University of Stuttgart, Germany (Dipl.-Ing.)
1992 – 2001	Lise-Meitner-Gymnasium, Remseck, Germany (High-School, Abitur)