Dominic Groß

Distributed Model Predictive Control with Event-Based Communication

kassel
university
press

Dominic Groß

Distributed Model Predictive Control with Event-Based Communication

kassel
university
press

This work has been accepted by the Faculty of Electrical Engineering / Computer Sciences of the University of Kassel as a thesis for acquiring the academic degree of Doktor der Ingenieurwissenschaften (Dr.-Ing.).

Supervisor: Prof. Dr.-Ing. Olaf Stursberg
Co-Supervisor: Prof. Dr.-Ing. Jan lunze

Further members of examination board:
Prof. Dr. rer. nat. Arno Linnemann
Prof. Dr. rer. nat. Bernhard Sick

Defense day: 12th November 2014

Bibliographic information published by Deutsche Nationalbibliothek
The Deutsche Nationalbibliothek lists this publication in the Deutsche Nationalbibliografie;
detailed bibliographic data is available in the Internet at http://dnb.dnb.de.

Zugl.: Kassel, Univ., Diss. 2014
ISBN 978-3-86219-910-5 (print)
ISBN 978-3-86219-911-2 (e-book)
URN urn:nbn:de:0002-3911

www.upress.uni-kassel.de/

Printed in Germany

Contents

Summary

In this thesis, several algorithms for distributed model predictive control over digital communication networks with parallel computation are developed and analyzed. Distributed control aims at efficiently controlling large scale dynamical systems which consist of interconnected dynamical systems by means of communicating local controllers. Such distributed control problems arise in applications such as chemical processes, formation control, and control of power grids. In distributed model predictive control the underlying idea is to solve a large scale model predictive control problem in a distributed fashion in order to achieve faster computation and better robustness against local failures. Distributed model predictive control often heavily relies on frequent communication between the local model predictive controllers. However, a digital communication network may induce uncertainties such as a communication delays, especially if the load on the communication network is high. One topic of this thesis is to develop a distributed model predictive control algorithm for subsystems interconnected by constraints and common control goals which is robust with respect to time-varying communication delays.

The main focus of this thesis is to reduce the communication requirements in distributed model predictive control by means of event-based communication, i.e. a controller only communicates if a triggering condition is met. While the paradigm of event-based communication has been analyzed in great detail in the context of networked control systems it has not been applied to distributed model predictive control. In this thesis a well-known cooperative distributed model predictive control algorithm, which optimizes the input sequences of the subsystems in parallel, is extended to event-based communication. In the original algorithm every controller communicates with every other controller in every iteration. No results with respect to the convergence rate of the algorithm are available in the literature. Therefore, the convergence properties of the cooperative distributed model predictive control algorithm with periodic communication between the controllers are analyzed in detail, a bound on the convergence rate is derived, and two approaches to choose parameters used in the algorithm are discussed.

Based on these results two approaches to cooperative distributed model predictive control with event-based communication are developed. In the first method the convergence results are used to analyze how the locally optimized input sequences influence the global convergence. This allows defining communication events for each controller which are only triggered if this results in a sufficient improvement of closed-loop performance. If a communication event is triggered for a controller the controller has to communicate with all other controllers. The second approach is

based on analyzing when and between which controllers communication is required in order to achieve good closed-loop performance. The resulting triggering functions answer the question of when to communicate for each pair of controllers, thereby further reducing the load on the communication network. These results also allow quantifying the performance loss due to communication delays and packet loss. Another aim of this thesis is to extend some of the results on distributed MPC with event-based communication to the class of piecewise affine dynamics.

Distributed model predictive control often only results in suboptimal solutions, for example when an iterative algorithm has to be terminated early due to time constraints. This suboptimality, as well as uncertain communication and control goals such as consensus can significantly complicate the stability analysis. To account for these aspects, the notion of input-to-state practical stability (ISpS) and ISpS-Lyapunov functions is extended in this thesis to allow considering stability with respect to a set (e.g. a consensus subspace), and to obtain stronger stability results for suboptimal (distributed) model predictive control. This result is used throughout the thesis for stability analysis of the proposed algorithms for different problems setups, i.e. with and without terminal constraint and for different classes of interconnections.

Acknowledgments

First and foremost, I would like to thank my advisor Prof. Dr.-Ing. Olaf Stursberg for his guidance and encouragement during the past four years. He gave me the freedom to follow my ideas, was always supportive, and provided valuable advice. I am grateful to Prof. Dr. rer. nat. Arno Linnemann for always finding time for fruitful discussions and his courses which first sparked my interest in control systems.

I would like to thank my former colleagues Dr.-Ing. Matthias Rungger, for his helpful advice when I first started on this endeavor, and Dipl.-Ing. Martin Jilg, with whom I collaborated on distributed control. While mostly unrelated to the results of this thesis, I greatly enjoyed our collaboration and discussions.

Furthermore, I am grateful for the support by the German Science Foundation (DFG) within the research priority program 1305 “Control Theory of Digitally Networked Dynamical Systems” and EU-funded Network of Excellence HYCON 2. Both the financial support from the German Science Foundation as well as the many workshops and discussions within the research priority program 1305 are greatly appreciated. I would like to thank Prof. Dr.-Ing. Jan Lunze for his willingness to be on the thesis committee, and him and his team for their commitment in coordinating the research priority program 1305 and the work on the book [74].

Last but not least, I would like to thank my parents for their support and encouragement over the years and my girlfriend Pia for her love and support during the long hours of writing this thesis.

Part I.

Introduction and Theoretical Background

1. Introduction and Literature Review

1.1. Introduction

In this thesis, distributed model predictive control algorithms for interconnected dynamical systems are developed and analyzed. The general problem structure is shown in Figure 1.1. The overall control system consists of interconnected dynamical systems $\mathcal{P}^i$ and local model predictive controllers $\mathcal{C}^i$, which exchange information over a digital communication network $\mathscr{C}_{k,p}$. The subsystems may be interconnected either physically or through a common control goal. Physical interconnections as well as common control goals typically arise in applications such as chemical processes, water distribution systems, or control of power grids. In contrast, dynamical systems only coupled by a common control goal are often encountered in formation control problems, for example autonomous vehicles tasked with keeping a formation while following a lead vehicle.

Model predictive control (MPC) is an optimal control method based on planning optimized future control actions by means of online numerical optimization [90]. A main advantage of MPC is that constraints on the operating region and control inputs can be considered, and the performance of the closed-loop can be optimized

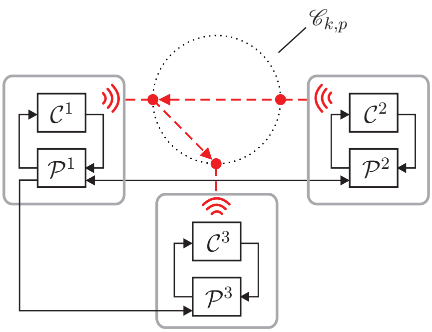

Figure 1.1.: Dynamical systems $\mathcal{P}^i$ with local model predictive controllers $\mathcal{C}^i$, interconnections (black) and communication links (dashed, red).

with respect to a cost function. The cost function can be used to encode control goals, such as stabilization of a set point, tracking of a reference trajectory, or specifications arising from economic considerations [3]. The main idea behind MPC is to utilize a model of the dynamical system under control to predict the future behavior of the system for an input sequence planned over a finite time horizon. This planned input sequence is optimized in each time step in order to achieve optimal performance and to accommodate hard constraints. The first input of the predicted sequence is then applied to the dynamical system and the procedure is repeated in every time step, such that the controller always plans ahead. MPC has received much interest in the field of control theory as well as in a wide range of applications, and a comprehensive theory for analyzing stability, robustness, and optimality of MPC is available if a single centralized MPC is considered (cf. [49], [90]). This is not the case for the distributed setting shown in Figure 1.1, where the assumptions made in the literature with regard to the class of systems, types of interconnections, and communication between local model predictive controllers widely vary and many open questions remain.

Typically, control problems arising from physical interconnections are modeled as interconnected dynamics of the subsystems $\mathcal{P}^i$ and have been tackled to a considerable extent within the framework of decentralized control (cf. [99], [75]). Within this framework, the structure of the interconnection between the subsystems is exploited in order to decompose the overall system into a set of smaller subsystems, for which control laws only requiring knowledge of the local state of each subsystem may be designed. Such a control law is called decentralized. However, in the general case no stabilizing decentralized controller may exist, and even if it exists, the closed-loop performance may be significantly degraded compared to a centralized approach. Therefore, decentralized control of strongly interconnected systems is a challenging problem. At the same time, centralized control of large scale dynamical systems is often not desirable because it requires fast transmission of large amounts of data to and from a centralized location, and the control system is more sensitive to faults (e.g. breakdown of the centralized controller). Furthermore, centralized model predictive control of large scale dynamical systems may not be feasible due to the computational complexity of the resulting optimization problems.

Today, embedded systems and digital communication networks can be deployed at low cost and in a wide range of environments, while offering ample computational power and bandwidth. These technologies can be utilized to increase control performance by exchanging information between the controllers of different subsystems. This results in distributed control laws [94] which can be used if no stabilizing decentralized control law exists, and which typically also offer increased performance compared to decentralized control. Based on this paradigm, distributed MPC aims to combine the advantages of decentralized MPC [80] and centralized MPC by using communicating local controllers to spread the computational complexity across the controllers of the subsystems, while at the same time offering increased closed-loop performance compared to decentralized control. Typically, two main challenges arise in such a distributed control scheme: the communication may be uncertain, and the local controllers lack information about the state of the overall system. The thesis aims to provide insight into these problems for some classes of interconnected linear systems, and extend some results to piecewise affine systems.

From the point of view of communication aspects such as the scheduling protocols of the underlying communication network, time-varying delays, and packet loss, often have to be considered. While the effect of these phenomena on closed-loop stability and performance of centralized and, more recently, decentralized controllers connected to multiple sensors and actuators through a communication network has been analyzed in the area of networked control systems (NCS), few results in this direction are available for distributed model predictive control or distributed control in general. Besides the general advantages mentioned earlier, MPC seems particularly suited to deal with uncertain communication, because the planned state and input sequences can be used to compensate for network effects.

The second, more important, challenge is that the local controllers lack information about the state of the overall system. Exchanging the states and planned inputs between all controllers via the communication network, if feasible, results in a high load on the communication network. This is not desirable, because it often results in large delays or packet loss. Hence, the main aim of this thesis is to develop distributed model predictive algorithms in which a communication link between two controllers $\mathcal{C}^i$ is not always active, but only activated if it is required to achieve the desired closed-loop performance. This raises the question of how to achieve a good trade-off between control performance and load on the communication network. In particular, it is of interest to analyze when and between which controllers communication is required in order to achieve high performance. In the context of networked control systems the first question of when to communicate has lead to the paradigm of event-based control, in which communication does not occur at fixed time intervals, but only if a triggering condition is met. The idea of this thesis is to extend this concept to the communication between distributed model predictive controllers. This approach is motivated by the observation that: (i) there is often no significant change in the communicated data between subsequent messages and that (ii) the input computed by a local controller is often only weakly affected by information communicated by controllers of weakly interconnected subsystems.

Finally, the thesis aims to extend some of the results to the case of piecewise affine subsystems, which can be used to approximate hybrid and nonlinear dynamics [53]. Hybrid dynamics are characterized by a combination of continuous and discrete states and inputs and can be used to model a wide range of dynamical systems [76]. While it may seem that results on distributed MPC for linear systems can easily be extended to piecewise affine systems, this is usually not the case due to the possibly discontinuous and complex dynamics exhibited by this class of systems. Consequently, only few results in the literature on distributed MPC are applicable to piecewise affine systems.

Outline of the Dissertation

In the remainder of this chapter relevant publications from the fields of network control systems and distributed control, with a focus on distributed model predictive control, are reviewed. Chapter 2 presents the theoretical background and begins with a formal definition of the problem setup, i.e. the system dynamics, control goals, and different classes of interconnections considered throughout this thesis are introduced. Next, the concept of input-to-state practical stability (ISpS) is introduced, and a novel sufficient condition for ISpS using ISpS-Lyapunov functions is presented. The chapter closes with an introduction to basic concepts from optimization theory and an overview of results from the field of distributed optimization.

Distributed MPC algorithms with communication at each time step (and iteration) are presented in Chapter 3 and 4. A distributed MPC algorithm for dynamically decoupled systems with common control goals and delayed communication is presented in Chapter 3. Chapter 4 starts by briefly reviewing a cooperative distributed MPC algorithm. This algorithm is used as the starting point for the results on event-based communication in the remainder of the thesis. Subsequently, results on the convergence rate and choice of parameters of the algorithm are provided.

These results are required for the main contribution of this thesis with respect to event-based communication in distributed MPC, which are presented in Chapter 5 to 7. In Chapter 5, a cooperative distributed MPC algorithm, in which a controller communicates with all other controllers if an event is triggered, is developed and analyzed. In Chapter 6, this idea is extended to events which trigger communication between specific controllers. These triggering conditions provide insight into the question of when and between which controllers communication is required. Furthermore, the proposed framework allows to analyze the impact of communication delays and packet loss on the closed-loop performance of the algorithm.

Results on distributed model predictive control with event-based communication for piecewise affine systems are presented in Chapter 7. Finally, in Chapter 8 the results are summarized, the proposed distributed MPC schemes are compared, and some possible directions for future research are discussed.

1.2. Literature Review

This section provides an overview over the relevant literature from the field of networked control systems, distributed control and distributed MPC. Distributed MPC algorithms are classified by their underlying architectures, communication requirements, as well as the assumptions made with respect to the interconnections between subsystems. Based on this classification the state of the art of distributed MPC algorithms with respect to the aims of this thesis is discussed in more detail.

For a broader introduction to the field of control over digital communication networks the reader is referred to the book [74].

Networked Control Systems with single-loop Structure

Within the field of networked control systems the research mostly focuses on the analysis of stability and performance when a given controller, which was designed without considering the effects of the communication network, is connected to sensors and actuators via a digital communication network. Specifically, effects such as communication delays, packet loss, clock offsets, quantization, and the influences of protocols for scheduling and access to the communication network may have to be considered (cf. [54]). For example, in the popular try-once-discard (TOD) protocol only the node with the largest difference to its last communicated value is granted access to a shared communication network [113].

An important development in this area is the paradigm of event-based control, in which communication does not occur periodically but only when an event occurs. These events are often based on the difference of the current value to a previously communicated value or directly based on the decrease of Lyapunov functions [89]. In [54] networked control is investigated in a very general setting, in which nonlinear dynamics in continuous time as well as many network induced phenomena are considered. For a given system, controller, and protocol (e.g. TOD), small gain type arguments are used to obtain bounds on the communication delays and transmission intervals of the underlying communication network which guarantee a certain degree of performance. For linear systems and controllers less conservative results are obtained in [25] by integrating the system model, controller, a model of the communication network, and scheduling protocols into a switched linear system.

To further reduce the load on the communication network, event-based control is combined with a model based event-generator in [77]. The main idea is to use a model of the dynamical system in both the controller and the event-generator, which is located at the sensor, to predict the behavior of the closed-loop system. In this scheme, communication is only required if the actual behavior of the dynamical system differs from the predicted behavior (e.g. because of disturbances).

In all these cases, the controller and the dynamical system are described in continuous time and communication events may be triggered at any time. Therefore, the state of the system needs to be sampled continuously and the underlying communication network has to be designed for the shortest possible time between events which may arise. In order to improve resource allocation and avoid the continuous monitoring of the system state, periodic event-triggered control has been proposed in [52]. In this framework, the triggering conditions are only evaluated at periodic sampling times, thereby combining the advantages of periodic sampling and communication with the reduced amount of communication resulting from event-based control.

In contrast, only few results on MPC considering a wider range of network induced uncertainties or event-based communication are available. For example, centralized model predictive control over communication networks has been considered in [9], where stochastic delays and sampling intervals are considered, and in [8], where the communication between the sensors and the model predictive controllers is event-based.

Networked Control Systems with multiple control loops

The approaches discussed so far consider the case of a single controller. Recently, decentralized control with event-based communication between the local sensors, decentralized controllers, and local actuators has been analyzed in [4] considering network induced uncertainties. The case without network induced uncertainties is considered in more detail in [106]. In [74] Section 5.4 and Section 5.6 optimal decentralized event-based control based on dynamic programming is discussed. In all these works the controllers are decentralized, i.e. do not use measurements or inputs of interconnected subsystems. To achieve improved performance, the approach from [106] has been extended in [105] by introducing additional events, which trigger communication between the decentralized controllers.

Generally, the paradigm of event-based control is well suited to provide an answer to the question of when to communicate to trade-off load on the communication network and robustness / performance. However, the question of where (i.e. between which controllers) communication is required in a distributed control structure remains largely unanswered within this framework.

Linear Distributed Control

In contrast to these results on event-based control, the research in the field of linear distributed control is mostly concerned with designing controllers which achieve a good trade-off between control performance and the number of communication links between controllers. The main idea in this line of thought is that the distributed controllers can directly access the local sensors and actuators and communicate with each other. The goal is to design a controller and a communication topology which is as sparse as possible. For example, simultaneous optimization of distributed state feedback controllers and the underlying communication topology with respect to a quadratic closed-loop performance criterion and the cost incurred by each communication link is considered in [45]. Similarly, designing sparse linear quadratic regulators has been investigated [71] and the design of sparse $\mathcal{H}_\infty$ controllers has been considered in [98]. In other words, these approaches provide insight into which controllers need to communicate in order to achieve good closed-loop performance, but the communication is time-triggered and periodic.

Distributed Model Predictive Control

Many significantly different algorithms for distributed MPC have been proposed in the literature and the assumptions with respect to the type of interconnections between subsystems and the communication network differ largely. While this makes a comparison difficult, the algorithms can be classified according to these aspects. This classification is based on the categories previously introduced by the author in [48].

With respect to the **communication**, the following categorization is possible:

(a) It is assumed that the communication between the controllers does not induce any uncertainties or delays, or their effect is assumed to be negligible. Therefore, all network properties are ignored in the analysis. The vast majority of distributed MPC algorithms (e.g. [91], [112], [83], [26], [50], [86], [108], [65], [37], [59], [104]) fall into this category.

(b) The communication channel is assumed to introduce a constant delay (e.g. [33], [24], [57]).

(c) The communication network induces bounded, time-varying delays (e.g. [42], [72], [47]).

(d) Packet loss induced by the communication network is considered in a worst-case fashion, e.g. by utilizing upper bounds on the number of consecutive packet losses (e.g. [42], [1])

(e) The communication network induces bounded, time-varying delays and packet loss (e.g. [42])

In general, relatively few investigations study the effect of delayed information exchange and there are no results for distributed model predictive control which consider a network model as detailed as the ones used to analyze the impact of communication delays, packet loss, clock offsets, and protocols in the field of network control systems (e.g. [54]).

With respect to the **topology of the communication network** and the type of information exchanged the following cases are considered:

(a) Any local controller can exchange state and optimization variables with all other controllers at the same time (cf. [112], [104]).

(b) Only the controllers of subsystems which are directly interconnected exchange state and optimization variables. Commonly this is referred to as neighboring communication (cf. [47],[83],[91],[37]).

(c) Only the states of neighboring subsystems are exchanged (cf. [61], [1]).

An exchange of information between all controllers, if feasible, usually results in an unnecessarily high load on the communication network and negatively affects scalability. On the other hand, the effects of only partially available information can significantly complicate the design and analysis of distributed MPC algorithms. Thus, a main aim of this dissertation is to analyze the effect of communicated information on the overall control performance and only activate a communication link if it is required.

Distributed model predictive control algorithms can be further classified with respect to the **frequency** at which local optimizations and communication between the controllers take place:

(a) Controllers optimize and exchange information only once per sampling period (non-iterative), for example see [42], [30], [26], [83], [108], [91].

(b) Information is optimized and exchanged iteratively (i.e. multiple times) within each sampling period (cf. [112], [104], [37]).

While iterative algorithms allow for higher performance and may result in the centralized optimal solution, they often require many iterations and are often not suited to deal with communication delays and packet loss.

A related aspect is the **architecture** of the algorithm. In particular the following distinction can be made:

(a) All controllers solve their local problems and exchange information in parallel as discussed in e.g. [33], [47], [37], [86], and [112].

(b) Local problems are solved in a given sequence and each local controller has to wait for the information of the preceding controller in the sequence before starting its own computations or rely on previously communicated information (cf. [91], [83], [50], [108], [65]).

(c) Local problems are solved in parallel and information is exchanged with a coordinator who solves a master problem and redistributes the results to all controllers (cf. [23], [2], [94], [95])

In practice, sequential algorithms may be problematic for medium to large-scale systems because the local computation times and communication delays between the controllers are aggregated.

Finally, the following **types of interconnections** can be considered:

(a) The subsystems are interconnected by dynamics, i.e. states and inputs of one subsystem influence the states of other subsystems, see e.g. [112], [30], [92].

(b) The subsystems are dynamically decoupled but a common control goal, for instance consensus or synchronization, leads to interconnection by costs (cf. [47], [83], [33]).

(c) In principle, all types of interconnections can also be modeled by common equality and inequality constraints, which involve the states and inputs of different subsystems. Common examples are collision avoidance constraints in formation control (cf. [42], [91]) and interconnected dynamics (cf. [37], [115]).

More detailed comparisons between the methods proposed in this thesis and the algorithms proposed in the literature can be found at the end of the corresponding chapters.

Sequential algorithms for dynamically decoupled linear systems coupled only by constraints have been proposed in [91], [108], [109], [65], and nonlinear systems coupled by constraints and costs are considered in [83]. All these algorithms employ terminal constraints, communicate periodically in time triggered fashion, and do not consider any uncertainties induced by the communication network. While the algorithms proposed in [91], [108], [109], [65] offer robustness with respect to local disturbances, the algorithm in [83] is not robust. A non robust extension of [91], which does not require a terminal constraint can be found in [50]. However, the assumptions required to ensure stability of the algorithm are very hard to verify in general. The case of decoupled dynamics, coupled constraints and costs typically arises in applications such as formation control of robots or control of groups of autonomous vehicles. In such a scenario, the controllers often employ wireless communication, which is inherently uncertain, and have limited local computation power. This is why time-varying communication delays as well as computation times should be considered. However, only a few limited results in this direction were available at the outset of this thesis. Therefore, one of the goals of this thesis is to develop a method for the case of decoupled dynamics which offers robustness with respect to communication delays and offers parallel computation to mitigate the impact of computation delays. A modified version of the algorithm proposed in [91] with parallel optimization was recently proposed in [86], but no communication delays are considered. It should be noted that the vast majority of the relevant works cited in this section has been developed in parallel to the results presented in this thesis.

To the best of the author's knowledge no non-iterative sequential or parallel algorithm has been proposed which guarantees convergence to the optimal solution for interconnected MPC problems. Nonetheless, several algorithms for dynamically coupled subsystems which use parallel optimization and often only require neighboring communication have been proposed. In most cases, these results rely on the inherent robustness and stability properties of the local controllers. Within this line of research the local controllers are often non-cooperative [30] and control tasks such as set point stabilization, tracking of reference trajectories [29], and so called plug-and-play distributed control (cf. [92], [59]) are considered. In plug-and-play distributed control a subsystem and its local controller may join or leave the overall system during runtime, provided that certain conditions hold. These conditions are checked online in a distributed fashion. Most of these results rely on set-theoretic small-gain-type conditions and therefore may not be suitable for strongly interconnected subsystems. Similarly, in [110] dissipativity properties are utilized in order to obtain a non-iterative distributed MPC scheme with limited information exchange for nonlinear systems in a cascade structure. A non-iterative distributed MPC algorithm with neighboring communication for dynamically interconnected subsystems based on game theory has been proposed (e.g. [79]). However, in comparison this algorithm often exhibits lower control performance than comparable algorithms [2]. Overall, despite the inherent conservativeness of small gain approaches these algorithms have been successfully applied to application examples and utilize a rather sparse communication topology. However, it is not clear how the restriction to neighboring communication affects the control performance, or if a different, equally sparse, communication topology may result in improved performance.

To avoid the drawbacks of non-iterative algorithms so called iterative algorithms have been proposed which are based on the exchange of either primal (i.e. inputs) or dual variables (see Section 2.4 for an introduction to the concepts of duality in optimization and distributed optimization). The idea of this thesis is to start from an iterative scheme with full communication and direct optimization of the local inputs, analyze when and between which controllers communication is required, and provide conditions under which the scheme can be terminated early (i.e. after any number of iterations). Based on the paradigm of event-based control the aim of this thesis is to leverage the flexibility of digital communication networks and decide when and where to communicate by means of triggering functions, which are periodically evaluated online. In the context of dynamically decoupled systems coupled only by cost, a so called event-based scheme is proposed in [28] and [27], in which the controllers exchange information in a time triggered periodic fashion, and the local MPC problems are computed in an event-based fashion. In contrast, the aim of this thesis is to develop algorithms which reduce the load on the communication network by exchanging information between controllers only if communication is required to ensure good control performance.

Iterative schemes based on dual decomposition have been proposed in e.g. [115], [37], [23], [34]. A distinction should be made between distributed model predictive control algorithms, which consider the closed-loop nature of model predictive control (cf. [34]), and results on distributed optimization algorithms [36]. The latter class of algorithms aims at directly solving a centralized MPC problems and focuses on aspects such as computation time, optimality, and convergence properties. Typically, closed-loop properties are not considered in this context. A brief overview of results from distributed optimization for control is given in Section 2.4.

In the context of duality based distributed MPC, [37] describes an iterative algorithm based on dual decomposition and a distributed suboptimal stopping criterion. However, it is not guaranteed that the distributed stopping criterion holds after a finite number of iterations. A primal-dual hierarchical distributed MPC algorithm, which guarantees convergence to a feasible solution in a finite number of iterations has been proposed in [23]. While dual decomposition can be used to tackle a wide class of problems, the number of iterations and messages required to obtain a stabilizing solution is often large. Furthermore, the primal cost may increase over the iterations, which is problematic if the optimization has to be terminated early due to time or communication constraints.

A cooperative iterative distributed MPC algorithm which directly optimizes the inputs (i.e. the primal variables) of subsystems interconnected by states, inputs, and possibly input constraints is investigated in [112] and [104]. This algorithm ensures stability of the closed-loop when terminated at any iteration, and is guaranteed to converge to the global optimal solution in the limit (i.e. for an infinite number of iteration). However, no results on the convergence rate or how to chose crucial parameters of the algorithm are available in the original works. Furthermore, the algorithm requires a full information exchange between all controllers in every iteration and time step. To reduce the load on the network, a hierarchical algorithm was proposed in [103], in which subsystems are grouped into clusters by the designer and communication within a cluster is used more frequently than between clusters. However, no results are available on how to perform the clustering to achieve a good trade-off between communication and performance.

Aims of the Dissertation

In conclusion, it can be seen that network induced uncertainties such as time-varying communication delays have received little attention in the literature on distributed MPC. However, these aspects may have strong implications on stability and feasibility of distributed MPC. Therefore, one aim of this thesis is to analyze the impact of **uncertain communication** on distributed MPC, and to provide results on the robustness with respect to time-varying delays.

Similarly, the use of event-based communication has been studied to a considerable extent for centralized and decentralized controllers in the field of networked control systems, but no results are available for distributed MPC. In particular, it has been shown in the context of networked control systems, that event-based communication is well suited to reduce the load on the communication network while maintaining high control performance. The main aim of this thesis is to develop distributed MPC algorithms which utilize **event-based communication** between the controllers. Specifically, it needs to be analyzed **when** and **between which** controllers communication is required in order to achieve good closed-loop performance.

Finally, only few results on distributed MPC are directly applicable to **hybrid systems**. Hence, this thesis aims to extend some of the results on **distributed MPC with event-based communication** to the class of **piecewise affine dynamics**.

2. Problem Setup and Theoretical Background

In this Chapter the problem setup, notation, and relevant theoretical background on stability analysis and mathematical optimization is presented. The following section introduces the system dynamics, control goals, and model of the communication network considered throughout this thesis. In Section 2.3 relaxed conditions for input-to-state practical stability are presented which allow considering suboptimal solutions, robustness aspects, and stability with respect to a set (e.g. consensus). Finally, in Section 2.4 some basic concepts of mathematical optimization, which are required to analyze and discuss the properties of different distributed MPC algorithms, are presented.

2.1. Problem Setup and Notation

The general problem structure considered throughout this thesis is shown in Figure 1.1. The overall control system consists of a set $\mathcal{N} = \{1, \dots, N_s\}$ of possibly interconnected dynamic systems $\mathcal{P}^i$ with local model predictive controllers $\mathcal{C}^i$. The subsystems may be interconnected by dynamics or through a common cost function. To achieve cooperation of the controllers, a distributed control scheme is used in which the controllers $\mathcal{C}^i$ exchange information over a communication network $\mathscr{C}_{k,p}$, which may induce uncertainties such as packet loss or delays. The aim of this thesis is to develop distributed model predictive controllers in which a communication link between two controllers is not always active, but only activated if it is required to achieve the desired closed-loop performance. The time-invariant interconnections between subsystems due to dynamics, constraints, and costs may be described by a directed graph $\mathscr{G} = (\mathcal{N}, \mathcal{I})$, where the set of nodes corresponds to the subsystems and each edge $\mathcal{I} \subseteq \mathcal{N} \times \mathcal{N}$ represents an interconnection between two subsystems. The index set $\mathcal{N}^i = \{i_1, \dots, i_{N_s^i}\} \subseteq \mathcal{N}$ contains the indices of all N_s^i subsystems interconnected with subsystem $\mathcal{P}^i$: $\mathcal{N}^i := \{j | (i,j) \in \mathcal{I} \text{ or } (j,i) \in \mathcal{I}, i \neq j\}$. In the following it is assumed that $\mathscr{G}$ is connected. If $\mathscr{G}$ is not connected, the problem can be decomposed into a set of fully disconnected problems which often results in faster convergence and reduced communication.

System Dynamics

The dynamics of the overall system are given by the following time-invariant discrete-time system with discrete-time k:

$$x_{k+1} = f(x_k, u_k), \tag{2.1}$$

where $x_k \in \mathbb{X} \subseteq \mathbb{R}^n$, and $u_k \in \mathbb{U} \subseteq \mathbb{R}^m$ are the state and input vector of the overall system, $f : \mathbb{X} \times \mathbb{U} \to \mathbb{R}^n$, and $\mathbb{X}$ and $\mathbb{U}$ are the state and input constraints. The state and input vector are partitioned, such that $x_k = (x_k^1; \ldots; x_k^{N_s})$, $u_k = (u_k^1; \ldots; u_k^{N_s})$ and $x_k^i \in \mathbb{R}^{n^i}$, where e.g. $(x_k^1; \ldots; x_k^{N_s}) := [(x_k^1)^T \; \ldots \; (x_k^{N_s})^T]^T$ denotes a stacked column vector, and $u_k^i \in \mathbb{R}^{m^i}$ are the local state and input variables of $\mathcal{P}^i$, which may be coupled by the dynamics, costs, or constraints. In particular, the subsystem dynamics considered are linear time-invariant, such that the dynamics of the overall system are given by

$$x_{k+1} = Ax_k + Bu_k, \tag{2.2}$$

with $A \in \mathbb{R}^{n \times n}$, $B \in \mathbb{R}^{n \times m}$. In Chapter 7, decoupled piecewise affine systems:

$$x_{k+1}^i = A_{\mathfrak{p}^i}^i x_k^i + B_{\mathfrak{p}^i}^i u_k^i + g_{\mathfrak{p}^i}^i, \quad \text{if } x_k^i \in \mathbb{X}_{\mathfrak{p}^i}^i, \tag{2.3}$$

are considered, where $\mathbb{X}_{\mathfrak{p}^i}^i$ is a convex polyhedral region of the partitioned state-space of $\mathcal{P}^i$. The affine dynamics parametrized by $A_{\mathfrak{p}^i}^i$, $B_{\mathfrak{p}^i}^i$, $g_{\mathfrak{p}^i}^i$ is valid in the region with index $\mathfrak{p}^i \in \{1, \ldots, N_{\mathfrak{p}}^i\}$.

In the following $x_{k+l|k}$ denotes the state at time $k+l$ predicted at time k. Bold vectors denote a sequence over a finite prediction horizon $N \geq 2$, i.e. $\mathbf{u}_k^i = (u_{k|k}^i; \ldots; u_{k+N-1|k}^i) \in \mathbb{R}^{Nm^i}$, and $\mathbf{x}_k^i = (x_{k|k}^i; \ldots; x_{k+N|k}^i) \in \mathbb{R}^{(N+1)n^i}$ denotes the sequence of planned inputs and predicted states of the subsystem $\mathcal{P}^i$. The stacked global state and input vector are denoted by $\mathbf{x}_k = (\mathbf{x}_k^1; \ldots; \mathbf{x}_k^{N_s}) \in \mathbf{X} \subseteq \mathbb{R}^{(N+1)n}$ and $\mathbf{u}_k = (\mathbf{u}_k^1; \ldots; \mathbf{u}_k^{N_s}) \in \mathbf{U} \subseteq \mathbb{R}^{Nm}$, with corresponding constraints $\mathbf{X}$, $\mathbf{U}$ obtained by stacking $\mathbb{U}$ and $\mathbb{X}$.

Control Objectives and Centralized MPC Problem

The control objective is specified by a set, which is used to define the global cost function:

$$\Sigma := \{z \in \mathbb{X} \times \mathbb{U} \,|\, z = (x; u), \, \Gamma_x x = 0, \Gamma_u u = 0\}, \tag{2.4}$$

with matrices $\Gamma_x \in \mathbb{R}^{n \times n}$, $\Gamma_u \in \mathbb{R}^{m \times m}$. The set Σ either specifies an equilibrium of the overall system (i.e. $z = (0; 0)$) or may be used to define control tasks such as synchronization (e.g. $x^1 = \ldots = x^i = \ldots = x^{N_s}$). Based on this set, a cost function $\mathbf{V} : \mathbf{X} \times \mathbf{U} \to \mathbb{R}_{\geq 0}$ is defined which encodes a common objective:

$$\mathbf{V}(\mathbf{x}_k, \mathbf{u}_k) = \|x_{k+N|k}\|_P^2 + \textstyle\sum_{l=0}^{N-1} \|x_{k+l|k}\|_Q^2 + \|u_{k+l|k}\|_R^2, \tag{2.5}$$

where e.g. $\|x_{k+N|k}\|_P^2 = (x_{k+N|k})^T P x_{k+N|k}$ and the weighting matrices are given by $Q := \Gamma_x^T \Gamma_x$, $R := \Gamma_u^T \Gamma_u$ such that $Q = Q^T \succeq 0$ and $R = R^T \succeq 0$. The curled inequality symbols $\succ, \succeq$ denote strict and non-strict matrix inequalities. It directly follows that $\|x\|_Q^2 + \|u\|_R^2 = 0,\ \forall z \in \Sigma$, and $\|x\|_Q^2 + \|u\|_R^2 > 0,\ \forall z \notin \Sigma$. It should be noted that the set Σ is not used as a constraint, but is used to establish properties of the cost function.

Further restrictions for the weighting matrices, such as $R \succ 0$, will be introduced in the following chapters when required. Finally, a terminal constraint $\mathbb{T} \subseteq \mathbb{R}^n$ for the last predicted state $x_{k+N|k}$ may be required in some cases.

Assumption 2.1. *It is assumed that Σ is chosen such that $z_k \in \Sigma$ implies that $\exists u_{k+1} : (x_{k+1}; u_{k+1}) \in \Sigma$. Furthermore, it is assumed, that the state, terminal and input constraints $\mathbb{X}$, $\mathbb{T}$, and $\mathbb{U}$ are full-dimensional, compact and contain 0 in their interior.*

The centralized MPC problem is then given by

$$\begin{aligned} V^*(x_k) := &\min_{u_{k+l|k}, x_{k+l|k}} \mathbf{V}(\mathbf{x}_{k|k}, \mathbf{u}_k) && (2.6)\\ \text{s.t. } & x_{k+l+1|k} = f(x_{k+l|k}, u_{k+l|k}), && \forall l \in \{0, \dots, N-1\} \\ & x_{k+l|k} \in \mathbb{X}, u_{k+l|k} \in \mathbb{U}, && \forall l \in \{0, \dots, N-1\} \\ & x_{k|k} = x_k,\ x_{k+N|k} \in \mathbb{T} \end{aligned}$$

In centralized MPC the state x_k is sampled at each time step k, the problem (2.6) is solved, and the first input of the optimized input sequence $u^*(x_k) = u^*_{k|k}$ is applied to the system. It can be seen that the model predictive controller is defined implicitly through an optimization problem. While the explicit solution to MPC problems with linear and piecewise affine dynamics can, in principle, be computed (cf. [6], [13]) the required computations are too complex even for some small to medium scale systems. Therefore, the explicit computation of model predictive controllers for medium to large scale interconnected systems is not feasible.

Models for Distributed MPC

As discussed in the previous chapters, the aim of this thesis is to develop distributed model predictive control algorithms in which each controller $\mathcal{C}^i$ only optimizes the input sequence $\mathbf{u}_k^i$ and event-based communication between the controllers is used. However, given these requirements, the centralized problem (2.6) is too general. In particular, considering all types of coupling may lead to problems which can only be solved efficiently by a centralized algorithm. For this reason, the vast majority of the literature on distributed MPC focuses on more specific cases and throughout this thesis different cases focusing on specific types of coupling will be considered and commonalities between the different problem setups will be discussed.

The first case focuses on decoupled dynamics and coupling by state constraints and control goals. This structure arises in applications where there is no physical interconnection between the subsystems, but a common control goal should be achieved.

Case 2.1. *The dynamics are decoupled and given either by linear dynamics* (2.2) *with* $A = blkdiag(A^1, \ldots, A^{N_s})$, $B = blkdiag(B^1, \ldots, B^{N_s})$ *and the pairs* (A^i, B^i) *are stabilizable for all* $i \in \mathcal{N}$, *or by piecewise affine dynamics* (2.3) *with* $g_1 = 0$, $\Sigma = \{0\}$ *and the pairs* (A_1^i, B_1^i) *are stabilizable for all* $i \in \mathcal{N}$.

Coupling is induced by a common control goal encoded by potentially fully coupled costs and coupled state constraints $\mathbb{X}$. *The input constraints are compact and decoupled, i.e.* $\mathbb{U} := \mathbb{U}^1 \times \ldots \times \mathbb{U}^{N_s}$.

In the case of linear dynamics (2.2) *it is assumed that the constraints are specified by the following polytopes:*

$$\mathcal{B}_\epsilon^n(0) \subseteq \mathbb{X} = \{x_k \in \mathbb{R}^n \,| C_{\mathbb{X}} x_k \leq b_{\mathbb{X}}\}, \tag{2.7}$$

$$\mathcal{B}_\epsilon^{m^i}(0) \subseteq \mathbb{U}^i = \{u_k^i \in \mathbb{R}^{m^i} | C_{\mathbb{U}}^i u_k^i \leq b_{\mathbb{U}}^i\}, \tag{2.8}$$

where $C_{\mathbb{X}} \in \mathbb{R}^{h_{\mathbb{X}} \times n}$, $b_{\mathbb{X}} \in \mathbb{R}^{h_{\mathbb{X}}}$, $C_{\mathbb{U}}^i \in \mathbb{R}^{h_{\mathbb{U}}^i \times m^i}$, $b_{\mathbb{U}}^i \in \mathbb{R}^{h_{\mathbb{U}}^i}$. $\mathcal{B}_r^n(x)$ *denotes a closed ball of dimension* n, *radius* r, *and with center* x. *For the case of piecewise affine dynamics* (2.3) *the input constraints are defined accordingly but the definition of the state constraints is more involved and will be given in Chapter 7.*

This structure typically arises in formation control problems. For example, to control platooning vehicles in a leader-follower scenario the cost can be used to formulate that all vehicles follow the lead vehicle as closely as possible subject to coupled state constraints which ensure collision avoidance.

In contrast, in the second class of problems only linear dynamics and local constraints are considered. The following cases all result in decoupled local constraints.

Case 2.2. *[104] The dynamics may be fully coupled,* $R \succ 0$ *holds, the pair* (A, B) *is stabilizable and is transformed into a distributed model based on its Kalman canonical form. The terminal constraint* $\mathbb{T}$ *is an equality constraint, and* N *needs to be sufficiently large to zero the unstable modes of the distributed model at the end of the prediction horizon. The decoupled input constraints* $\mathbb{U} := \mathbb{U}^1(x_k) \times \ldots \times \mathbb{U}^{N_s}(x_k)$ *are compact, include so called "stability" constraints, and are given by polytopes which contain* 0 *in their interior. Finally, no state constraints are present, i.e.* $\mathbb{X} := \mathbb{R}^n$. *The endpoint and "stability" constraints do not result in any coupled input constraints (for details see [104]), and the "stability" constraints are not required in this thesis. In addition to the case considered in [104] coupled costs are considered in this thesis.*

This formulation is motivated by physically interconnected subsystems. Typical application examples for this class of interconnections are chemical processes [104], water distribution systems [107], as well as power generation [112]. In contrast, the next case considers physically decoupled subsystems interconnected by a common control goal, such as formation control.

Case 2.3. *The costs are coupled and* $Q \succ 0$, $R \succ 0$, *i.e.* $\Sigma = \{0\}$. *The dynamics are decoupled, i.e.* $A = \mathit{blkdiag}(A^1, \dots, A^{N_s})$, *and* $B = \mathit{blkdiag}(B^1, \dots, B^{N_s})$, *and* (A^i, B^i) *are controllable. The input, state, and terminal constraints are compact and decoupled, i.e.* $\mathbb{U} := \mathbb{U}^1 \times \ldots \times \mathbb{U}^{N_s}$, $\mathbb{X} := \mathbb{X}^1 \times \ldots \times \mathbb{X}^{N_s}$, $\mathbb{T} := \mathbb{T}^1 \times \ldots \times \mathbb{T}^{N_s}$ *and are given by polytopes containing* 0 *in their interior.*

Finally, the last case considers a similar setup to Case 2.2, but without terminal constraint. The motivation for removing the terminal constraint is, that a terminal constraint often requires using a large prediction horizon, complicates the MPC problem and is hard to satisfy when using a decentralized initialization. However, removing the terminal constraint may significantly complicate the theoretical analysis of both centralized and distributed MPC.

Case 2.4. *The costs and dynamics may be fully coupled, it holds that* $R \succ 0$, *and the pair* (A, B) *is stabilizable. The input constraints* $\mathbb{U} := \mathbb{U}^1 \times \ldots \times \mathbb{U}^{N_s}$ *are compact and decoupled and no state constraints and terminal costs are present, i.e.* $\mathbb{T} = \mathbb{X} := \mathbb{R}^n$ *and* $P = 0$.

Assumption 2.2. *For the Cases 2.2 to 2.4, it is assumed that the constraints, if they are applicable to the specific case, are given by polytopes:*

$$\mathcal{B}_\epsilon^{n^i}(0) \subseteq \mathbb{X}^i = \{x_k^i \in \mathbb{R}^{n^i} | C_{\mathbb{X}}^i x_k^i \leq b_{\mathbb{X}}^i\}, \tag{2.9}$$

$$\mathcal{B}_\epsilon^{m^i}(0) \subseteq \mathbb{U}^i = \{u_k^i \in \mathbb{R}^{m^i} | C_{\mathbb{U}}^i u_k^i \leq b_{\mathbb{U}}^i\}, \tag{2.10}$$

$$\mathcal{B}_\epsilon^{n^i}(0) \subseteq \mathbb{T}^i = \{x_k^i \in \mathbb{R}^{n^i} | C_{\mathbb{T}}^i x_k^i \leq b_{\mathbb{T}}^i\}. \tag{2.11}$$

In the case of coupled constraints the decomposition of the global system into subsystems may be altered such that the constraints are decoupled (see e.g. [104]). However, this may result in large local optimization. From a theoretical point of view, the decoupled constraints are crucial for guaranteeing convergence of the cooperative distributed algorithms considered in this thesis to the centralized optimizer despite the fact that each controller only optimizes its local input sequences.

2.2. Model of the Communication Network

In order to evaluate the local control law $\mathcal{C}^i$ the state x_k^j of $\mathcal{P}^j$ and input sequence $\mathbf{u}_k^j$ of $\mathcal{C}^j$, $j \in \mathcal{N} \setminus i$ may be required. Therefore, this information is exchanged between the controllers via a communication network. The topology of the network is modeled by the time-varying communication graph $\mathscr{C}_{k,p} = (\mathcal{N}, \mathcal{E}_{k,p})$ with nodes $\mathcal{N}$ corresponding to the controllers $\mathcal{C}^i$, and a time-varying set of edges $\mathcal{E}_{k,p} \subseteq \mathcal{N} \times \mathcal{N}$. The edges are controlled, e.g. a communication link from $\mathcal{C}^i$ to $\mathcal{C}^j$ can be activated or deactivated by the controller $\mathcal{C}^i$. However, in some cases it is assumed that packets may be delayed or lost by the communication network after a they have been sent by a controller.

Therefore, the states and inputs of interconnected subsystems and controllers may not be known exactly for one of the following reasons:

- the communication link between $\mathcal{C}^j$ and $\mathcal{C}^i$ is not active,
- $\mathcal{C}^j$ was recomputed directly after communication, resulting in a different value,
- the communication network introduces uncertainties, such as delays or packet loss.

To model that the received information does not always match the exact values, the superscript $\hat{(\cdot)}^{i,j}$ denotes a variable used by $\mathcal{C}^i$ instead of the variable $(\cdot)^j$ defined for subsystem $\mathcal{P}^j$. For instance, consider the case shown in Figure 2.1 in which two controllers may exchange information on the local state and input. The local variables $\hat{x}_k^{i,j}$, $\hat{u}_k^{i,j}$ are updated if information is received via the communication network. If no updated information is received, for example due to packet loss, the values $\hat{x}_k^{i,j}$, $\hat{u}_k^{i,j}$ may no longer be identical to x_k^j and u_k^j. In contrast, $\hat{x}_k^{i,j} = x_k^j$ and $\hat{u}_k^{i,j} = u_k^j$ is only guaranteed to hold if no communication imperfections are considered and the controllers do not change their local values without communicating the new values first.

Time and Time Synchronization

An important aspect in distributed control, both from the point of view of communication and control, is time synchronization. From the communication point of view, a deterministic transmission of communicated values can often only be

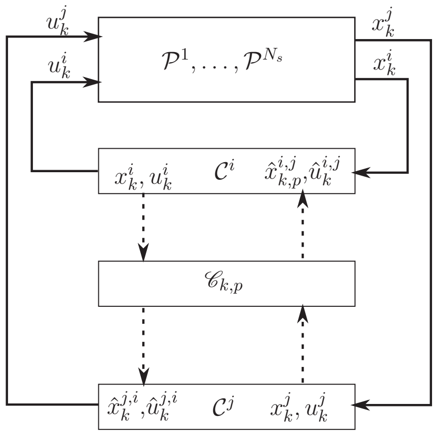

Figure 2.1.: Update of local variables $\hat{x}_k^{i,j}$, $\hat{u}_k^{i,j}$ by communication.

achieved if a corresponding time slot for the transmission was reserved by the network scheduler. Otherwise, collisions and packet loss may occur. Furthermore, communication protocols which do not rely on predetermined schedules but use arbitration to control access to the communication network online also crucially rely on clock synchronization (see e.g. [21]).

Similarly, distributed control algorithms often require time synchronization to ensure that communicated state or input sequences can be interpreted correctly by the receiver. For instance, a state measurement corresponds to a particular absolute time, but if the clocks of the controllers are not synchronized different controllers will associate this measurement with different points in time. Finally, in order to compensate for communication delays, messages sent by the local controllers may contain a timestamp, which allows the receiver to reconstruct possible communication delays if the clocks are synchronized.

In distributed MPC the optimization of the input sequence may be performed iteratively within each time-step. To this end, the extended time vector (k, p) is introduced, where k is the discrete time and p is the iteration index at time k. The corresponding sampling time for the discrete time k is denoted by Δt, and the time for each iteration is denoted by $\Delta t_p \ll \Delta t$. The extended discrete time vector starts at $(0, 0)$ and is updated as follows based on the absolute time t:

$$(k, p) := \begin{cases} (k+1, 0) & \text{if } t = k\Delta t + \Delta t, \\ (k, p+1) & \text{if } t = k\Delta t + p\Delta t_p + \Delta t_p. \end{cases} \tag{2.12}$$

In other words, at the end of each sampling interval Δt the discrete time k is incremented and the iteration index p is reset to 0. Within a sampling interval the iteration index p is incremented when the time Δt_p has passed. The progress of (k, p) is shown on a timeline in Figure 2.2.

Assumption 2.3. *It is assumed that the clocks of the controllers are synchronized and sampling, actuation, and iterations are performed synchronously.*

This assumption is implicitly made in the vast majority of publications dealing with distributed MPC and distributed control in general. In the context of networked control systems few works consider bounded clock offsets / clock jitter

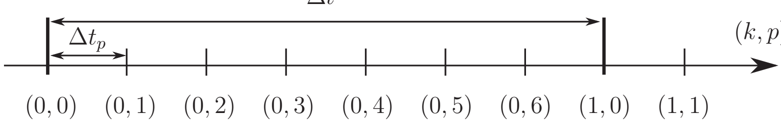

Figure 2.2.: Timeline showing the extended time vector (k, p), sampling time Δt, and time per iteration Δt_p.

but these results are not directly applicable to distributed model predictive control and, in most cases, not directly applicable to distributed control in general. For instance, in [54] jitter is considered for the analysis of networked implementations of a single, explicitly known controller which communicates with multiple sensors, actuators and a shared communication network. However, as discussed before, it is not computationally feasible to compute the explicit solution of distributed MPC.

In the area of distributed optimization several algorithms with asynchronous iterations have been proposed (cf. [17] and the references therein). However, when applying these algorithms to distributed MPC problems sampling and actuation would still require clock synchronization between the controllers. Furthermore, the method proposed in [17] is only applicable to problems with decoupled costs. In [111] a distributed MPC scheme with asynchronous iterations and possibly asynchronous sampling / actuation is proposed, where processes are grouped according to their time scales. For each time scale the cooperative distributed MPC from [112] is applied and the controllers of different time scales are periodically synchronized. This algorithm still requires perfectly synchronized clocks and leads to a rather complex scheme with additional communication requirements which grow with the number of time scales under consideration.

In most cases, the requirement of synchronized clocks is not problematic as shown in the wide literature on clock synchronization in communication networks. Even in multi-hop and wireless networks clock synchronization with high accuracy is possible (cf. [38]) and often required if a deterministic communication protocol is used (see e.g. [20] for a protocol well suited for event-based control in multi-hop networks).

2.3. Input-to-State Stability

As discussed before, the input sequence computed by distributed MPC algorithms may not be optimal and uncertainties induced by the communication network may affect stability of distributed MPC. Furthermore, using event-based communication the closed-loop system is often only practically stable, i.e. the state converges to a bounded neighborhood of the control goal. Throughout this thesis the notion of input-to-state practical stability (ISpS) and ISpS-Lyapunov functions is used as framework for stability analysis. This allows for investigating suboptimal solutions, practical stability, and robustness with respect to uncertain communication.

In the following a definition of ISpS with respect to a set of states is introduced. The main purpose of this extension is to allow for a wider range of control goals (e.g. consensus) instead of only considering stability with respect to a given set point. Based on this definition, conditions for input-to-state practical stability in terms of ISpS-Lyapunov functions are given which are less restrictive than those proposed in the literature (cf. [58], [68]). Specifically, the new conditions allow establishing ISpS with respect to a set and relax a condition which may not hold when using suboptimal or distributed MPC. To establish the ISpS properties the following classes of comparison functions, which were first introduced in [51], are used:

Definition 2.1. *A function $\alpha_c : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ is of class $\mathscr{K}$ if it is continuous, strictly increasing and $\alpha_c(0) = 0$; it is of class $\mathscr{K}_\infty$ if it is a $\mathscr{K}$-function and $\alpha_c(s) \to \infty$ as $s \to \infty$. A function $\beta_c : \mathbb{R}_{\geq 0} \times \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ is of class $\mathscr{KL}$ if, for each fixed $k \geq 0$, the function $\beta_c(\cdot, k)$ is of class $\mathscr{K}$, and for each fixed $s \geq 0$, the function $\beta_c(s, \cdot)$ is decreasing and $\beta_c(s, k) \to 0$ as $k \to \infty$.*

Based on these classes of comparison functions, the following definition introduces the notion of regional input-to-state practical stability (ISpS). The notion of input-to-state stability (ISS) was first proposed by Sontag in the context of continuous time systems (cf. [101]). In the following, let x_{k+l} denote the solution of a perturbed discrete-time dynamical system $x_{k+1} = f_d(x_k, \omega_k)$ at time $k + l$ obtained by forward recursion starting at the state x_k. The following definition is required to ensure existence of the solution under all possible realizations of the bounded disturbance $\omega_k \in \mathbb{W}$ for all $k \geq 0$.

Definition 2.2. *A set $\mathcal{X} \subseteq \mathbb{X}$ is robust forward invariant with respect to the dynamics $f_d : \mathbb{X} \times \mathbb{W} \to \mathbb{R}^n$ if for all $x \in \mathcal{X}$ and all disturbances $\omega \in \mathbb{W}$ holds that $f_d(x, \omega) \in \mathcal{X}$.*

Here a discrete-time version of input-to-state practical stability (ISpS) with respect to a set Σ is used which extends the definitions proposed in [58] and [68] to stability with respect to a set Σ. To this end, let $\|x_k\|_\Sigma := \inf_{z \in \Sigma} \|x_k - z\|$ denote the distance of x_k from the set Σ.

Definition 2.3. *A system $x_{k+1} = f_d(x_k, \omega_k)$ is input-to-state practically stable (ISpS) in $\mathcal{X}$ with respect to a set Σ, if $\mathcal{X}$ is robust forward invariant with respect to $x_{k+1} = f_d(x_k, \omega_k)$, $\omega_k \in \mathbb{W}$, and there exists a $\mathscr{KL}$-function β_c, a $\mathscr{K}$-function γ_c, and a constant $d_c \geq 0$ such that for each $x_0 \in \mathcal{X}$, all $\omega_l \in \mathbb{W}$, $l \in \mathbb{N}_0$, and all $k > 0$ it holds that*

$$\|x_k\|_\Sigma \leq \beta_c(\|x_0\|_\Sigma, k) + \gamma_c(\|\omega_{[0:k-1]}\|) + d_c,$$

where $\omega_{[k_1:k_2]} := (\omega_{k_1}; \ldots; \omega_{k_2})$ denotes a time sequence starting at time k_1 and truncated at time k_2, and $\|\omega_{[k_1:k_2]}\| := \max_{l \in \{k_1, \ldots, k_2\}} \|\omega_l\|$. The system is said to be input-to-state stable (ISS) in $\mathcal{X}$ with respect to Σ if $d_c = 0$.

If Σ only contains the origin (i.e. $\Sigma = \{0\}$) the standard definition of regional or global (if $\mathcal{X} = \mathbb{R}$) ISpS [68] is recovered. If $d_c = 0$ and $\Sigma = \{0\}$ the definition of ISS from [58] is recovered. Finally, the system is asymptotically stable in the sense of Lyapunov if $d_c = 0$, $\Sigma = \{0\}$, and $\omega_k = 0$ for all $k \geq 0$.

The following result is an extension of the results in [68] and shows that regional ISpS (ISS) with respect to a set can be established by means of ISpS- / ISS-Lyapunov functions.

Theorem 2.1. *Let $d_1, d_2 \in \mathbb{R}_{\geq 0}$, let $a_1, a_2, a_3, a_e \in \mathbb{R}_{>0}$ with $a_3 < a_2$, and $L \in \mathbb{N}_{>0}$. Furthermore, let $\alpha_1(s) := a_1 s^{a_e}$, $\alpha_2(s) := a_2 s^{a_e}$, $\alpha_3(s) := a_3 s^{a_e}$, $\sigma \in \mathscr{K}$, and it holds that $\mathcal{X}$ is robust forward invariant with respect to $x_{k+1} = f_d(x_k, \omega_k)$, $\omega_k \in \mathbb{W}$. Let $\mathcal{V} : \mathcal{X} \to \mathbb{R}_{\geq 0}$ be a function such that for all $x \in \mathcal{X}$, it holds that*

$$\alpha_1(\|x\|_\Sigma) \leq \mathcal{V}(x) \leq \alpha_2(\|x\|_\Sigma) + d_1, \tag{2.13}$$

and for all $x_k \in \mathcal{X}$, and all $\omega_l \in \mathbb{W}, l \in \{k, \dots, k+L-1\}$ it holds that

$$\mathcal{V}(x_{k+1}) - \mathcal{V}(x_k) \leq \sigma(\|\omega_k\|) + d_2, \tag{2.14}$$

$$\mathcal{V}(x_{k+L}) - \mathcal{V}(x_k) \leq -\alpha_3(\|x_k\|_\Sigma) + L\sigma(\|\omega_{[k:k+L-1]}\|) + Ld_2. \tag{2.15}$$

Then it holds that system $x_{k+1} = f_d(x_k, \omega_k)$ is ISpS in $\mathcal{X}$ with respect to Σ and the ISpS property holds for

$$\beta_c(s,k) := \alpha_1^{-1}\left(3\varrho^{\lfloor \frac{k}{L} \rfloor}\alpha_2(s)\right), \tag{2.16}$$

$$\gamma_c(s) := \alpha_1^{-1}\left(\frac{3L}{1-\varrho}\sigma(\|\omega_{[0:k-1]}\|)\right), \tag{2.17}$$

$$d_c := \begin{cases} \alpha_1^{-1}\left(3d_1 + \frac{3}{1-\varrho}d_2\right) & \text{if } L = 1, \\ \alpha_1^{-1}\left(3(L+1)d_1 + \frac{3L}{1-\varrho}d_2\right) & \text{if } L > 1, \end{cases} \tag{2.18}$$

where $\lfloor y \rfloor$ denotes the largest integer that does not exceed y, $\varrho = 1 - a_3/a_2$, and $\alpha_i^{-1}(\cdot)$ denotes the inverse function of $\alpha_i(\cdot)$.

Proof. Following the arguments used in [68] for the case $\Sigma = \{0\}$, it holds for all $x \in \mathcal{X} \setminus \Sigma$ that $(\alpha_2(\|x\|_\Sigma))^{-1}(\mathcal{V}(x) - d_1) \leq 1$ and

$$\mathcal{V}(x) - \alpha_3(\|x\|_\Sigma) \leq \mathcal{V}(x) - \frac{\alpha_3(\|x\|_\Sigma)}{\alpha_2(\|x\|_\Sigma)}(\mathcal{V}(x) - d_1). \tag{2.19}$$

With $\varrho = 1 - \frac{a_3}{a_2} \in (0,1)$ it holds for all $x \in \mathcal{X} \setminus \Sigma$ that

$$\mathcal{V}(x) - \alpha_3(\|x\|_\Sigma) \leq \varrho\mathcal{V}(x) + (1-\varrho)d_1. \tag{2.20}$$

Using the same argument as in [68], it can be shown that this inequality holds for all $x \in \mathcal{X}$ because $\mathcal{V}(x) - \alpha_3(\|x\|_\Sigma) = \varrho\mathcal{V}(x) + (1-\varrho)\mathcal{V}(x) \leq \varrho\mathcal{V}(x) + (1-\varrho)d_1$ holds for all $x \in \Sigma$. Next, in contrast to [68], the relaxed conditions (2.14) and (2.15) are used. It directly follows from (2.14) and $(1-\varrho)d_1 \geq 0$ that

$$\mathcal{V}(x_{k+1}) \leq \mathcal{V}(x_k) + (1-\varrho)d_1 + \sigma(\|\omega_k\|) + d_2 \tag{2.21}$$

holds, and substituting (2.20) into (2.15) results in

$$\mathcal{V}(x_{k+L}) \leq \varrho\mathcal{V}(x_k) + (1-\varrho)d_1 + L\sigma(\|\omega_{[k:k+L-1]}\|) + Ld_2. \tag{2.22}$$

Because $\mathcal{X}$ is robust forward invariant the inequalities (2.21) and (2.22) can be applied to obtain an upper bound on the Lyapunov function at a time k and the worst case disturbance is given by considering $\|\omega_l\| \leq \|\omega_{[0:k]}\|$ for all $l \in \{0, \dots, k\}$. Next, (2.21) implies for all $k \in \{0, \dots, L-1\}$ that

$$\mathcal{V}(x_k) \leq \mathcal{V}(x_0) + k(\sigma(\|\omega_{[0:k-1]}\|) + d_2 + (1-\varrho)d_1). \tag{2.23}$$

Furthermore, for $k \in \{L, \dots, 2L-1\}$ inequality (2.22) and $L \geq 1$ implies that

$$\mathcal{V}(x_k) \leq \varrho\mathcal{V}(x_{k-L}) + L(\sigma(\|\omega_{[0:k-1]}\|) + d_2 + (1-\varrho)d_1).$$

Using that $k - L \in \{0, \dots, L-1\}$ and substituting $\mathcal{V}(x_{k-L})$ from (2.23) results in

$$\mathcal{V}(x_k) \leq \varrho\mathcal{V}(x_0) + (\varrho(k-L) + L)(\sigma(\|\omega_{[0:k-1]}\|) + d_2 + (1-\varrho)d_1).$$

Considering any interval $k \in \{a_L L, \dots, (a_L+1)L-1\}$ induction over $a_L \in \mathbb{N}_0$ gives:

$$\mathcal{V}(x_k) \leq \varrho^{a_L}\mathcal{V}(x_0) + (L + \varrho L + \ldots + \varrho^{a_L}(k - a_L L))(\sigma(\|\omega_{[0:k-1]}\|) + d_2 + (1-\varrho)d_1).$$

It follows for $k = a_L L$:

$$\mathcal{V}(x_k) \leq \varrho^{a_L}\mathcal{V}(x_0) + (L + \varrho L + \ldots + \varrho^{a_L-1}L)(\sigma(\|\omega_{[0:k-1]}\|) + d_2 + (1-\varrho)d_1),$$

and for all $k \neq a_L L$:

$$\mathcal{V}(x_k) \leq \varrho^{a_L}\mathcal{V}(x_0) + (L + \varrho L + \ldots + \varrho^{a_L}L)(\sigma(\|\omega_{[0:k-1]}\|) + d_2 + (1-\varrho)d_1).$$

This bound can be rewritten as follows:

$$\mathcal{V}(x_k) \leq \varrho^{\lfloor \frac{k}{L} \rfloor}\mathcal{V}(x_0) + \left(\sum_{l=0}^{\lfloor \frac{k-1}{L} \rfloor} \varrho^l\right) L(\sigma(\|\omega_{[0:k]}\|) + d_2 + (1-\varrho)d_1),$$

and the empty sum $\sum_{l=0}^{-1} \varrho^l$ is defined to be 0. Using the upper bound on the Lyapunov function from (2.13) it holds that

$$\mathcal{V}(x_k) \leq \varrho^{\lfloor \frac{k}{L} \rfloor}(\alpha_2(\|x_0\|_\Sigma) + d_1) + \left(\sum_{l=0}^{\lfloor \frac{k-1}{L} \rfloor} \varrho^l\right) L(\sigma(\|\omega_{[0:k-1]}\|) + d_2 + (1-\varrho)d_1).$$

Furthermore, considering $\varrho \in (0,1)$ it holds that $\varrho^{\lfloor \frac{k}{L} \rfloor} - \varrho^{\lfloor \frac{k-1}{L} \rfloor + 1} + L \leq \bar{L}$, where

$$\bar{L} := \begin{cases} 1 & \text{if } L = 1, \\ L+1 & \text{if } L > 1. \end{cases}$$

In combination with the identity $\sum_{l=0}^{\lfloor \frac{k-1}{L} \rfloor} \varrho^l = (1 - \varrho^{\lfloor \frac{k-1}{L} \rfloor + 1})/(1-\varrho) \leq 1/(1-\varrho)$ this results in

$$\mathcal{V}(x_k) \leq \varrho^{\lfloor \frac{k}{L} \rfloor}\alpha_2(\|x_0\|_\Sigma) + \bar{L}d_1 + \frac{L}{1-\varrho}(\sigma(\|\omega_{[0:k-1]}\|) + d_2),$$

for all $x_k \in \mathcal{X}$, $\omega_l \in \mathbb{W}$,$l \in \{k, \ldots, k+L-1\}$, $k \geq 0$, and $L \geq 1$. The remainder of the proof again follows the arguments made in [68]. In particular, the bound $\alpha_1(\|x_k\|_\Sigma) \leq \mathcal{V}(x_k)$ can be used to establish that

$$\|x_k\|_\Sigma \leq \alpha_1^{-1}\left(\varrho^{\lfloor \frac{k}{L} \rfloor}\alpha_2(\|x_0\|_\Sigma) + \bar{L}d_1 + \frac{L}{1-\varrho}(\sigma(\|\omega_{[0:k-1]}\|) + d_2)\right),$$

and applying the inequality $\alpha_1^{-1}(s_1 + s_2 + s_3) \leq \alpha_1^{-1}(3s_1) + \alpha_1^{-1}(3s_2) + \alpha_1^{-1}(3s_3)$ results in

$$\|x_k\|_\Sigma \leq \alpha_1^{-1}\left(3\varrho^{\lfloor \frac{k}{L} \rfloor}\alpha_2(\|x_0\|_\Sigma)\right) + \alpha_1^{-1}\left(3\frac{L}{1-\varrho}\sigma(\|\omega_{[0:k-1]}\|)\right) + \alpha_1^{-1}(3\zeta),$$

where $\zeta = \bar{L}d_1 + \frac{L}{1-\varrho}d_2$. The theorem follows by noting that $\beta_c(s,k) \in \mathscr{KL}$ because it is decreasing in k, $\lim_{k\to\infty} \varrho^{\lfloor \frac{k}{L} \rfloor} = 0$, $\alpha_2(s) \in \mathscr{K}_\infty$, and $\alpha_1^{-1}(s) \in \mathscr{K}_\infty$. Furthermore $1/(1-\varrho) > 0$ implies that $d_c \geq 0$, and $\gamma_c(s) \in \mathscr{K}$ because $\sigma(s) \in \mathscr{K}$ □

It should be noted that this result does not require any assumptions on the continuity of either $f_d(x_k, \omega_k)$ or $\mathcal{V}(x_k)$. Furthermore, letting $L = 1$ and $\Sigma = \{0\}$ the results from [68] are recovered. Besides formulating the result with respect to a set Σ instead of the origin, the main difference to [68] is that Theorem 2.1 does not require the condition (2.15) to hold for $L = 1$. Among other cases, this relaxation is useful when analyzing the stability properties of suboptimal model predictive control policies. Also, this result can be seen as extension of the ideas used to prove LaSalle's invariance principle (cf. [66] for the discrete-time case) to robust control problems. In particular, it can be seen that the Lyapunov function is not required to be strictly decreasing over time, even if $\omega_k = 0, d_2 = 0$, as long as it decreases between k and $k+L$. This is conceptually similar to the arguments used by LaSalle, namely that an undisturbed discrete-time system with continuous dynamics $f_d(x_k, 0)$ and Lyapunov function $\mathcal{V}(x_k)$, with $\mathcal{V}(x_{k+1}) \leq \mathcal{V}(x_k)$, is asymptotically stable with respect to the largest set on which $\mathcal{V}(x_k)$ remains constant for all time. For example, one may consider the case where E denotes the set of states x_k for which $\mathcal{V}(x_k)$ is constant, and the largest invariant set contained in E is the origin. In this case, starting anywhere in $E \setminus \{0\}$ the state will leave the set $E \setminus \{0\}$ in finite time and the Lyapunov function will be strictly decreasing at this time. The result in Theorem 2.1 can be interpreted as extension of this argument to robust control and the case of discontinuous dynamics, such as the piecewise affine systems discussed in Chapter 7.

Stability of Distributed Model Predictive Control

In nominal centralized MPC (i.e. $\omega_k = 0$ for all k) the optimal finite horizon cost $V^*(x_k)$ given by (2.6) is typically used as Lyapunov function, and under suitable assumptions the conditions in Theorem 2.1 hold for $V^*(x_k)$ with $d_1 = d_2 = 0$, and $L = 1$. These assumptions are, that there exists an input sequence such that the dynamics remain inside the constraints and can be steered to $x = 0$, that $(x; u) = (0; 0)$ is an equilibrium of the dynamics, and that the costs are positive definite with respect to the equilibrium (i.e. $Q \succ 0$, $R \succ 0$). In this case optimality implies $V^*(0) = 0$ and (2.13) holds with $d_1 = 0$. In addition (2.22) holds with $d_2 = 0$ if either a suitable terminal constraint is used, or the prediction horizon N is chosen large enough [49].

However, if (2.6) is not solved to optimality the conditions generally only hold for $d_1 > 0$, i.e. only practical stability can be established. In the field of suboptimal model predictive control this problem is usually resolved by employing a terminal constraint and using a terminal controller to reinitialize the input sequence of the MPC once the state reaches a subset of the terminal constraint [87]. In this case, the upper bound in (2.13) holds with $d_1 = 0$. Because a distributed MPC algorithm may not converge to the global optimum, or may have to be terminated before reaching the global optimum due to time or communication constraints distributed model predictive control is often inherently suboptimal. However, the approach from [87] is not desirable in the distributed setting, because a suitable terminal constraint and terminal control law may involve all subsystems and introduce further strong interconnections into the problem. Furthermore, the reinitialization procedure requires additional global communication. Another approach to deal with suboptimality is to utilize a terminal constraint and an extended state vector $(x_k, \mathbf{u}_k)$ instead of the state vector x_k. In this case, the bounds in (2.13) hold with $d_1 = 0$ but (2.15) does not hold with $L = 1$ and $d_2 = 0$. In [104] an additional "stability" constraint is added to the distributed problem in order to ensure that (2.15) holds with $L = 1$ and $d_2 = 0$. However, this additional constraint complicates the problem and may be problematic if disturbances are present (i.e. if $\omega_k \neq 0$). In contrast, a terminal constraint and an extended state vector can be used to establish that the conditions in Theorem 2.1 hold for suboptimal solutions with $L = N$ and $d_1 = d_2 = 0$. Therefore, both asymptotic stability and robust stability can be established without the additional "stability" constraints.

2.4. Some Basics of Mathematical Optimization

Throughout this thesis model predictive controllers, which aim to solve an optimization problem of the type shown in (2.6) in a distributed fashion, are considered. For the case of linear subsystems, quadratic costs, and convex constraints the resulting optimization problems are convex. If the dynamics are piecewise affine the MPC problem becomes non-convex, namely a mixed-integer quadratic program. This section briefly reviews important results on mathematical optimization.

Convex Sets and Functions

Definition 2.4. *(cf. [10], [15])*

(a) A subset $\mathcal{S}$ of $\mathbb{R}^{n_y}$ is convex if it includes for every pair of points the line segment that joins them, or in other words, if

$$(1-\theta)y_1 + \theta y_2 \in \mathcal{S}, \quad \forall y_1, y_2 \in \mathcal{S}, \forall \theta \in [0,1].$$

(b) A function $f_c : \mathcal{S} \to \mathbb{R}$ on a convex set $\mathcal{S}$ is convex relative to $\mathcal{S}$ if

$$f_c(\theta y_1 + (1-\theta)y_2) \leq \theta f_c(y_1) + (1-\theta)f(y_2), \quad \forall y_1, y_2 \in \mathcal{S}, \forall \theta \in [0,1],$$

and the function f_c is strictly convex if the above inequality is strict for all $y_1, y_2 \in \mathcal{S}$ with $y_1 \neq y_2$, and all $\theta \in (0,1)$.

(c) A differentiable function $f_c : \mathcal{S} \to \mathbb{R}$ on a convex set $\mathcal{S}$ is strongly convex relative to $\mathcal{S}$ if there exists $m \in \mathbb{R}_{>0}$ such that

$$f_c(y_2) \geq f_c(y_1) + \nabla f_c(y_1)^T (y_2 - y_1) + \frac{m}{2}\|y_2 - y_1\|^2, \quad \forall y_1, y_2 \in \mathcal{S}.$$

It is strictly convex if the above inequality is strict for $y_1 \neq y_2$ and $m = 0$ (i.e. strong convexity implies strict convexity), and convex if the above inequality holds for $m = 0$.

Convex Optimization

The problem of finding an $y \in \mathcal{S}$ which minimizes a convex function $f_{\text{obj}} : \mathbb{R}^{n_y} \to \mathbb{R}$ over a set $\mathcal{S} \subseteq \mathbb{R}^{n_y}$ and the corresponding optimal value f_{obj}^* are described by

$$\min_{y \in \mathcal{S}} \quad f_{\text{obj}}(y), \qquad f_{\text{obj}}^* = \inf_{y \in \mathcal{S}} f_{\text{obj}}(y), \tag{2.24}$$

where inf denotes the infimum, i.e. the greatest lower bound on $f_{\text{obj}}(y)$ with $y \in \mathcal{S}$. Likewise, in the case of maximization the optimization problem and optimal value are denoted by

$$\max_{y \in \mathcal{S}} \quad f_{\text{obj}}(y), \qquad f_{\text{obj}}^* = \sup_{y \in \mathcal{S}} f_{\text{obj}}(y), \tag{2.25}$$

where sup denotes the supremum, i.e. the least upper bound on $f_{\text{obj}}(y)$ with $y \in \mathcal{S}$. If there exists $y^* \in \mathcal{S}$ such that $f_{\text{obj}}(y^*) = f_{\text{obj}}^*$ then f_{obj} attains its minimum / maximum over $\mathcal{S}$ at y^*, where y^* denotes the optimal solution. The following version of the Weierstrass extreme value theorem gives conditions under which the minimum is attained.

Theorem 2.2. *(cf. [10], Proposition A.8) Let $\mathcal{S}$ be a nonempty subset of $\mathbb{R}^{n_y}$ and let $f_{obj} : \mathcal{S} \to \mathbb{R}$ be lower semicontinuous at all points of $\mathcal{S}$. Assume that one of the following conditions holds:*

(a) $\mathcal{S}$ *is compact,*

(b) $\mathcal{S}$ *is closed and* f_{obj} *is coercive (i.e.* $f_{obj}(y) \to \infty$ *as* $\|y\| \to \infty$*),*

(c) There exists a scalar c_{Lev} *such that the level set* $\{y \in \mathcal{S} | f_{obj}(y) \leq c_{Lev}\}$ *is nonempty and compact,*

(d) $f_{obj}(y)$ *is strongly convex.*

Then, there exists $y^* \in \mathcal{S}$ *such that* $f_{obj}(y^*) = \inf_{y \in \mathcal{S}} f_{obj}(y)$*.*

A proof for (a) to (c) can be found in [10]. To establish (d) note that if $f_{\text{obj}}(y)$ is strongly convex it is also continuous and all its level sets are bounded by closed balls (see [15], Section 9.1.2). It directly follows that (d) implies (c). In the following, min will be used instead of inf if the minimum is attained, and max instead of sup if the maximum is attained. If $\mathcal{S} = \emptyset$ the optimization problem is said to be infeasible. In this case f^*_{obj} takes values in $\bar{\mathbb{R}} := [-\infty, \infty]$. In particular, $\inf_{y \in \emptyset} f_{\text{obj}}(y) = \infty$ and $\sup_{y \in \emptyset} f_{\text{obj}}(y) = -\infty$ (cf. [93]). The following Proposition from [10] is concerned with first-order optimality conditions:

Theorem 2.3. *(cf. [15],[10]) Let* $\mathcal{S}$ *be a nonempty convex subset of* $\mathbb{R}^{n_y}$*. If* f_{obj} *is convex and differentiable then* $y^* \in \mathcal{S}$ *is optimal if and only if*

$$\nabla f_{obj}(y^*)^T (y - y^*) \geq 0, \quad \forall y \in \mathcal{S},$$

and every local minimum is a global minimum. If in addition f_{obj} *is strictly convex, then the optimizer* y^* *is unique.*

A proof can be found in [15], Section 4.2.3, and the uniqueness of the optimizer follows from [10], Proposition B.10.

The set $\mathcal{S}$ may be described by h_i inequality constraints $f_{\text{ineq},i}(y) \leq 0$, $i \in \{1, \ldots, h_i\}$, and h_e equality constraints $f_{\text{eq},i}(y) = 0$, $i \in \{1, \ldots, h_e\}$:

$$\mathcal{S} = \left\{ y \in \mathbb{R}^{n_y} \;\middle|\; \begin{array}{ll} f_{\text{ineq},i}(y) \leq 0, & \forall i \in \{1, \ldots, h_i\}, \\ f_{\text{eq},i}(y) = 0, & \forall i \in \{1, \ldots, h_e\} \end{array} \right\},$$

where $f_{\text{ineq},i} : \mathbb{R}^{n_y} \to \mathbb{R}$ and $f_{\text{eq},i} : \mathbb{R}^{n_y} \to \mathbb{R}$. Then, the optimization problem (2.24) is equivalently described by

$$\min_{y} \quad f_{\text{obj}}(y) \tag{2.26}$$

$$\text{s.t.} \quad f_{\text{ineq},i}(y) \leq 0, \quad \forall i \in \{1, \ldots, h_i\}, \tag{2.27}$$

$$f_{\text{eq},i}(y) = 0, \quad \forall i \in \{1, \ldots, h_e\}, \tag{2.28}$$

and (2.25) can be rewritten in similar fashion.

Duality

The Lagrangian $\mathcal{L} : \mathbb{R}^{n_y} \times \mathbb{R}^{h_i} \times \mathbb{R}^{h_e} \to \mathbb{R}$ of problem (2.24) is given by

$$\mathcal{L}(y, \lambda, \nu) = f_{\text{obj}}(y) + \sum_{i=1}^{h_i} \lambda_i f_{\text{ineq},i}(y) + \sum_{i=1}^{h_e} \nu_i f_{\text{eq},i}(y), \tag{2.29}$$

and $\lambda = [\lambda_1, \dots, \lambda_{h_i}]^T$, $\nu = [\nu_1, \dots, \nu_{h_e}]^T$ are the Lagrange multipliers associated with inequality and equality constraints. Based on the Lagrangian the Lagrange dual function

$$g_d(\lambda, \nu) := \inf_{y \in C} \mathcal{L}(y, \lambda, \nu)$$

is defined, which has the property that $g_d(\lambda, \nu) \le f^*_{\text{obj}}$ holds for all $\lambda \ge 0$, all ν, and the optimal value f^*_{obj} of (2.24) (cf. [15]). Furthermore, the so called Lagrange dual problem is given by

$$g^*_d := \max_{\lambda \ge 0} g_d(\lambda, \nu), \tag{2.30}$$

with optimal value g^*_d and optimal multipliers (λ^*, ν^*). If the primal problem (2.24) is convex and a so called constraint qualification holds, then strong duality holds. This implies $g^*_d = f^*_{\text{obj}}$. The simplest constraint qualification is Slater's condition, which requires that a strictly feasible point, i.e. $y \in \text{int}(\mathcal{S})$, exists. Another type of constraint qualification is Linear Independence Constraint Qualification (LICQ), which is satisfied if the gradients of the active constraints, evaluated at the primal optimizer y^*, are independent (cf. [88]).

KKT optimality conditions

Assuming that a constraint qualification holds, there exist primal and dual points y^*, (λ^*, ν^*) such that Karush-Kuhn-Tucker (KKT) conditions for optimality hold. The KKT conditions are given by (cf. [15])

$$f_{\text{ineq},i}(y^*) \le 0, \ \ \forall i \in \{1, \dots, h_i\} \tag{2.31}$$
$$f_{\text{eq},i}(y^*) = 0, \ \ \forall i \in \{1, \dots, h_e\} \tag{2.32}$$
$$\lambda^*_i \ge 0, \ \ \forall i \in \{1, \dots, h_i\} \tag{2.33}$$
$$\lambda^*_i f_{\text{ineq},i}(y^*) = 0, \ \ \forall i \in \{1, \dots, h_i\} \tag{2.34}$$
$$\nabla f_{\text{obj}}(y^*) + \sum_{i=1}^{h_i} \lambda^*_i \nabla f_{\text{ineq},i}(y^*) + \sum_{i=1}^{h_e} \nu^*_i \nabla f_{\text{eq},i}(y^*) = 0. \tag{2.35}$$

The first two conditions imply primal feasibility, the third condition dual feasibility and the last two conditions imply optimality. When the problem is convex, and the equality constraints are affine, the KKT conditions are both necessary and sufficient for optimality (cf. [15], Section 5.5.3).

Mixed-Integer Programming

In the case of hybrid dynamics, which involve discrete states and inputs, the MPC problem (2.6) becomes a mixed-integer program, which involves continuous variables as well as variables which are restricted to the set of integers $\mathbb{Z}$. The optimization problem then becomes:

$$\min_{y} \quad f_{\text{obj}}(y) \tag{2.36}$$

$$\text{s.t.} \quad f_{\text{ineq},i}(y) \leq 0, \quad \forall i \in \{1, \ldots, h_i\}, \tag{2.37}$$

$$f_{\text{eq},i}(y) = 0, \quad \forall i \in \{1, \ldots, h_e\}, \tag{2.38}$$

$$y_i \in \mathbb{Z}, \quad \forall i \in \mathscr{I}, \tag{2.39}$$

where $\mathscr{I}$ is the index set of optimization variables restricted to $\mathbb{Z}$. The optimization problem (2.36) is called a convex mixed-integer nonlinear program (convex MINLP) if it becomes convex when the constraint (2.39) is relaxed to a convex constraint. If the costs are linear and the constraints are affine, it is called a mixed-integer linear program (MILP, cf. [97]). Convex MINLP and MILP may have multiple global optima and can be solved by algorithms such as branch and bound and cutting plane approaches which iteratively solve convex or linear approximations of the convex MINLP and are guaranteed to find a global optimum (cf. [70]). However, in the worst case the computation time of these algorithms grows combinatorial with the number of integer variables. In the case of piecewise affine dynamics (2.3), polytopic inequality constraints, affine equality constraints, and a quadratic cost function (2.5) the MPC problem (2.6) becomes a mixed-integer quadratic program. In this case, if the integer variables y_i with $i \in \mathscr{I}$ are either fixed or relaxed to the set $y_i \in [0, 1]$ a quadratic program is obtained. Thus, this problem belongs to the class of convex MINLP. Properties of convex MINLP and solution algorithms for this class of problems can be found in, e.g., [70] and [69]. In [62] fast numerical methods for the (centralized) solution of model predictive control problems involving integer variables are investigated, however only discrete inputs are considered.

Distributed Optimization

In general most distributed optimization methods of interest in the context of distributed model predictive control fall into the category of either primal or dual decomposition. The main idea of these methods is to decompose a large optimization problem into several small problems which can be solved in parallel. This concept is especially useful with respect to distributed MPC because the computations can be distributed across communicating local controllers, spreading the computational load across controllers of different subsystems. The following optimization problem:

$$f^*_{\text{obj}} = \min_{y_1 \in \mathcal{S}_1, y_2 \in \mathcal{S}_2} \quad f_{\text{obj},1}(y_1) + f_{\text{obj},2}(y_2) \tag{2.40}$$

$$\text{s.t.} \quad f_{\text{ineq},1}(y_1) + f_{\text{ineq},2}(y_2) \leq 0, \tag{2.41}$$

consists of two convex subproblems, local variables $y_1 \in \mathbb{R}^{n_{y_1}}$, $y_2 \in \mathbb{R}^{n_{y_2}}$ and convex coupled constraints. While many different formulations are used in the literature, most distributed MPC algorithms using dual decomposition rely on this type of formulation. Therefore it is used here to provide a concise introduction to decomposition methods.

In primal decomposition, the coupling constraint is split by introducing a variable $y_t \in \mathbb{R}^{n_p}$, such that the following two subproblems are obtained:

$$\begin{aligned} f_{\text{obj},1}(y_t) &= \inf_{y_1 \in \mathcal{S}_1} f_{\text{obj},1}(y_1), \text{ s.t. } f_{\text{ineq},1}(y_1) \leq y_t \\ f_{\text{obj},2}(y_t) &= \inf_{y_2 \in \mathcal{S}_2} f_{\text{obj},2}(y_2), \text{ s.t. } f_{\text{ineq},2}(y_2) \leq -y_t. \end{aligned}$$

Next, a so called master problem is introduced which minimizes the cost over y_t and is equivalent to the original problem (2.40):

$$f_{\text{obj}}(y_t) = \min_{y_t} \; f_{\text{obj},1}(y_t) + f_{\text{obj},2}(y_t).$$

This problem is solved iteratively by solving $f^*_{\text{obj},1}(y_t)$ and $f^*_{\text{obj},2}(y_t)$ for a given y_t, updating the variable y_t by a method with low computational complexity (e.g. subgradient methods) and repeating the process until a stopping condition holds. Given a feasible initialization for y_t the cost $f_{\text{obj}}(y_t)$ upper bounds f^*_{obj} at any iteration and converges to f^*_{obj} if a suitable method is used to update y_t. Since the update of y_t is computationally cheap large savings in computation time may result due to the parallel optimization of the subproblems.

In dual decomposition the coupling constraint is dualized by forming a partial Lagrangian (2.29) with respect to the coupled constraints:

$$\mathcal{L}(y_1, y_2, \lambda) = f_{\text{obj},1}(y_1) + f_{\text{obj},2}(y_2) + \lambda^T (f_{\text{ineq},1}(y_1) + f_{\text{ineq},2}(y_2)).$$

This results in the following two subproblems for given dual variables λ:

$$\begin{aligned} g_{d,1}(\lambda) &= \inf_{y_1 \in \mathcal{S}_1} f_{\text{obj},1}(y_1) + \lambda^T f_{\text{ineq},1}(y_1) \\ g_{d,2}(\lambda) &= \inf_{y_2 \in \mathcal{S}_2} f_{\text{obj},2}(y_2) + \lambda^T f_{\text{ineq},2}(y_2), \end{aligned}$$

and the master problem is given by maximizing the dual function of the original problem (2.40):

$$g_d(\lambda) = \max_{\lambda \geq 0} g_{d,1}(\lambda) + g_{d,2}(\lambda).$$

This problem is again solved iteratively by updating λ with a suitable method (e.g. a subgradient step). Then $g_d(\lambda)$ lower bounds f^*_{obj} in every iteration and converges to f^*_{obj} over the iterations. For more details on subgradient methods used to update y_t or the dual variables λ in such a scheme see e.g. [35] and the references therein.

While the framework of primal and dual decomposition in combination with subgradient methods offers a powerful tool for distributed convex optimization, the convergence of such schemes is often slow and requires many iterations. Furthermore, using dual decomposition in combination with subgradient methods may not yield a feasible solution in a finite number of iterations and the primal cost $f_{\text{obj},1}(y_1) + f_{\text{obj},2}(y_2)$ may increase from one iteration to the next. This is problematic if the algorithm has to be terminated early due to computation or communication constraints. In particular, because the dual function only provides a lower bound on the cost of the original problem convergence rates for the dual function do not necessarily have any implication for convergence of the primal cost. Thus, if a distributed algorithm based on dual decomposition and the subgradient method is terminated early it is in general not possible to give bounds on the suboptimality of the resulting primal solution (cf. [63]).

Different methods have been proposed in the literature to address these shortcomings. In [37] a duality based distributed MPC algorithm and distributed stopping criterion based on the relaxed dynamic programming inequality are presented, which guarantee bounded suboptimality. However, this distributed stopping criterion may not hold at the optimum, in which case a centralized stopping criterion is used. A dual algorithm which generates a feasible solution in a finite number of iterations was proposed in [23] using primal updates and a constraint tightening approach. In [36] an accelerated gradient method with significantly improved convergence rate compared to previous dual decomposition based methods is proposed. Recently the alternating direction method of multipliers (ADMM) first proposed in [16] has gained interest in the field of duality based distributed model predictive control [31] to obtain stronger guarantees if the distributed optimization has to be terminated before convergence. Furthermore, other approaches such as distributed interior point methods [85] and distributed simplex methods [18] have been investigated for special classes of distributed optimization problems arising in control. In [82] Nesterov's first order scheme and a proximal center algorithm are used in combination with event-based communication to solve convex optimization problems with either separable cost or separable constraints. While this method reduces the communication requirements, the number of iterations and communication events is still far too large to be applied to distributed MPC problems. A second order method for solving quadratic programs in a distributed fashion is proposed in [63]. While the results are rather promising and show a significant speed-up compared to gradient based methods, no proof of convergence is available. Furthermore, in many cases the communication requirements may again be prohibitive for distributed MPC. Overall, only few works on distributed optimization algorithms consider closed-loop properties.

Finally, most distributed optimization algorithms are ill-suited for the discontinuity and non-convexity of mixed-integer programs arising from hybrid dynamics. In fact, there are only very few results on distributed MPC and optimization of hybrid or piecewise affine systems, such as [14] where dual decomposition and Lagrangian relaxation are applied to a distributed MPC problem arising from discrete-time linear system with discrete inputs.

Also, none of the methods discussed above can be directly applied to the distributed solution of optimal control problems which involve nonlinear continuous-time dynamics. A possible solution in this case was proposed in [64], where the solution of the continuous-time dynamics is distributed among all nodes and a sequence of convex problems is constructed in order to approximate the original optimal control problem. These convex problems may then be solved by the algorithms discussed above.

Part II.

Distributed MPC with Time-Triggered Communication

3. Distributed Model Predictive Control Based on Robust Optimization

This chapter deals with distributed model predictive control (distributed MPC) for interconnected linear time invariant systems with decoupled dynamics and coupling via a common cost function as well as common convex state constraints. Such problems commonly arise in formation control problems such as vehicle platooning or control of satellite formations. For instance, the control task for a platoon of vehicles in a leader-follower scenario is to follow the lead vehicle with a constant spacing while avoiding collisions.

Similar control tasks have been considered in [83] in the framework of a non-iterative sequential distributed MPC algorithm without communication delays, and in [33] where constant communication delays and parallel computation is considered for subsystems only interconnected by costs. The aim of this chapter is to develop a non-iterative distributed MPC algorithm with parallel optimization which can be used for subsystems interconnected by constraints, as well as control tasks such as the synchronization of subsystem trajectories, and offers robustness against bounded time-varying communication delays.

This chapter is mostly based on results previously published in [42], [47] and the book chapter [48]. In the algorithm presented in this chapter, local model predictive controllers optimize the local inputs in parallel. The controllers exchange state measurements and predicted input sequences via possibly delayed communication and consider information received from controllers of interconnected systems in the local optimization problem. However, due the communication delays the communicated information is inherently uncertain. The main idea in [42] is to introduce constraints which ensure that the difference between previously communicated input sequences and actual inputs is bounded and to communicate those bounds. Locally robust model predictive controllers (cf. [39]) are used to ensure robustness with respect to these bounded uncertainties. The local robust MPCs optimize a sequence of nominal inputs as well as feedback policies for the delayed information. This optimization can be implemented as a tractable quadratic program. The resulting distributed MPC algorithm renders the overall system input-to-state practically stable [47]. In this chapter, the stability proof given in [47] is adapted to the framework presented in Theorem 2.3. This provides additional insight into the problem, as well as improved bounds. In particular, in [47] the neighborhood of the control goal that the state is guaranteed to converge to for $k \to \infty$ strongly depends on the initialization of the distributed MPC at $k = 0$. This drawback has been resolved here by applying the results from Theorem 2.3.

3.1. Distributed System Model and Control Objectives

In this chapter, N_s discrete-time linear and time invariant (LTI) systems $\mathcal{P}^i$ which are modeled by the difference equations

$$x^i_{k+1} = A^i x^i_k + B^i u^i_k \tag{3.1}$$

are considered. These subsystems with local input $u^i_k \in \mathbb{U}^i \subseteq \mathbb{R}^{m^i}$ are interconnected by global state constraints $x_k \in \mathbb{X} \subseteq \mathbb{X}^1 \times \ldots \times \mathbb{X}^{N_s} \subseteq \mathbb{R}^n$. In order to obtain a suitable decomposition, it is assumed that each coupled cost term and constraint is assigned to one local controller. In other words, if a constraint involves x^1_k and x^2_k a decomposition is chosen such that either $\mathcal{C}^2$ depends on x^2_k and x^1_k, and $\mathcal{C}^1$ only depends on x^1_k, or $\mathcal{C}^1$ depends on x^1_k and x^2_k, and $\mathcal{C}^2$ only depends on x^2_k. This decomposition allows using neighboring, one way, communication between each pair of controllers, resulting in low communication requirements and an algorithm which is well suited to deal with delayed communication.

The index set $\mathcal{N}^i = \{i_1, \ldots, i_{N^i_s}\} \subseteq \mathcal{N}$ contains the indices of all the subsystems with which $\mathcal{P}^i$ is coupled through costs or constraints $\mathbb{X}$. In other words, if $(j, i) \in \mathcal{I}$ then $\mathcal{C}^i$ depends on the states of $\mathcal{P}^j$ or inputs computed by $\mathcal{C}^j$. The communication graph for the non-iterative algorithm proposed in this chapter is time-invariant and given by $\mathcal{E}_{k,0} := \mathcal{I}$ for all k.

Assumption 3.1. *It is assumed that the interconnection graph $\mathcal{G}$ does not contain any cycles.*

For example, consider the subsystems shown in Figure 3.1. Without the assumption that there are no cycles in the graph $\mathcal{G}$, uncertainties could be amplified while propagating through the distributed controllers $\mathcal{C}^i$ and could affect the controller they originated from. Without this assumption some properties used in this chapter may not hold.

In order to consider the coupled costs and constraints, the following augmented prediction model for the subsystem $\mathcal{P}^i$ is defined:

$$\tilde{x}^i_{k+l+1|k} = \tilde{A}^i \tilde{x}^i_{k+l|k} + \tilde{B}^i \tilde{u}^i_{k+l|k}, \tag{3.2}$$

where $\tilde{A}^i = \text{blkdiag}(A^i, A^{i_1}, \ldots, A^{i_{N^i_s}})$, $\tilde{B}^i = \text{blkdiag}(A^i, B^{i_1}, \ldots, B^{i_{N^i_s}})$, and the augmented state and input vector are given by $\tilde{x}^i_{k+l|k} := (x^i_{k+l|k}; x^{i_1}_{k+l|k}; \ldots; x^{i_{N^i_s}}_{k+l|k}) \in$

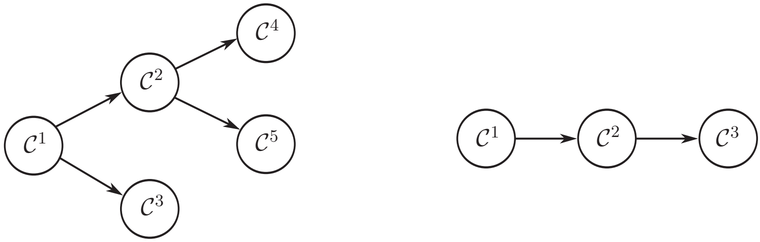

Figure 3.1.: Interconnection graphs $\mathcal{G}$ which satisfy Assumption 3.1.

$\mathbb{R}^{\tilde{n}^i}$ and $\tilde{u}^i_{k+l|k} := (u^i_{k+l|k}; u^{i_1}_{k+l|k}; \ldots; u^{i_{N^i_s}}_{k+l|k}) \in \mathbb{R}^{\tilde{m}^i}$, with $\tilde{n}^i = n^i + \sum_{j \in \mathcal{N}^i} n^j$, and $\tilde{m}^i = m^i + \sum_{j \in \mathcal{N}^i} m^j$. It should be noted that (3.2) does not induce coupling in the dynamics. Instead, the purpose of introducing states of interconnected subsystems into a local model is to explicitly consider the coupled constraints and the coupled costs. Using this augmented model, corresponding constraints $\tilde{x}^i_{k|k} \in \tilde{\mathbb{X}}^i$ can be defined, which include the coupled constraints. Furthermore, the set Σ, defined in (2.4), can be decomposed into sets

$$\Sigma^i := \left\{ \tilde{z}^i \in \mathbb{R}^{\tilde{n}^i} \times \mathbb{R}^{\tilde{m}^i} \,\middle|\, \tilde{z}^i = (\tilde{x}^i; \tilde{u}^i), \Gamma^i_x \tilde{x}^i = 0, \Gamma^i_u \tilde{u}^i = 0 \right\}, \tag{3.3}$$

which depend on extended local state vectors $\tilde{z}^i_k$ and collect the equalities of Σ corresponding to the decomposition given by $\mathcal{N}^i$. A local cost function can be formulated which is positive definite with respect to Σ^i:

$$\mathbf{V}^i(\tilde{\mathbf{x}}^i_k, \tilde{\mathbf{u}}^i_k) = \|\tilde{x}^i_{\tau,k+N|k}\|^2_{\tilde{P}^i} + \sum_{l=0}^{N-1} \|\tilde{x}^i_{k+l|k}\|^2_{\tilde{Q}^i} + \|\tilde{u}^i_{k+l|k}\|^2_{\tilde{R}^i}, \tag{3.4}$$

where $\tilde{Q}^i := (\Gamma^i_x)^T \Gamma^i_x \succeq 0$ and $\tilde{R}^i = (\Gamma^i_u)^T \Gamma^i_u \succeq 0$ represent the weighting matrices obtained by decomposition of the control objectives given by (2.5). It follows that $\|\tilde{x}^i_k\|^2_{\tilde{Q}^i} + \|\tilde{u}^i_k\|^2_{\tilde{R}^i} = 0$, $\forall \tilde{z}^i_k \in \Sigma^i$, and $\|\tilde{x}^i_k\|^2_{\tilde{Q}^i} + \|\tilde{u}^i_k\|^2_{\tilde{R}^i} > 0$, $\forall \tilde{z}^i_k \notin \Sigma^i$. The term $\|\tilde{x}^i_{\tau,k+N|k}\|^2_{\tilde{P}^i}$ is a terminal cost defined below.

The local cost functions (3.4) and constraints $\tilde{x}^i_{k|k} \in \tilde{\mathbb{X}}^i$ depend on the states and control actions of interconnected subsystems. This introduces some degree of cooperation between the local controllers which locally optimize (3.4). However, the states and planned inputs of interconnected subsystems $\mathcal{P}^j$, $j \in \mathcal{N}^i$ are not exactly known by the controller $\mathcal{C}^i$ due to the communication delays and parallel optimization. With respect to the communication the following assumption is made:

Assumption 3.2. *It is assumed that the communication between controllers is subject to a bounded time-varying communication delay, that the clocks of the controllers are synchronized, and messages include a timestamp. The delay of a message sent by $\mathcal{C}^j$ and received at time k by $\mathcal{C}^i$ is denoted by $\tau^{i,j}_k$ and bounded by*

$1 \leq \tau_k^{i,j} \leq \tau_{\max}$, *where the lower bound follows from the fact that the controllers* $\mathcal{C}^i$ *optimize in parallel. It is further assumed that the prediction horizon of the MPC is chosen such that* $N \gg \tau_{\max}$ *and that* A^j *and* B^j *are known to all controllers* $\mathcal{C}^i$, $j \in \mathcal{N}^j$.

To model the discrepancy between the actual states, inputs, and values obtained through delayed communication, let $\hat{u}^{i,j}_{k+l|k} = u^j_{k+l|k+l} + \delta\hat{u}^{i,j}_{k+l|k} \in \mathbb{R}^{m^j}$ and $\hat{x}^{i,j}_{k+l|k} = x^j_{k+l|k+l} + \delta\hat{x}^{i,j}_{k+l|k} \in \mathbb{R}^{n^j}$ denote uncertain predictions that the controller $\mathcal{C}^i$ has of the states $x^j_{k|k}$ and inputs $u^j_{k|k}$ of a subsystem $\mathcal{P}^j$, $j \in \mathcal{N}^i$ with corresponding uncertainties $\delta\hat{x}^{i,j}_{k|k}$ and $\delta\hat{u}^{i,j}_{k+l|k} \in \Delta\mathbb{U}^{i,j}_{k+l|k}$. The predictions for $\mathcal{P}^j$, $j \in \mathcal{N}^i$ are obtained through communication as follows: At each time step k each local controller $\mathcal{C}^j$ communicates the state $x^j_{k|k}$ and its planned inputs $\hat{u}^i_{k+l|k}$ for all $l \in \{0, \dots, N-1\}$. This information is received by $\mathcal{C}^i$ with a delay of $\tau_k^{i,j}$ time steps, such that e.g. $\hat{x}^{i,j}_{k-\tau_k^{i,j}|k} := x^j_{k-\tau_k^{i,j}}$ and $\hat{u}^{i,j}_{k+l|k} := \hat{u}^{j,j}_{k+l|k-\tau_k^{i,j}}$. Using this information, the following local prediction model is obtained:

$$\hat{\tilde{x}}^i_{k+l+1|k} = \tilde{A}^i \hat{\tilde{x}}^i_{k+l|k} + \tilde{B}^i \hat{\tilde{u}}^i_{k+l|k}, \tag{3.5}$$

where the state, input and corresponding uncertainties for the non-local states and inputs are given by

$$\hat{\tilde{x}}^i_{k+l|k} := \begin{bmatrix} x^i_{k+l|k} \\ \hat{x}^{i,i_1}_{k+l|k} \\ \vdots \\ \hat{x}^{i,i_{N_s^i}}_{k+l|k} \end{bmatrix}, \; \hat{\tilde{u}}^i_{k+l|k} := \begin{bmatrix} u^i_{k+l|k} \\ \hat{u}^{i,i_1}_{k+l|k} \\ \vdots \\ \hat{u}^{i,i_{N_s^i}}_{k+l|k} \end{bmatrix}, \; \delta\hat{\tilde{u}}^i_{k+l|k} := \begin{bmatrix} \delta\hat{u}^{i,i_1}_{k+l|k} \\ \vdots \\ \delta\hat{u}^{i,i_{N_s^i}}_{k+l|k} \end{bmatrix}, \; \delta\hat{\tilde{x}}^i_{k+l|k} := \begin{bmatrix} \delta\hat{x}^{i,i_1}_{k+l|k} \\ \vdots \\ \delta\hat{x}^{i,i_{N_s^i}}_{k+l|k} \end{bmatrix},$$

and $\hat{\tilde{x}}^i_{k|k} \in \tilde{\mathbb{X}}^i \subseteq \mathbb{R}^{\tilde{n}^i}$, $\delta\hat{\tilde{x}}^i_{k|k} \in \mathbb{R}^{\tilde{n}^i - n^i}$, $\hat{\tilde{u}}^i_{k|k} \in \mathbb{R}^{\tilde{m}^i}$, $\delta\hat{\tilde{u}}^i_{k|k} \in \mathbb{R}^{\tilde{m}^i - m^i}$. In the worst case, the last exactly known state at time k may be $\tilde{x}^i_{k-\tau_{\max}|k-\tau_{\max}}$ and the last predictions received are $\hat{u}^{j,j}_{k+l|k-\tau_{\max}}$ with $l \in \{-\tau_{\max}, \dots, N-1-\tau_{\max}\}$ for all $j \in \mathcal{N}^i$, i.e. $\mathcal{C}^i$ can perform a forward prediction starting from $\hat{x}^i_{k-\tau_{\max}|k} = \tilde{x}^i_{k-\tau_{\max}|k-\tau_{\max}}$ using the model (3.5). The error of the prediction of non-local states can also be obtained by forward prediction according to:

$$\delta\hat{x}^{i,j}_{k+l+1|k} = \tilde{A}^i \delta\hat{x}^{i,j}_{k+l|k} + \tilde{B}^i \delta\hat{u}^{i,j}_{k+l|k}, \quad \forall j \in \mathcal{N}^i, \tag{3.6}$$

where $\delta\hat{x}^{i,j}_{k-\tau_{\max}|k} = 0$. It follows that $\delta\hat{\tilde{x}}^i_{k|k}$ can be eliminated from the problem at hand because it is a function of $\delta\hat{\tilde{u}}^{i,j}_{k+l|k}$ with $l \in \{-\tau_{\max}, \dots, 0\}$. Thus, a prediction model which considers the uncertainty of the communicated information is obtained. In the following, $u^i_{k|k}$ will be constrained to a time-varying set $\mathbb{U}^i_{k|k}$ by each controller $\mathcal{C}^i$ to ensure that the uncertainties affecting the interconnected controllers $\mathcal{C}^j$ bounded.

To establish recursive feasibility and practical stability of the distributed MPC algorithm, a terminal control law given by a delayed state feedback of the form

$$\tilde{u}_k^i = \underbrace{\begin{bmatrix} K_{\mathbb{T}}^i \\ 0_{\tilde{m}^i - m^i \times \tilde{n}^i} \end{bmatrix}}_{=:\tilde{K}_{\mathbb{T}}^i} \tilde{x}_{k+1-\tau_{\max}}^i \tag{3.7}$$

is used to account for possible communication delays.

In order to explicitly include such a delay into a linear discrete time system for controller design, the extended state vector $\tilde{x}_{\tau,k}^i = (\tilde{x}_k^i; \dots; \tilde{x}_{k-\tau_{\max}+1}^i) \in \mathbb{R}^{\tau_{\max}\tilde{n}^i}$ is defined (cf. [22]), which leads to the following standard assumptions [81] adapted to the delayed feedback formulation.

Assumption 3.3. *For each system* (3.5)*, there exists a terminal controller* (3.7) *with* $K_{\mathbb{T}}^i \in \mathbb{R}^{m^i \times \tilde{n}^i}$*, a corresponding terminal weight* $\tilde{P}^i = (\tilde{P}^i)^T \succeq 0$ *, and a terminal set* $\tilde{\mathbb{T}}^i \subseteq \tilde{\mathbb{X}}^i \times \ldots \times \tilde{\mathbb{X}}^i$*, such that the following holds for all* $\tilde{x}_{\tau,k}^i \in \tilde{\mathbb{T}}^i$*:*

1. *The terminal controller renders the terminal set robust forward invariant with respect to* $\delta\hat{\tilde{u}}_k^i$*, i.e.* $\tilde{x}_{\tau,k+1}^i \in \tilde{\mathbb{T}}^i$ *holds for all* $\delta\hat{u}_k^{i,j} \in \mathbb{U}^j$ *and:*

$$\tilde{x}_{\tau,k+1}^i = \left[\begin{array}{cc|c} \tilde{A}^i & 0_{(\tau_{\max}-2)\tilde{n}^i \times \tilde{n}^i} & \tilde{B}^i \tilde{K}_{\mathbb{T}}^i \\ \hline \multicolumn{2}{c|}{I_{(\tau_{\max}-1)\tilde{n}^i}} & 0_{(\tau_{\max}-1)\tilde{n}^i \times \tilde{n}^i} \end{array}\right] \tilde{x}_{\tau,k}^i + \begin{bmatrix} \tilde{B}^i \\ 0_{(\tau_{\max}-1)\tilde{n}^i \times \tilde{m}^i} \end{bmatrix} \begin{bmatrix} 0 \\ \delta\hat{\tilde{u}}_k^i \end{bmatrix}. \tag{3.8}$$

2. *The terminal control law satisfies the input constraints:*

$$u_k^i = \begin{bmatrix} 0_{m^i \times (\tau_{\max}-1)\tilde{n}^i} & K_{\mathbb{T}}^i \end{bmatrix} \tilde{x}_{\tau,k}^i \quad \in \mathbb{U}^i.$$

3. *The terminal cost is a Lyapunov function for the undisturbed closed-loop system* (3.8) *in the sense that*

$$\|\tilde{x}_{\tau,k+1}^i\|_{\tilde{P}^i}^2 - \|\tilde{x}_{\tau,k}^i\|_{\tilde{P}^i}^2 \leq -\|\tilde{x}_{\tau,k}^i\|_{\tilde{\mathcal{Q}}^i}^2,$$

for $\delta\hat{\tilde{u}}_k^i = 0$*, and with* $\tilde{\mathcal{Q}}^i = \mathit{blkdiag}\left(\tilde{Q}^i, 0_{\tilde{n}^i \times \tilde{n}^i}, \dots, 0_{\tilde{n}^i \times \tilde{n}^i}, \left[\begin{smallmatrix} K^i \\ 0_{\tilde{m}^i \times \tilde{n}^i} \end{smallmatrix}\right]^T \tilde{R}^i \left[\begin{smallmatrix} K^i \\ 0_{\tilde{m}^i \times \tilde{n}^i} \end{smallmatrix}\right]\right)$,

4. $\tilde{z}_{k+l} \in \Sigma, \forall l \in \{-\tau_{\max}+1, \dots, 0\}$ *implies* $\|\tilde{x}_{\tau,k+1}^i\|_{\tilde{P}^i}^2 = 0$ *for the undisturbed closed-loop system* (3.8)*.*

The terminal set is specified by the polytope

$$\tilde{\mathbb{T}}^i = \{\tilde{x}_{\tau,k}^i \in \mathbb{R}^{\tau_{\max}\tilde{n}^i} | C_{\mathbb{T}}^i \tilde{x}_{\tau,k}^i \leq b_{\mathbb{T}}^i\}, \tag{3.9}$$

with $C_{\mathbb{T}}^i \in \mathbb{R}^{h_{\mathbb{T}}^i \times \tau_{\max}\tilde{n}^i}$, $b_{\mathbb{T}}^i \in \mathbb{R}^{h_{\mathbb{T}}^i}$ and can be computed using standard set-theoretic approaches (c.f. [12]).

Overall, it can be seen that Assumption 3.1 is crucial to the decomposition used in this chapter. In particular, Assumption 3.3 would need to be extended with a small gain condition or similar arguments if Assumption 3.1 does not hold. While extensions in this direction may be possible along the lines of the arguments made in [92] they are not essential to the conclusions drawn in this section. Furthermore, it is well known that results based on small gain techniques may be very conservative. Instead, if the coupling graph contains cycles the cooperative methods proposed in subsequent chapters may be preferable.

3.2. Robust Optimization for Distributed MPC

The distributed model presented in the previous section is subject to uncertainties $\delta\hat{u}^i_{k+l|k}$, hence the local MPC has to compute a sequence of future control inputs, such that the state constraints are maintained for all possible future realizations of $\delta\hat{u}^i_{k+l|k}$. However, computing an open-loop control sequence which robustly satisfies the constraints may be very conservative or impossible even for relatively small uncertainties.

Considering Assumption 3.2, the actual input u^j_k of a subsystem $j \in \mathcal{N}^i$ will be known to the controller $\mathcal{C}^i$ with a delay of $\tau_{\max}$ time steps. The uncertainties in (3.5) can be computed at time $k + l + \tau_{\max}$ based on $\delta\hat{u}^{i,j}_{k+l|k} = \hat{u}^{i,j}_{k+l|k} - u^j_{k+l|k+l}$. Hence, a delayed affine feedback

$$u^i_{k+l|k} = \hat{u}^i_{k+l|k} + \sum\nolimits_{r=1-\tau_{\max}}^{l-\tau_{\max}} K^i_{l,r|k}\,\delta\hat{\hat{u}}^i_{k+r|k}, \quad \forall l \in \{0, \ldots, N-1\}, \tag{3.10}$$

of the uncertainties $\delta\hat{\hat{u}}^i_{k+r|k}$ can be applied to the distributed control problem and the uncertainty in $u^i_{k+l|k}$ only arises from the feedback of values which are not known at time k. It should be noted that within the distributed MPC algorithm this feedback is optimized at time k for future times $k+l$ with $l \in \{1, \ldots, N-1\}$. Because the actual communication delays at $k + l$ are not known at time k the worst-case delay $\tau_{\max}$ has to be used in (3.10) and the optimization.

Remark 3.1. *A similar approach is used in robust optimization: For instance in [7], an approach to uncertain linear programming is presented where a part of the solution, referred to as "adjustable variables", are parametrized in affine form in the uncertain parameters. In robust MPC for linear systems $x_{k+1} = Ax_k + Bu_k + \omega_k$ with additive disturbance ω_k similar control laws which are affine in the past disturbances have been proposed in [73], [39] and [100], i.e. the predicted control input $u_{k+l|k}$ for $l \in \{0, \ldots, N-1\}$ is an affine function of the disturbances ω_k to ω_{k+l-1}. An efficient method to optimize such a control law under constraints and disturbances bounded by a polytope is presented in [39], where it was shown that the resulting problem is convex and equivalent to an affine state feedback parametrization.*

The feedback (3.10) policy is expressed as follows:

$$\mathbf{u}^i_k = \hat{\mathbf{u}}^i_k + \mathbf{K}^i_k \delta\hat{\hat{\mathbf{u}}}^i_k, \tag{3.11}$$

where bold letters denote vectors which contain values over the prediction horizon N, such that at time k:

$$\hat{\mathbf{u}}_k^i := \begin{bmatrix} \hat{u}_{k|k}^i \\ \vdots \\ \hat{u}_{k+N-1|k}^i \end{bmatrix}, \; \hat{\tilde{\mathbf{x}}}_k^i := \begin{bmatrix} \hat{\tilde{x}}_{k|k}^i \\ \vdots \\ \hat{\tilde{x}}_{k+N|k}^i \end{bmatrix}, \; \hat{\tilde{\mathbf{u}}}_k^i := \begin{bmatrix} \hat{\tilde{u}}_{k|k}^i \\ \vdots \\ \hat{\tilde{u}}_{k+N-1|k}^i \end{bmatrix}, \; \delta\hat{\tilde{\mathbf{u}}}_k^i := \begin{bmatrix} \delta\hat{\tilde{u}}_{k+1-\tau_{\max}|k}^i \\ \vdots \\ \delta\hat{\tilde{u}}_{k+N-1|k}^i \end{bmatrix},$$

and the vector $\hat{\mathbf{u}}_k^i$ can be interpreted as the planned control input of the controller $\mathcal{C}^i$ in the absence of uncertainties. Let $[T_{\mathbf{K}}^i, T_{\mathbf{K}}^{\setminus i}]$ denote a permutation matrix, such that $\hat{\tilde{\mathbf{u}}}_k^i = [T_{\mathbf{K}}^i, T_{\mathbf{K}}^{\setminus i}]\ (\hat{\mathbf{u}}_k^i; \hat{\mathbf{u}}_k^{\setminus i})$, and $\hat{\mathbf{u}}_k^{\setminus i} = (\hat{u}_{k|k}^{\setminus i}; \ldots; \hat{u}_{k+N-1|k}^{\setminus i})$ contains all predictions of non-local inputs $\hat{u}_{k+l|k}^{\setminus i} = (\hat{u}_{k+l|k}^{i,i_1}; \ldots; \hat{u}_{k+l|k}^{i,i_{N_s^i}})$. Using $T_{\mathbf{K}}^i$ the feedback policy (3.11) can be rewritten such that

$$\tilde{\mathbf{u}}_k^i = \hat{\tilde{\mathbf{u}}}_k^i + T_{\mathbf{K}}^i \mathbf{K}_k^i \delta\hat{\tilde{\mathbf{u}}}_k^i, \tag{3.12}$$

Based on this feedback policy an uncertain prediction model over the whole prediction horizon can be formulated:

$$\hat{\tilde{\mathbf{x}}}_k^i = \tilde{\mathbf{A}}^i \hat{\tilde{x}}_{k|k}^i + \tilde{\mathbf{B}}^i (\hat{\tilde{\mathbf{u}}}_k^i + T_{\mathbf{K}}^i \mathbf{K}_k^i \delta\hat{\tilde{\mathbf{u}}}_k^i) + \mathbf{G}^i \delta\hat{\tilde{\mathbf{u}}}_k^i, \tag{3.13}$$

which explicitly considers the bounded uncertainties $\delta\hat{\tilde{\mathbf{u}}}_k^i \in \boldsymbol{\mathcal{D}}_k^i \subseteq \mathbb{R}^{q^i}$, where $q^i = (N+\tau_{max}-1)(\tilde{m}^i - m^i)$, and the matrix $\mathbf{G}^i := \tilde{\mathbf{B}}^i T_{\mathbf{K}}^{\setminus i}$. In other words, the uncertainty $\delta\hat{u}_{k+l|k}^{i,j}$ only affects the local inputs $u_{k+l|k}^i$ through the feedback matrix $\mathbf{K}_k^i$. In the following this model will be used to optimize over local feedback policies which guarantee robust constraint satisfaction for the overall system. The state constraints $\tilde{\mathbf{X}}^i = \tilde{\mathbb{X}}^i \times \ldots \times \tilde{\mathbb{X}}^i \times \tilde{\mathbb{T}}^i$, the time-varying input constraints $\tilde{\mathbf{U}}_k^i \subseteq \tilde{\mathbb{U}}_{k|k}^i \times \ldots \times \tilde{\mathbb{U}}_{k+N-1|k}^i$ with $\tilde{\mathbb{U}}_{k|k}^i := \mathbb{U}_{k|k}^i \times \mathbb{R}^{\tilde{m}^i - m^i}$, and the set $\boldsymbol{\mathcal{D}}_k^i$ over the prediction horizon are expressed by the following inequalities:

$$\tilde{\mathbf{X}}^i = \left\{ \tilde{\mathbf{x}}_k^i \in \mathbb{R}^{\tilde{n}^i (N+1)} \middle| C_{\tilde{\mathbf{X}}^i} \tilde{\mathbf{x}}_k^i \leq b_{\tilde{\mathbf{X}}^i} \right\}, \tag{3.14}$$

$$\tilde{\mathbf{U}}_k^i = \left\{ \tilde{\mathbf{u}}_k^i \in \mathbb{R}^{\tilde{m}^i N} \middle| C_{\tilde{\mathbf{U}}_k^i} \tilde{\mathbf{u}}_k^i \leq b_{\tilde{\mathbf{U}}_k^i} \right\}, \tag{3.15}$$

$$\boldsymbol{\mathcal{D}}_k^i = \left\{ \delta\hat{\tilde{\mathbf{u}}}_k^i \in \mathbb{R}^{q^i \times 1} \middle| C_{\boldsymbol{\mathcal{D}}_k^i} \delta\hat{\tilde{\mathbf{u}}}_k^i \leq b_{\boldsymbol{\mathcal{D}}_k^i} \right\}, \tag{3.16}$$

where $C_{\tilde{\mathbf{U}}_k^i} \in \mathbb{R}^{h_{\mathbf{U}}^i \times \tilde{m}^i N}$, $b_{\tilde{\mathbf{U}}_k^i} \in \mathbb{R}^{h_{\mathbf{U}}^i}$, $C_{\boldsymbol{\mathcal{D}}_k^i} \in \mathbb{R}^{h_{\boldsymbol{\mathcal{D}}}^i \times q^i}$, $b_{\boldsymbol{\mathcal{D}}_k^i} \in \mathbb{R}^{h_{\boldsymbol{\mathcal{D}}}^i}$, and $h_{\boldsymbol{\mathcal{D}}}^i$ is the number of faces of the disturbance polytope of $\mathcal{C}^i$ given by $\boldsymbol{\mathcal{D}}_k^i := \Delta\tilde{\mathbb{U}}_{k+1-\tau_{\max}|k}^i \times \ldots \times \Delta\tilde{\mathbb{U}}_{k+N-1|k}^i$ and $\Delta\tilde{\mathbb{U}}_{k+l|k}^i := \Delta\mathbb{U}_{k+l|k}^{i,i_1} \times \ldots \times \Delta\mathbb{U}_{k+l|k}^{i,i_{N_s^i}}$.

The feedback $\mathbf{K}_k^i \in \mathbb{K}^i \subseteq \mathbb{R}^{m^i N \times q^i}$ has the following block structure:

$$\mathbf{K}_k^i = \begin{bmatrix} 0_{m^i \times \tilde{m}^i - m^i} & 0_{m^i \times \tilde{m}^i - m^i} & \cdots & 0_{m^i \times \tilde{m}_\tau^i} \\ K_{k+1,k+1-\tau_{\max}|k}^i & 0_{m^i \times \tilde{m}^i - m^i} & \cdots & 0_{m^i \times \tilde{m}_\tau^i} \\ \vdots & \ddots & \ddots & \vdots \\ K_{k+N-1,k+1-\tau_{\max}|k}^i & \cdots & K_{k+N-1,k+N-1-\tau_{\max}|k}^i & 0_{m^i \times \tilde{m}_\tau^i} \end{bmatrix}, \tag{3.17}$$

where $\tilde{m}^i_\tau := \tau_{\max}(\tilde{m}^i - m^i)$, $K^i_{l,r|k} \in \mathbb{R}^{m^i \times (\tilde{m}^i - m^i)}$ is the feedback gain at time l for the uncertainty of the inputs of connected subsystems $\mathcal{P}^j$, $j \in \mathcal{N}^i$ at time r. The set $\mathbb{K}^i$ encodes the block structure shown in (3.17), which ensures that at time $k+l$ only the uncertainties which are known at this time are used in the control law. Considering the control law (3.12) and the prediction model (3.13), the set of admissible control policies is given by

$$\mathcal{K}^i\left(\hat{\tilde{x}}^i_{k|k}, \tilde{\mathbf{U}}^i_k, \boldsymbol{\mathcal{D}}^i_k\right) := \left\{ (\mathbf{K}^i_k, \hat{\mathbf{u}}^i_k) \;\middle|\; \begin{array}{l} \forall \delta\hat{\tilde{\mathbf{u}}}^i_k \in \boldsymbol{\mathcal{D}}^i_k : \\ \hat{\tilde{\mathbf{x}}}^i_k = \tilde{\mathbf{A}}^i \hat{\tilde{x}}^i_{k|k} + \tilde{\mathbf{B}}^i \hat{\mathbf{u}}^i_k + (\tilde{\mathbf{B}}^i T^i_{\mathbf{K}} \mathbf{K}^i_k + \mathbf{G}^i)\delta\hat{\tilde{\mathbf{u}}}^i_k, \\ \hat{\tilde{\mathbf{x}}}^i_k \in \tilde{\mathbf{X}}^i, \quad \hat{\mathbf{u}}^i_k + T^i_{\mathbf{K}}\mathbf{K}^i_k \delta\hat{\tilde{\mathbf{u}}}^i_k \in \tilde{\mathbf{U}}^i_k, \quad \mathbf{K}^i_{k|k} \in \mathbb{K}^i \end{array} \right\}. \tag{3.18}$$

In order to simplify the notation, the following matrices are defined similar to the ones in [39]:

$$\mathbf{F}^i_{\hat{x}^i} := \begin{bmatrix} C_{\tilde{\mathbf{X}}^i}\tilde{\mathbf{A}}^i \\ 0_{h^i_{\mathbf{U}}N \times \tilde{n}^i} \end{bmatrix}, \quad \mathbf{F}^i_k := \begin{bmatrix} C_{\tilde{\mathbf{X}}^i}\tilde{\mathbf{B}}^i \\ C_{\tilde{\mathbf{U}}^i_k} \end{bmatrix}, \quad \mathbf{F}^i_{\boldsymbol{\delta}^i} := \begin{bmatrix} C_{\tilde{\mathbf{X}}^i}\mathbf{G}^i \\ 0_{h^i_{\mathbf{U}}N \times q^i} \end{bmatrix}, \quad \mathbf{f}^i_k := \begin{bmatrix} b_{\tilde{\mathbf{X}}^i} \\ b_{\tilde{\mathbf{U}}^i_k} \end{bmatrix}. \tag{3.19}$$

At time k, a control law admissible for the constraints is a pair $(\mathbf{K}^i_k, \hat{\mathbf{u}}^i_k)$ which satisfies the state, input and terminal constraint, (3.12), (3.17), and the dynamics (3.13) for all disturbances given by the set (3.16). The set (3.18) can be rewritten as follows:

$$\mathcal{K}^i\left(\hat{\tilde{x}}^i_{k|k}, \tilde{\mathbf{U}}^i_k, \boldsymbol{\mathcal{D}}^i_k\right) = \left\{ (\mathbf{K}^i_k, \hat{\mathbf{u}}^i_k) \;\middle|\; \begin{array}{l} \mathbf{K}^i_{k|k} \in \mathbb{K}^i : \\ \mathbf{F}^i_k \hat{\tilde{\mathbf{u}}}^i_{k|k} + \max\limits_{\delta\hat{\tilde{\mathbf{u}}}^i_k \in \boldsymbol{\mathcal{D}}^i_k} \left(\mathbf{F}^i_k T^i_{\mathbf{K}} \mathbf{K}^i_k + \mathbf{F}^i_{\boldsymbol{\delta}}\right) \delta\hat{\tilde{\mathbf{u}}}^i_k \le \mathbf{f}^i_k - \mathbf{F}^i_{\hat{x}^i}\hat{\tilde{x}}^i_{k|k} \end{array} \right\}, \tag{3.20}$$

where the maximization is to be interpreted row-wise and the dual of the maximization of the j-th row is given by

$$\min_{s^i_{k,j}} (b_{\boldsymbol{\mathcal{D}}^i_k})^T s^i_{k,j} \quad \text{s.t. } (C_{\boldsymbol{\mathcal{D}}^i_k})^T s^i_{k,j} = (\mathbf{F}^i_k T^i_{\mathbf{K}} \mathbf{K}^i_k + \mathbf{F}^i_{\boldsymbol{\delta}^i})_j, \quad s^i_{k,j} \ge 0, \tag{3.21}$$

where $(\cdot)_j$ denotes the j-th row. Under the assumption that the disturbance set contains 0 in its interior, i.e. $\exists \delta\hat{\tilde{\mathbf{u}}}^i_k \in \operatorname{int}(\boldsymbol{\mathcal{D}}^i_k)$, Slater's constraint qualification holds and the row-wise maximization in (3.20) and minimization in (3.21) give the same result. Considering that the primal is a maximization problem any feasible pair of dual parameters can be used to upper bound the result of the maximization (cf. Section 2.4 and note that $\max f_{\text{obj}} = \min -f_{\text{obj}}$). By combining the vectors $s^i_{k,j}$ a matrix S^i_k with slack variables is obtained, and utilizing (3.21), the set of admissible control policies is given by

$$\mathcal{K}^i\left(\hat{\tilde{x}}^i_{k|k}, \tilde{\mathbf{U}}^i_k, \boldsymbol{\mathcal{D}}^i_k\right) = \left\{ (\mathbf{K}^i_k, \hat{\mathbf{u}}^i_k) \;\middle|\; \begin{array}{r} \exists S^i_k \ge 0, \mathbf{K}^i_{k|k} \in \mathbb{K}^i : \\ S^i_k C_{\boldsymbol{\mathcal{D}}^i_k} = (\mathbf{F}^i_k T^i_{\mathbf{K}} \mathbf{K}^i_k + \mathbf{F}^i_{\boldsymbol{\delta}}), \\ \mathbf{F}^i_k \hat{\tilde{\mathbf{u}}}^i_k + S^i_k b_{\boldsymbol{\mathcal{D}}^i_k} \le \mathbf{f}^i_k - \mathbf{F}^i_{\hat{x}^i}\hat{\tilde{x}}^i_{k|k} \end{array} \right\}. \tag{3.22}$$

Because this set is the projection of a convex polytope, it is convex (cf. [39] for the non-distributed case). Furthermore, because the maximization in (3.20) is a convex problem there is no duality gap and it follows that (3.20) and (3.22) are equivalent. In combination with a suitable quadratic cost function for the states and inputs, a single quadratic program is obtained.

This formulation is particularly useful for the distributed case, since $\hat{\mathbf{u}}_k^i$ can be used as prediction of future inputs of $\mathcal{P}^i$. Furthermore, the control law (3.12) will maintain the state and input constraints over the prediction horizon in the presence of uncertainties. Deviations from the planned input sequence $\hat{\mathbf{u}}_k^i$ can be bounded by the following set $\Delta\mathbf{U}_k^i = \Delta\mathbb{U}_{k|k}^i \times \ldots \times \Delta\mathbb{U}_{k+N-1|k}^i$:

$$\Delta\mathbf{U}_k^i = \left\{ \delta\hat{\mathbf{u}}_k^i \in \mathbb{R}^{m^i N} \,\middle|\, \exists \delta\hat{\hat{\mathbf{u}}}_k^i \in \boldsymbol{\mathcal{D}}_k^i : \delta\hat{\mathbf{u}}_k^i = \mathbf{K}_k^i \delta\hat{\hat{\mathbf{u}}}_k^i \right\}, \tag{3.23}$$

and these sets could be directly communicated to the controllers $\mathcal{C}^j$ of interconnected subsystems $j \in \mathcal{N}^j$. However, computing the projection in (3.23) may be computationally expensive and may lead to a time-varying number of faces in the polytope $\Delta\mathbf{U}_k^{i,i}$. Both of these issues are avoided here by exploiting the structure of the problem formulation in (3.20).

Note that the term $S_k^i b_{\boldsymbol{\mathcal{D}}_k^i}$ in (3.22) tightens the state and input constraints on the nominal prediction, such that they are satisfied for the uncertain system and the affine uncertainty feedback. By collecting the rows of S_k^i corresponding to the local input constraints $\mathbb{U}_{k+l|k}^i$ in a matrix $S_{\mathbb{U}_{k+l|k}}^i \in \mathbb{R}^{h_{\mathbb{U}}^i}$, it follows from (3.19) to (3.22) that

$$S_{\mathbb{U}_{k+l|k}}^i b_{\boldsymbol{\mathcal{D}}_k^i} = \max_{\delta\hat{\mathbf{u}}_k^i \in \boldsymbol{\mathcal{D}}_k^i} \left(C_{\mathbb{U}_{k+l|k}}^i \mathbf{K}_k^{i,l} \right) \delta\hat{\hat{\mathbf{u}}}_k^i, \tag{3.24}$$

with row-wise maximization and $\mathbf{K}_k^i = (\mathbf{K}_k^{i,0}, \ldots, \mathbf{K}_k^{i,N-1})$. Based on this observation, the following set can be defined, which contains all possible deviations of the affine disturbance feedback (3.12) from the planned input sequence $\hat{\mathbf{u}}_k^i$:

$$\Delta\mathbb{U}_{k+l|k}^{i,i} = \left\{ \delta\hat{u}_{k+l|k}^i \in \mathbb{R}^{m^i} \,\middle|\, C_{\mathbb{U}_{k+l|k}}^i \delta\hat{u}_{k+l|k}^i \leq S_{\mathbb{U}_{k+l|k}}^i b_{\boldsymbol{\mathcal{D}}_k^i} \right\}, \tag{3.25}$$

and $\Delta\mathbb{U}_{k+l|k}^{i,i} = \mathcal{B}_{\epsilon_n}^{m^i}(0)$, where $\epsilon_n > 0$ denotes the numerical tollerance, if the right hand side in (3.25) is empty. Thus, Slater's constraint qualification holds for the maximization in (3.20). These sets are then communicated to the interconnected controllers for all $l \in \{0, \ldots, N-1\}$.

3.3. Distributed MPC Algorithm

In the distributed scheme each controller $\mathcal{C}^i$ communicates its state x_k^i, the sequence of predicted nominal inputs $\hat{u}_{k+l|k}^i$, and the uncertainty of the prediction $\Delta\mathbb{U}_{k+l|k}^i$

for all $l \in \{0, \dots, N-1+\tau_{\max}\}$ to the controllers $\mathcal{C}^j$, for all $j : i \in \mathcal{N}^j$, i.e. to all interconnected controllers. In order to compensate for communication delays, the sequences are prolonged according to:

$$\hat{u}^i_{k+l|k} := K^i_{\mathbb{T}} \hat{\tilde{x}}^i_{k+l|k}, \quad \Delta\mathbb{U}^i_{k+l|k} := \mathbb{U}^i, \quad \forall l \in \{N-1, \dots, N-1+\tau_{\max}\}. \tag{3.26}$$

To ensure feasibility of the scheme, local control actions have to be consistent with previously communicated information. To achieve this, each controller shifts its uncertain communicated information one step forward to obtain new input constraints for the local robust MPC problem at the next time step for $l \in \{1, \dots, N-1\}$:

$$\mathbb{U}^i_{k+l|k+1} := \Delta\mathbb{U}^i_{k+l|k} \oplus \{\hat{u}^i_{k+l|k}\} \subseteq \mathbb{U}^i_{k+l|k}, \tag{3.27}$$

where $\oplus$ denotes the minkowski sum, the last inequality directly follows from the definition of $\Delta\mathbb{U}^i_{k+l|k}$. If the controller $\mathcal{C}^i$ receives information from controller $\mathcal{C}^j$, it computes the corresponding delay $\tau^{i,j}_k$ and stores the most recent information in a buffer. The most recent values are given by $\bar{k}^{i,j} = \arg\max_{0 \le l \le k} l - \tau^{i,j}_l$ and the corresponding delay by $\tau^{i,j}_{\bar{k}}$. In each time step these values are used to initialize the predictions for interconnected subsystems as follows:

$$\hat{x}^{i,j}_{\bar{k}-\tau^{i,j}_{\bar{k}}|k} := x^j_{\bar{k}-\tau^{i,j}_{\bar{k}}}, \qquad \forall l \in \{-\tau^{i,j}_{\bar{k}}, \dots, N-1\}, \tag{3.28}$$

$$\hat{u}^{i,j}_{k+l|k} := u^j_{k+l|\bar{k}-\tau^{i,j}_{\bar{k}}}, \qquad \forall l \in \{-\tau^{i,j}_{\bar{k}}, \dots, N-1\}, \tag{3.29}$$

$$\Delta\mathbb{U}^{i,j}_{k+l|k} := \Delta\mathbb{U}^j_{k+l|\bar{k}-\tau^{i,j}_{\bar{k}}}, \qquad \forall l \in \{-\tau^{i,j}_{\bar{k}}, \dots, N-1\}, \tag{3.30}$$

for all $(i,j) \in \mathcal{N} \times \mathcal{N}^i$. Beyond the prediction horizon it is assumed that the inputs of interconnected subsystems are given by $\hat{u}^{i,j}_{k+l|k} := 0$, and $\Delta\mathbb{U}^{i,j}_{k+l|k} := \mathbb{U}^j$ for all $l > N-1$ (cf. Assumption 3.3).

Finally, the following local optimization problem is obtained, which depends on the local state, the time-varying input constraints, and the delayed information of interconnected controllers:

$$\begin{aligned} V^{i*}(\hat{\tilde{x}}^i_{k|k}, \hat{\mathbf{u}}^{\setminus i}_k, \hat{\mathbf{u}}^i_k) = \min_{\hat{\mathbf{u}}^i_k, \mathbf{K}^i_{k|k}} \; & \|\hat{x}^i_{\tau k+N|k}\|^2_{\tilde{P}^i} + \sum_{l=0}^{N-1} \|\hat{\tilde{x}}^i_{k+l|k}\|^2_{\tilde{Q}^i} + \|\hat{\tilde{u}}^i_{k+l|k}\|^2_{\tilde{R}^i} \\ \text{s.t.} \quad & \hat{\tilde{x}}^i_{k+l+1|k} = \tilde{A}^i \hat{\tilde{x}}^i_{k+l|k} + \tilde{B}^i \hat{\tilde{u}}^i_{k+l|k}, \quad \forall l \in \{0, \dots, N-1\} \\ & \hat{x}^i_{\tau k+N|k} = (\hat{\tilde{x}}^i_{k+N|k}, \dots, \hat{\tilde{x}}^i_{k+N-\tau_{\max}+1|k}), \\ & \left(\mathbf{K}^i_k, \hat{\mathbf{u}}^i_k\right) \in \mathcal{K}^i\left(\hat{\tilde{x}}^i_{k|k}, \tilde{\mathbf{U}}^i_k, \boldsymbol{\mathcal{D}}^i_k\right), \end{aligned} \tag{3.31}$$

and the input sequences of interconnected subsystems $\hat{\mathbf{u}}^{\setminus i}_k$ are fixed to values obtained from delayed communication. The last line in (3.31) implies that the state and input constraints and the terminal constraint are satisfied for all disturbances

(3.16), and $(\hat{\mathbf{u}}^i_k, \mathbf{K}^i_{k|k})^*$ denotes the locally optimized pair of the control policy parameters. In the following $\delta\mathbf{u}^{\setminus i}_{k+1} = \hat{\mathbf{u}}^{\setminus i}_{k+1} - \hat{\mathbf{u}}^{\setminus i}_{k}$ denotes the difference between predictions used by $\mathcal{C}^i$ at consecutive time steps.

Algorithm 3.1 describes the overall distributed MPC scheme. The input constraints used by $\mathcal{C}^i$ and consequently the disturbance sets of interconnected controllers $\mathcal{C}^j$ are time-varying. The first control obtained from (3.31) is applied in a receding horizon fashion, such that $u^i_k = \hat{u}^i_{k|k}$ and it follows that $\delta\hat{\hat{u}}^{i,j}_{k+l|k+l} = 0$ for all $l \leq -\tau_{\max}$.

The input constraints restrict the solution of $\mathcal{C}^i$ at $k+1$ to input sequences which are consistent with previously communicated information. Thus, they ensure that a controller can only deviate from its plan by a previously communicated amount. It follows from (3.27) and the communication scheme that

$$\Delta\mathbb{U}^{i,j}_{k+l|k+1} \oplus \{\hat{u}^{i,j}_{k+l|k+1}\} \subseteq \Delta\mathbb{U}^{i,j}_{k+l|k} \oplus \{\hat{u}^{i,j}_{k+l|k}\}, \tag{3.32}$$

i.e. the uncertainty for a time $k+l$ predicted at time k cannot increase at $k+1$. In the following theorem this property of Algorithm 3.1 and Assumption 3.3 is used

---

Algorithm 3.1: Distributed MPC Algorithm

INITIALIZATION: At $k = 0$ the state $x^j_{-\tau^{i,j}_k}$ and initial input sequence $u^j_{l|-\tau^{i,j}_k}$ for all $l \in \{-\tau^{i,j}_k, \dots, N-1\}$ are known to all $\mathcal{C}^i$, $i : j \in \mathcal{N}^i$, and $\Delta\mathbb{U}^{i,j}_{l|0} := \mathbb{U}^j$.
for all $k \geq 0$ **do**
  for all subsystems $i \in \mathcal{N}$ **do**
    for all $j \in \mathcal{N}^i$ **do**
      If information is received, compute the delay $\tau^{i,j}_k$ based on timestamps
      Store the most recent information (i.e. received at $\bar{k}^{i,j} = \arg\max_{0 \leq l \leq k} l - \tau^{i,j}_l$) in a buffer
      Update the predictions according to (3.28) to (3.30)
    end for
    Measure the local state x^i_k and compute the local predictions $\hat{x}^{i,j}_{k|k}$ according to (3.5)
    Solve the local optimization problem (3.31), and apply $u^i_k = \hat{u}^i_{k|k}$
    Prolong the sequence of inputs and disturbance sets according (3.26)
    Compute the input consistency constraints $\mathbb{U}^i_{k+l|k+1}$ for $k+1$ according to (3.25) and (3.27)
    Communicate x^i_k, $\hat{u}^i_{k+l|k}$, $\Delta\mathbb{U}^i_{k+l|k}$ for all $l \in \{0, \dots, N-1+\tau_{\max}\}$, and a timestamp to all $\mathcal{C}^j$, $j : i \in \mathcal{N}^j$
  end for
  Set $k := k+1$
end for

---

to establish recursive feasibility of the distributed MPC algorithm.

Theorem 3.1. *If Assumption 3.3 holds and the problem* (3.31) *is feasible for all subsystems at time $k = 0$, the problem remains feasible for all times $k > 0$, and the state and input constraints $x_k \in \mathbb{X}$, $u_k^i \in \mathbb{U}^i$ are satisfied for all times $k > 0$.*

Proof. Feasibility of the problem at $k = 0$ implies that there exists a pair $(\mathbf{K}_k^i, \hat{\hat{\mathbf{u}}}_k^i)$ which steers the system into the terminal constraint $\tilde{\mathbb{T}}^i$ in N steps under all possible realizations of the uncertainties $\delta\hat{\hat{\mathbf{u}}}_k^i$, while satisfying all constraints. Furthermore, at $k+1$ the uncertainty $\delta\hat{\hat{u}}_{k+1-\tau_{\max}}^i$ will be known exactly. If new predictions are received from interconnected controllers, the consistency constraints ensure that $\delta u_{k+l|k+1}^{i,j} = \hat{u}_{k+l|k+1}^{i,j} - \hat{u}_{k+l|k}^{i,j} \in \Delta\mathbb{U}_{k+l|k}^{i,j}$ for all $l \in \{1, \dots, N\}$ and all $j \in \mathcal{N} \setminus i$. This implies that at $k+1$ a feasible input sequence up to $k+N-2$ is given by

$$\begin{bmatrix} \hat{u}_{k+1|k+1}^i \\ \vdots \\ \hat{u}_{k+N-2|k+1}^i \end{bmatrix} = (\hat{\mathbf{u}}_k^i + \mathbf{K}_k^i \delta\mathbf{u}_{k+1}^{\setminus i})_+, \quad (\hat{\mathbf{u}}_k^i)_+ = \begin{bmatrix} \hat{u}_{k+1|k}^i \\ \vdots \\ \hat{u}_{k+N-2|k}^i \end{bmatrix}, \tag{3.33}$$

where $(\cdot)_+$ denotes shifting a sequence one time step forward, i.e. the terminal constraint is satisfied for $k+N-1$.

Furthermore, by Assumption 3.3 the control law (3.7) renders the terminal constraint robust forward invariant while satisfying the local input constraints. Thus, the terminal controller can be used to prolong the sequence of inputs and controllers $(\mathbf{K}_k^i, \hat{\hat{\mathbf{u}}}_k^i)$, such that the terminal constraint is satisfied at $k+N$ under all realizations of the uncertain parameters. Because the terminal constraint is a subset of the state constraints, it follows that all constraints are satisfied at $k+N$.

To this end, the terminal control law is applied to the state $\tilde{x}_{k+N+1-\tau_{\max}}^i$:

$$u_{k+N}^i = \mathbf{K}_{\mathbb{T}}^i \tilde{x}_{k+N+1-\tau_{\max}}^i, \tag{3.34}$$

$$u_{k+N}^i = \mathbf{K}_{\mathbb{T}}^i \hat{\hat{x}}_{k+N+1-\tau_{\max}}^i + \mathbf{K}_{\mathbb{T}}^i \delta\hat{\hat{x}}_{k+N+1-\tau_{\max}}^i, \tag{3.35}$$

where $\hat{\hat{x}}_{k+N+1-\tau_{\max}}^i$ can be found by forward recursion of (3.5) based on $\hat{\hat{x}}_{k-\tau_k|k+1}^i$ and $\hat{\hat{u}}_{k+l|k+1}^i$ with $l \in \{-\tau_k, \dots, N-\tau_{\max}\}$, resulting in

$$\hat{u}_{k+N-1|k+1}^i = \mathbf{K}_{\mathbb{T}}^i \hat{\hat{x}}_{k+N+1-\tau_{\max}}^i.$$

The second term in (3.35) is uncertain and by recursively applying (3.6) it can be seen that there exists matrices $\mathcal{A}_l^i$, such that

$$\delta\hat{\hat{x}}_{k+N+1-\tau_{\max}|k+1}^i := \textstyle\sum_{l=2-\tau_{\max}}^{N-\tau_{\max}} \mathcal{A}_l^i \delta\hat{\hat{u}}_{k+l|k}^i.$$

Overall, it follows that a feedback for u_{k+N}^i which satisfies the causality constraint (3.17) can be constructed based on $\mathbf{K}_k^i$ and the terminal control law, such that $\mathbf{K}_{k+1}^i$

is given by

$$\begin{bmatrix} 0_{m^i\times\tilde{m}^i-m^i} & 0_{m^i\times\tilde{m}^i-m^i} & \cdots & 0_{m^i\times\tilde{m}^i-m^i} & 0_{m^i\times\tilde{m}^i_\tau} \\ K^i_{k+2,k+2-\tau_{\max}|k} & 0_{m^i\times\tilde{m}^i-m^i} & \cdots & 0_{m^i\times\tilde{m}^i-m^i} & 0_{m^i\times\tilde{m}^i_\tau} \\ \vdots & \ddots & \ddots & \vdots & \vdots \\ K^i_{k+N-1,k+2-\tau_{\max}|k} & \cdots & K^i_{k+N-1,k+N-1-\tau_{\max}|k} & 0_{m^i\times\tilde{m}^i-m^i} & 0_{m^i\times\tilde{m}^i_\tau} \\ \mathbf{K}^i_{\mathbb{T}}\mathcal{A}^i_{2-\tau_{\max}} & \cdots & \cdots & \mathbf{K}^i_{\mathbb{T}}\mathcal{A}^i_{N-\tau_{\max}} & 0_{m^i\times\tilde{m}^i_\tau} \end{bmatrix},$$

Finally, it follows from (3.32) that sets $\Delta\mathbb{U}^{i,j}_{k+l|k+1}$ which bound the uncertainty are shifted by the same amount as the predictions (i.e. $\delta v^{i,j}_{k+l|k}$) and cannot increase over time. Considering the dynamics (3.13) it follows that the control policy at $k=1$, constructed based on the control policy at $k=0$, is feasible. Feasibility for all times $k>1$ follows by induction over k, which directly implies that the state constraints hold for all time under all realizations of the uncertainties. □

Next, the framework of input-to-state practical stability (ISpS) presented in Section 2.3 is used to investigate the stability properties of the overall system in closed-loop with the distributed MPC scheme given in Algorithm 3.1 with respect to Σ^i. To this end, the extended state vector $\tilde{\mathbf{z}}^i_k = (\tilde{z}^i_{k|k};\ldots;\tilde{z}^i_{k+N-1|k})$ and corresponding set $\mathbf{\Sigma}^i := \Sigma^i\times\ldots\times\Sigma^i$ are defined. The local prediction for $\tilde{\mathbf{z}}^i_k$ is denoted by $\hat{\tilde{\mathbf{z}}}^i_k$, and $\delta\hat{\mathbf{z}}^i_k := \hat{\tilde{\mathbf{z}}}^i_k - \hat{\tilde{\mathbf{z}}}^i_{k-1}$ denotes a deviation from previously received trajectories. Next, let $\boldsymbol{\mathcal{Z}}^i$ denote the set of initial extended states $\tilde{\mathbf{z}}^i_k$ for which the optimization problem (3.31) is feasible. Furthermore, $\mathbf{V}^i_z(\hat{\tilde{\mathbf{z}}}^i_k)$ denotes the cost formulated with respect to $\hat{\tilde{\mathbf{z}}}_k$, i.e. $\mathbf{V}^i_z(\hat{\tilde{\mathbf{z}}}^i_k)$ is obtained by substituting the dynamics into $\mathbf{V}^i(\tilde{\mathbf{x}}^i_k,\tilde{\mathbf{x}}^i_k)$ for $\tilde{x}^i_{k+N|k}$.

Theorem 3.2. *If the problem* (3.31) *is feasible for all subsystems at time* $k=0$, *the distributed MPC scheme in Algorithm 3.1 applied in closed-loop renders the extended state vectors* $\tilde{\mathbf{z}}^i_k$ *ISpS in* $\boldsymbol{\mathcal{Z}}^i$ *with respect to* $\mathbf{\Sigma}^i$ *and the uncertainties* $\delta\hat{\mathbf{z}}^i_k$.

Proof. In order to establish the ISpS property of the distributed MPC, the cost function $\mathbf{V}^i_z(\hat{\tilde{\mathbf{z}}}^i_k)$ is used as ISpS-Lyapunov function. Note that the cost does not depend on $(\mathbf{K}^i_{k|k})^*$ but only on the planned input and state sequences. The lower bound $\alpha^i_1(\|\hat{\tilde{\mathbf{z}}}^i_k\|_{\mathbf{\Sigma}^i})$ follows directly from the definition of the cost. Similarly, since $\mathbf{V}^i_z(\hat{\tilde{\mathbf{z}}}^i_k)$ is a quadratic function of $\hat{\tilde{\mathbf{z}}}^i_k$ and zero if $\hat{\tilde{\mathbf{z}}}^i_k\in\mathbf{\Sigma}^i$ (cf. Assumption 3.3.4) it directly follows that there exists $\alpha^i_2\in\mathscr{K}$ such that $\mathbf{V}^i_z(\hat{\tilde{\mathbf{z}}}^i_k)\le\alpha^i_2(\|\hat{\tilde{\mathbf{z}}}^i_k\|_{\mathbf{\Sigma}^i})$.

Next, consider the case that $\delta\hat{\mathbf{z}}^i_k=0$. The feasible input sequence constructed in the proof of Theorem 3.1 can be used to bound the cost at $k+1$ as follows:

$$\mathbf{V}^i_z(\hat{\tilde{\mathbf{z}}}^i_{k+1}) - \mathbf{V}^i_z(\hat{\tilde{\mathbf{z}}}^i_k) \le \|\hat{\tilde{x}}^i_{k+N|k+1}\|^2_{\tilde{Q}^i} + \|\hat{\tilde{u}}^i_{k+N|k+1}\|^2_{\tilde{R}^i} + \|\hat{x}^i_{\tau k+N+1|k+1}\|^2_{\tilde{P}^i} \quad (3.36)$$

$$- \|\hat{\tilde{x}}^i_{k|k}\|^2_{\tilde{Q}^i} - \|\hat{\tilde{u}}^i_{k|k}\|^2_{\tilde{R}^i} - \|\hat{x}^i_{\tau k+N|k}\|^2_{\tilde{P}^i}. \quad (3.37)$$

Assuming $\hat{u}^{i,j}_{k+N|k+1}=0$ and considering Assumption 3.3 it follows that

$$\mathbf{V}^i_z(\hat{\tilde{\mathbf{z}}}^i_{k+1}) - \mathbf{V}^i_z(\hat{\tilde{\mathbf{z}}}^i_k) \le -\|\hat{\tilde{x}}^i_{k|k}\|^2_{\tilde{Q}^i} - \|\hat{\tilde{u}}^i_{k|k}\|^2_{\tilde{R}^i}. \quad (3.38)$$

Applying this inequality recursively results in

$$\mathbf{V}_z^i(\hat{\tilde{\mathbf{z}}}_{k+N}^i) - \mathbf{V}_z^i(\hat{\tilde{\mathbf{z}}}_k^i) \leq \sum_{l=0}^{N-1} -\|\hat{\tilde{x}}_{k+l|k+l}^i\|_{\tilde{Q}^i}^2 - \|\hat{\tilde{u}}_{k+l|k+l}^i\|_{\tilde{R}^i}^2. \tag{3.39}$$

First, the case $\delta\hat{\mathbf{z}}_{k+l}^i = 0$ is considered, and the initialization (3.33) is used recursively starting at time k. This results in $\hat{\tilde{x}}_{k+l|k+l}^i = \hat{\tilde{x}}_{k+l|k}^i$ and $\hat{\tilde{u}}_{k+l|k+l}^i = \hat{\tilde{u}}_{k+l|k}^i$. Therefore, it follows that

$$\mathbf{V}_z^i(\hat{\tilde{\mathbf{z}}}_{k+N}^i) - \mathbf{V}_z^i(\hat{\tilde{\mathbf{z}}}_k^i) \leq \sum_{l=0}^{N-1} -\|\hat{\tilde{x}}_{k+l|k}^i\|_{\tilde{Q}^i}^2 - \|\hat{\tilde{u}}_{k+l|k}^i\|_{\tilde{R}^i}^2. \tag{3.40}$$

Next, the case $\delta\hat{\mathbf{z}}_k^i \neq 0$ is considered. Because $\hat{\tilde{\mathbf{z}}}_{k+N}^i$ linearly depends on the disturbances $\delta\hat{\mathbf{z}}_k^i$, $\mathbf{V}_z^i(\hat{\tilde{\mathbf{z}}}_k^i)$ is Lipschitz continuous, and by construction of $\tilde{Q}^i$ and $\tilde{R}^i$, there exists $\alpha_3^i \in \mathscr{K}$ and $\sigma^i \in \mathscr{K}$ such that

$$\mathbf{V}_z^i(\hat{\tilde{\mathbf{z}}}_{k+N}^i) - \mathbf{V}_z^i(\hat{\tilde{\mathbf{z}}}_k^i) \leq -\alpha_3^i(\|\hat{\tilde{\mathbf{z}}}_k^i\|_{\mathbf{\Sigma}^i}) + \sigma^i(\|\delta\hat{\mathbf{z}}_{[k:k+N-1]}^i\|). \tag{3.41}$$

Theorem 2.1 with $L = N$ and $d_1 = d_2 = 0$ then implies that

$$\|\hat{\tilde{\mathbf{z}}}_k^i\|_{\mathbf{\Sigma}^i} \leq \beta_c(\|\hat{\tilde{\mathbf{z}}}_0^i\|_{\mathbf{\Sigma}^i}, k) + \gamma_c(\|\delta\hat{\mathbf{z}}_{[0:k-1]}^i\|), \quad \forall k > 0.$$

Finally, due to the consistency constraints the maximal difference $\delta\tilde{\mathbf{z}}_{\max} := \max\|\tilde{\mathbf{z}}_k^i - \hat{\tilde{\mathbf{z}}}_k^i\|$ is bounded, and it directly follows that

$$\|\tilde{\mathbf{z}}_k^i\|_{\mathbf{\Sigma}^i} \leq \beta_c(\|\tilde{\mathbf{z}}_0^i\|_{\mathbf{\Sigma}^i}, k) + \gamma_c(\|\delta\hat{\mathbf{z}}_{[0:k-1]}^i\|) + d_c, \quad \forall k > 0,$$

where $d_c := \alpha_1^i(3\alpha_2^i(\delta\tilde{\mathbf{z}}_{\max}))^{-1} + \delta\tilde{\mathbf{z}}_{\max}$. □

3.4. Simulation Results

To illustrate the class of interconnected systems, consider the control of a platoon of three vehicles in a leader-follower scenario shown in Figure 3.2. The control goal is to ensure that the vehicles follow the lead vehicle with a given constant spacing while avoiding collisions. Instead of following a given reference trajectory, the lead vehicle aims to achieve its own control goal, for instance reaching a destination

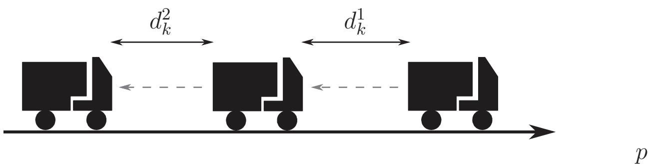

Figure 3.2.: Platooning vehicles in a leader-follower scenario with position p_k^i, distance d_k^i, and neighboring communication (dashed).

while reacting to traffic, speed limits, and other external influences. Therefore, the lead vehicle can only provide uncertain predictions to the followers. For illustrative purposes it is assumed that the vehicles cannot measure the distance to the preceding vehicle, i.e. the vehicles can only obtain information about neighboring subsystems by possibly delayed communication (cf. [42] for the case with distance measurements).

The local dynamics of the vehicles are modeled by double-integrators with sampling period of $\Delta T = 0.3$s and are given by

$$x^i_{k+1} = \begin{bmatrix} 1 & 0.3 \\ 0 & 1 \end{bmatrix} x^i_k + \begin{bmatrix} 0.045 \\ 0.3 \end{bmatrix} u^i_k \quad \forall i \in \{1, \ldots, 3\}. \tag{3.42}$$

The local states x^i_k are given by $x^i_k = ((x_p)^i_k - (i-1)c_s; (\dot{x}_p)^i_k)$, where $(x_p)^i_k$ is the position, $(\dot{x}_p)^i_k$ is the velocity, and $c_s = 15$m is the desired spacing between the vehicles. The local inputs represent the acceleration and are physically limited to $-3 \leq u_k \leq 3$. The common control goal is given by

$$\Sigma := \left\{ z \in \mathbb{R}^{n+m} \middle| \forall i \in \{1, 2\} : x^i = x^{i+1}, \, u^i_k = u^{i+1}_k \right\}.$$

The upper bound on the communication delay is $\tau_{\max} = 2$. Each vehicle is further coupled to its predecessor by constraints on the distance and the relative velocity given by $(-15\text{m}, -20\frac{\text{m}}{\text{s}}) \leq x^{i+1}_k - x^i_k \leq (100\text{m}, 20\frac{\text{m}}{\text{s}})$. In other words, the vehicles should not collide, the maximal distance between each pair of vehicles is 100m and the maximum speed difference is $20\frac{\text{m}}{\text{s}}$. In this example, each vehicle is only coupled to the preceding vehicle resulting in the index sets $\mathcal{N}^1 = \{1\}$, $\mathcal{N}^2 = \{2, 1\}$, $\mathcal{N}^3 = \{3, 2\}$, and the augmented state vector $\tilde{x}^i_k = (x^i_k; x^{i-1}_k)$ for $i = \{2, 3\}$. Introducing coupling between the first and third vehicle would lead to $\mathcal{N}^3 = \{3, 2, 1\}$ and may result in better control performance. However, this results in a higher computational complexity, more communication, and complicates Assumption 3.3.

The inputs of the last vehicle are constrained by $\mathbb{U}^{(3)} = [-3, 3]$. In order to fulfill Assumption 3.3, the input constraints have to be less restrictive along the platoon. For this example, the input sets are parametrized by $\mathbb{U}^{i-1} = c\mathbb{U}^i$ with $0 < c \leq 1$, i.e. the original input constraints are satisfied. Starting from $c = 1$ the parameter c was reduced until Assumption 3.3 holds, this was the case for $c = 0.74$. From a practical point of view, these tightened constraints are required due to the parallel computation and delayed communication. For instance, for a decelerating lead vehicle, a follower may only be able to react after a communication delay. Thus, only stronger declaration of the follower can prevent a collision.

In comparison to algorithms where the distance can be measured, Assumption 3.3 introduces some conservatism. In contrast, in [42] the distance can be measured without a delay and both the scaling factor c and the desired spacing can be chosen less conservatively. The cost function is identical for all followers with the weights:

$$\tilde{Q}^i = \begin{bmatrix} 5 & 0 & -5 & 0 \\ 0 & 1 & 0 & -1 \\ -5 & 0 & 5 & 0 \\ 0 & -1 & 0 & 1 \end{bmatrix}, \quad \tilde{R}^i = \begin{bmatrix} 0.1 & -0.1 \\ -0.1 & 0.1 \end{bmatrix}.$$

Simulation results for three vehicles, for the prediction horizon $N = 10$, and constant communication delay of $\tau_k^{i,j} = 2$ are shown in Figure 3.3. In the first row, the position, velocity, and input of the leader are shown. The subsequent rows show the distance to the preceding vehicle, the relative velocity, and the input of the followers. At time $t = 0$sec the distance between the vehicles starts to decrease, but due to the fact that the vehicles always deviate from previously communicated information the control error never goes to zero. Figure 3.4 shows the inputs of the first and second vehicle, as well as the predicted inputs and bounds at $t = 40$sec. In particular at $t = 40$sec, the lead vehicle communicates a mild breaking maneuver but actually applies full braking as shown in Figure 3.4. This leads to a decrease

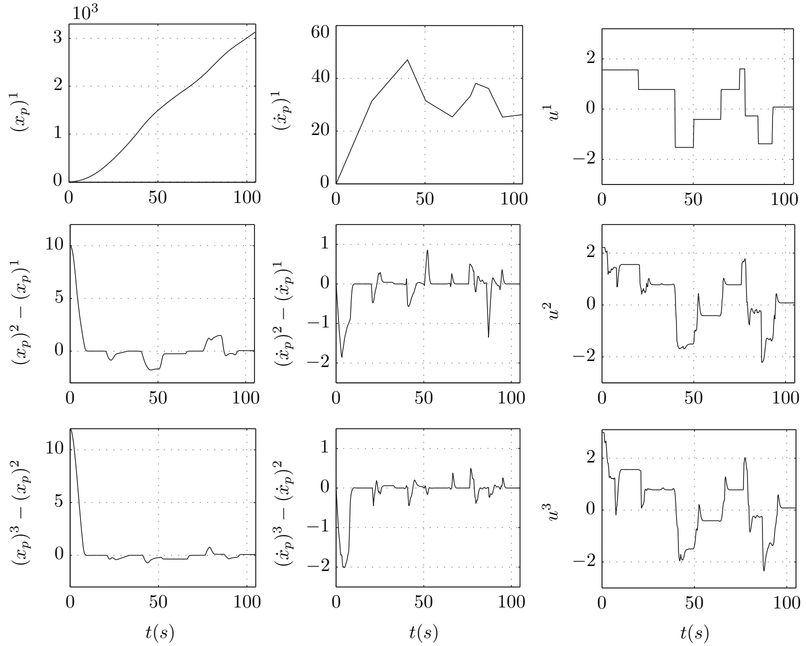

Figure 3.3.: Simulation results for a platoon of three vehicles using the distributed MPC scheme.

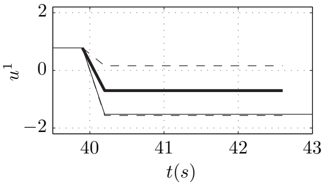

Figure 3.4.: Mismatch of communicated and actual behavior. The input of the lead vehicle and first follower deviates from the communicated predicted inputs (bold) but is within the communicated bounds (dashed).

in the distance between vehicle one and two, but collisions are robustly avoided. Similarly, the acceleration at ca. $t = 75$sec is not communicated by the lead vehicle, which leads to a temporary increase in distance between the first and second vehicle.

The control error of the third vehicle with respect to the second one is smaller because of the larger input set of the third vehicle. This enables the third vehicle to compensate deviations of the second vehicle from its communicated information more quickly.

Solving the local robust MPC problems for one vehicle and one time step takes between ca. 70ms and 100ms using one core of an AMD Phenom II X4 920 with 4GB RAM using CPLEX 12.1.

3.5. Discussion

In this chapter, a distributed closed-loop MPC algorithm based on results from robust optimization was presented which can be applied to dynamically decoupled linear systems coupled by convex constraints and costs.

A main advantage of the proposed algorithm is that the controllers $\mathcal{C}^i$ only require neighboring communication once per time-step and the local optimization problems are solved in parallel. The distributed MPC algorithm ensures that coupled constraints hold even if the communication between controllers is subject to bounded time-varying communication delays. This robustness is achieved by optimizing a control law which explicitly considers values which will be received at a later time. This control law can be computed efficiently by using approaches from centralized robust MPC and robust optimization.

While most distributed MPC algorithms only consider the stabilization of an a-priori fixed set point, the proposed method can also deal with more general control tasks, such as the synchronization of subsystem trajectories. Similar control tasks have been considered in [83], where a sequential distributed MPC for synchronization of non-linear systems is proposed which guarantees asymptotic stability. However, in [83] the subsystem do not optimize in parallel but in a given sequence. Furthermore, communication delays are not considered. Hence, the main complicating factors considered in this chapter are not present in [83].

In general only few results are available which consider distributed MPC with time-varying communication delays. Communication delays in distributed MPC for dynamically decoupled nonlinear systems coupled by a common cost function are analyzed in an input-to-state stability framework in [33]. However, the delays are assumed to be constant and no coupled constraints are considered, while the method proposed in this chapter offers strong robustness results even for time-varying delays. On the other hand, the construction of the terminal constraint and controller required by Assumption 3.3 is a hard problem if the coupling graph contains cycles. While small gain arguments may allow extending the proposed method to this case such results may be very conservative.

Robust local model predictive controllers are also employed in the distributed algorithms proposed in [91], [108], [109] for subsystems only interconnected by inequality constraints. However, these algorithms only offer robustness with respect to disturbances acting on the local systems, not uncertainties arising from the communication delays. More importantly, these algorithms are not directly applicable to the case of common control goals specified by coupled costs. A variant of the algorithm proposed in this chapter which considers local disturbances can be found in [42].

Finally, the robust satisfaction of constraints is guaranteed by means of constraints which ensure consistency of locally planned input sequences with previously communicated information. This principle is borrowed from [57] and [19], where a min-max distributed MPC for dynamically coupled nonlinear subsystems was proposed, but only a one-step delay is considered. Consistency constraints are also used in [26] to ensure only stability, not feasibility, and the consistency constraints are chosen a-priori by the designer. In contrast, in the algorithm at hand the consistency constraints are time-varying and constructively obtained online from the solution of the local robust MPC problems.

In the distributed MPC algorithm developed in this chapter, delayed communication and consistency constraints are used to ensure robust feasibility. However, a closer inspection of the proof of Theorem 3.2 reveals that in terms of the robust stability guarantees the consistency constraints only bound the increase of the cost due to communication and optimization. Even for the case $\tau_{\max} = 1$, where the controllers optimize in parallel and the state of the interconnected subsystem is known exactly, the cost may increase from one time step to the next due to the parallel optimization. Furthermore, the influence of the consistency constraints on the closed-loop performance of the algorithm is not clear. Hence, it is not clear how to compute consistency constraints in order to ensure good closed-loop performance, even if the consistency constraints are computed and adapted online as proposed in this chapter.

4. Cooperative Distributed Model Predictive Control

In this chapter, linear dynamics and problem formulations with decoupled constraints are considered. In contrast to the approach in the previous chapter, the algorithm discussed in this chapter utilizes communication between all controllers, is applicable to a wider class of interconnection structures, and introduces stronger cooperation between the controllers. The stronger cooperation between the controllers allows for a detailed analysis of the convergence properties, suboptimality, and the influence of the local optimization on the global convergence.

After presenting a common formulation resulting from Cases 2.2 and Case 2.4 (i.e. coupled dynamics and costs, no state constraints) and Case 2.3 (i.e. decoupled dynamics and constraints, coupled costs), the cooperative distributed model predictive control algorithm first proposed in [112] is briefly reviewed, and its properties are compared to the algorithm presented in the previous section. The main aim of this chapter is to provide bounds on the convergence rate of the cooperative distributed MPC algorithm, which also give insight into how the coupling between subproblems and the information exchanged by the controllers affect convergence. Furthermore, two approaches to determine parameters used in the algorithm to ensure fast convergence are proposed and compared.

It is well known that the iterative algorithm proposed in [112] is guaranteed to converge to the global optimum in the limit (cf. [104]). However, except for some of the results of this chapter which have been previously published in [44] no results on the convergence rate are available in the literature. In large parts, this chapter is based on the preliminary results on the convergence rate of the cooperative distributed MPC algorithm previously published in [44] and results submitted for publication in [46]. The analysis of the convergence properties of the algorithm with full communication is used as a starting point for developing the cooperative distributed MPC algorithm with event-based communication presented in the next chapter.

4.1. Review of Cooperative Distributed MPC with Parallel Optimization

This chapter focuses on linear discrete-time dynamics:

$$x_{k+1} = Ax_k + Bu_k, \tag{4.1}$$

and the cases 2.2 to 2.4 given in Section 2.1. For all three cases, the cost function $\mathbf{V}(\mathbf{x}_k, \mathbf{u}_k)$ can be equivalently rewritten into a function of the state x_k and the input sequence $\mathbf{u}_k$ by substituting the dynamics (4.1) into the cost function (2.5) (see Appendix A.1). This results in

$$V(x_k, \mathbf{u}_k) = \mathbf{u}_k^T H \mathbf{u}_k + x_k^T F \mathbf{u}_k + x_k^T H_x x_k, \tag{4.2}$$

with $H = H^T \succ 0$, $H \in \mathbb{R}^{Nm \times Nm}$, $F \in \mathbb{R}^{n \times Nm}$ and $H_x \in \mathbb{R}^{n \times n}$. Furthermore, decoupled local constraints $\mathbf{u}_k^i \in \mathbf{U}^i(x_k) \subseteq \mathbb{U}^i \times \ldots \times \mathbb{U}^i$ over the whole prediction horizon are obtained by considering the local input constraints, as well as substituting the dynamics into the state and terminal constraints (cf. Appendix A.1).

In this section, the cooperative algorithm first proposed in [112], which uses communication between all controllers in every time step and iteration, as well as its known properties are briefly reviewed. Throughout this chapter the following assumptions are made:

Assumption 4.1. *It is assumed that:*

1. *the communication between the controllers does not induce any uncertainties such as delays or packet loss,*
2. *the clocks of the controllers are synchronized and iterations are performed synchronously, i.e. after a fixed amount of time $\Delta t_p \ll \Delta t$ every controller has solved its local optimization and proceeds to the next iteration,*
3. *the distributed MPC algorithm is initialized with a feasible solution.*

Cooperative Distributed MPC Algorithm

Within the distributed MPC algorithm, given in Algorithm 4.1, the local input vector is optimized by each subsystem, in each time-step k, and in each iteration p:

$$\begin{aligned} \boldsymbol{\rho}_{k,p}^i := \arg\min\nolimits_{\mathbf{u}_k^i} \quad & V(x_k, \mathbf{u}_k) \\ \text{s.t.} \quad \mathbf{u}_k^i \in \mathbf{U}^i(x_k),\ & \mathbf{u}_k^j = \mathbf{u}_{k,p}^j, \forall j \in \mathcal{N} \setminus i. \end{aligned} \tag{4.3}$$

In other words, each controller only optimizes its local input, and the inputs of all other controllers $\mathcal{C}^j$, $j \in \mathcal{N} \setminus i$ are fixed to constant values obtained in the previous iteration. The algorithm can be seen as a form of primal decomposition.

After each iteration the optimized input sequences $\boldsymbol{\rho}^i_{k,p}$ are exchanged between all controllers and the following iteration is performed to obtain new input sequences for all subsystems:

$$\mathbf{u}_{k,p+1} := \textstyle\sum_{i\in\mathcal{N}} w^i_{k,p}\bar{\mathbf{u}}^i_{k,p}, \tag{4.4}$$

where $\bar{\mathbf{u}}^i_{k,p} := (\mathbf{u}^1_{k,p}; \ldots ; \boldsymbol{\rho}^i_{k,p}; \ldots ; \mathbf{u}^{N_s}_{k,p})$ denotes the candidate input sequence computed by controller $\mathcal{C}^i$, and thhe weights $w^i_{k,p} \in \mathbb{R}_{>0}$ have to satisfy $\sum_{i\in\mathcal{N}} w^i_{k,p} = 1$.

The iteration (4.4) can be equivalently formulated locally for all controllers $\mathcal{C}^i$, $i \in \mathcal{N}$:

$$\mathbf{u}^i_{k,p+1} := w^i_{k,p}\boldsymbol{\rho}^i_{k,p} + (1 - w^i_{k,p})\mathbf{u}^i_{k,p}. \tag{4.5}$$

The update of the input sequence (4.4) in each iteration can either be performed by a coordinator (i.e. centralized) or in a distributed fashion. Concerning the convergence results in this chapter, both these implementations are equivalent. However, the communication requirements are not identical. Therefore, in the following a distributed implementation is considered, i.e. each subsystem computes $\mathbf{u}_{k,p+1}$ in parallel based on (4.4).

It can be seen, that the distributed MPC algorithm is of the nonlinear Jacobi type [11] in which all local optimization problems are solved in parallel and exchanged between all controllers. In contrast to classic Jacobi algorithms, this algorithm also includes the convex update step (4.4), which ensures that the cost $V(x_k, \mathbf{u}_{k,p})$ is decreasing in the iterations p. In contrast to [112], the weights used in (4.5) may be time varying. This allows to consider algorithms which depend on the communicated information, e.g. weights which are optimized online. In this case, an algorithm may be obtained in which the local input sequences $\mathbf{u}^i_{k,p}$ are updated

Algorithm 4.1: Cooperative distributed MPC iterations in k for all $\mathcal{C}^i$ [112]

Given k, x_k, $\mathbf{u}_{k,0}$, $p = 0$, $p_{\max} > 0$, and $\epsilon > 0$:
while $p \leq p_{\max}$ **do**
    Solve (4.3) and communicate $\boldsymbol{\rho}^i_{k,p}$ to all controllers
    Compute $\mathbf{u}_{k,p+1}$ according to (4.4)
    if $\|\mathbf{u}_{k,p+1} - \mathbf{u}_{k,p}\| \leq \epsilon$ **then**
        break
    else
        $p := p + 1$
    end if
end while
Apply $u^i_{k|k,p}$ to the system (4.1)

sequentially (similar to nonlinear Gauss-Seidel algorithms [11]), but the optimization is still performed in parallel (see Section 4.3.2). Overall it can be seen, that the convergence speed may strongly depend on the weights $w^i_{k,p}$. Therefore, it is investigated in Section 4.3 how to choose the weights to obtain fast convergence.

When the distributed MPC is applied in closed loop, Algorithm 4.1 is performed in each time step until the stopping criterion $\|\mathbf{u}_{k,p+1} - \mathbf{u}_{k,p}\|^2 \leq \epsilon$ holds or the maximum number of iterations $p_{\max}$ is reached. In the algorithm it is assumed that the full state vector is known to each subsystem. However, because no disturbances act on the subsystems the state information only needs to be communicated once. Afterwards each controller can predict the current state of another subsystem exactly based on the known input sequences.

Convergence and Stability Properties

The main advantages of the cooperative distributed MPC algorithm over the distributed MPC based on robust optimization presented in Chapter 3 are that arbitrary coupling graphs can be considered, no consistency constraints are required, and the controllers do not have to solve robust model predictive control problems. While the consistency constraints used in the Chapter 3 only bound the increase in the cost, the convex combination in (4.4) ensures that the cost is decreasing in the iterations p. Given a feasible initial guess, (4.4) ensures that all iterates are feasible even in the presence of coupling constraints and, if no coupling constraints are present, as $p \to \infty$, the cost converges to the centralized optimum $V^*(x_k)$ (cf. [104]). In [104] an extension to ensure convergence in the presence of coupling constraints is proposed. Specifically, each controller $\mathcal{C}^i$ optimizes over all the inputs interconnected with $\mathbf{u}^i_{k,p}$ by a constraint. This procedure can be interpreted as changing the decomposition of the global input vector into local input vectors such that the resulting constraints are decoupled. While this idea can be directly combined with the results presented in this thesis, it potentially results in much more complex local optimization problems.

As discussed in Chapter 3, the consistency constraints also complicate the stability analysis. This is not the case for Algorithm 4.1. In fact, it was shown in [104] that for Case 2.2 an initialization $\mathbf{u}_{k+1,0}$ for the next time step $k+1$ can be obtained based on $\mathbf{u}_{k,\bar{p}_k}$, where $\bar{p}_k$ denotes the last iteration performed in time k, which is stabilizing for system (2.2). It therefore follows that, in this case, the algorithm can be stopped at any time (k,p) and still exponentially stabilizes the overall system.

On the other hand, a major drawback of the cooperative distributed MPC described in Algorithm 4.1 are its communication requirements. Specifically, all controllers communicate with all other controllers in every iteration, i.e. the edges of the communication graph $\mathscr{C}_{k,p}$ are given by $\mathcal{E}_{k,p} := (\mathcal{N} \times \mathcal{N}) \setminus \cup_{i\in\mathcal{N}}(i,i)$ for all (k,p). Obviously, this results in a very high load on the communication network. However, since no communication links are excluded a-priori the cooperative distributed MPC algorithm is a good starting point to investigate when and between which controllers communication is required.

Another issue is that no convergence rate is given in [112] and the subsequent works. Consequently it is not clear how the weights $w^i_{k,p}$ or the communicated information affect the convergence of the scheme. Finally, there are no results on how to choose the threshold $\epsilon \in \mathbb{R}_{>0}$ in the stopping criterion such that the suboptimality is bounded, or a good trade-off between suboptimality and the number of messages is obtained. Clearly, these last two points are problematic both from a theoretical and practical point of view.

The next section is concerned with investigating the convergence properties of the cooperative distributed MPC, analyzing how the convergence depends on the strength of coupling between subsystems, and computing weights $w^i_{k,p}$ which ensure fast convergence. Specifically, optimized time-invariant weights are compared to weights obtained in each iteration by online optimization. Based on these results, cooperative distributed MPC algorithms with event-based communication are derived in the following chapters.

4.2. Convergence Rate of Cooperative Distributed MPC

In order to analyze the convergence properties of Algorithm 4.1, the following centralized optimization problem with $\mathbf{U}(x_k) := \mathbf{U}^1(x_k) \times \ldots \times \mathbf{U}^{N_s}(x_k)$ is considered:

$$\mathbf{u}_k^* = \arg\min_{\mathbf{u}_k \in \mathbf{U}(x_k)} V(x_k, \mathbf{u}_k). \tag{4.6}$$

Because of $R \succ 0$ it holds that $V(x_k, \mathbf{u}_k)$ is strongly convex and the optimizer $\mathbf{u}_k^*$ exists and is unique (cf. Theorem 2.2). Next, let $V_d(x_k, \mathbf{u}_{k,p})$ denote the difference between the cost at the current iterate and the centralized optimum:

$$V_d(x_k, \mathbf{u}_{k,p}) := V(x_k, \mathbf{u}_{k,p}) - V(x_k, \mathbf{u}_k^*). \tag{4.7}$$

It should be noted that the centralized optimizer $\mathbf{u}_k^*$ is not known before convergence of the distributed MPC algorithm and is only used here to investigate how fast the algorithm converges to $V(x_k, \mathbf{u}_k^*)$. The following two preliminary results are concerned with bounding the gradient of $V(x_k, \mathbf{u}_{k,p})$ in terms of $V_d(x_k, \mathbf{u}_{k,p})$, and providing a measure of the coupling induced between local optimization problems by the Hessian H of (4.2), which will be used to analyze the convergence of the algorithm and to derive optimized weights $w^i_{k,p}$.

To analyze the coupling in the Hessian H, note that the Hessian H can be rearranged to obtain

$$(\mathbf{u}_{k,p})^T H \mathbf{u}_{k,p} = \begin{bmatrix} \mathbf{u}^i_{k,p} \\ \mathbf{u}^{\setminus i}_{k,p} \end{bmatrix}^T \begin{bmatrix} H^i & H^i_c \\ (H^i_c)^T & H^{\setminus i} \end{bmatrix} \begin{bmatrix} \mathbf{u}^i_{k,p} \\ \mathbf{u}^{\setminus i}_{k,p} \end{bmatrix}, \tag{4.8}$$

where $H^i \in \mathbb{R}^{Nm\times Nm}$ and $\mathbf{u}^i_{k,p}$ are the local part of the Hessian and the input sequence of $\mathcal{P}^i$, $H^{\setminus i} \in \mathbb{R}^{N(m-m^i)\times N(m-m^i)}$ is the non-local part of the Hessian, and $\mathbf{u}^{\setminus i}_{k,p} \in \mathbb{R}^{N(m-m^i)}$ contains the inputs of all subsystem $\mathcal{P}^j, j \in \mathcal{N} \setminus i$. In other words $\mathbf{u}^{\setminus i}_{k,p}$ is given by

$$\mathbf{u}^{\setminus i}_{k,p} := (\mathbf{u}^1_{k,p}; \dots ; \mathbf{u}^{i-1}_{k,p}; \mathbf{u}^{i+1}_{k,p}; \dots ; \mathbf{u}^{N_s}_{k,p}).$$

Finally, $H^i_c \in \mathbb{R}^{Nm\times N(m-m^i)}$ is the coupling between $\mathbf{u}^i_{k,p}$ and the matrices are directly obtained by rearranging the rows and columns of H accordingly.

Proposition 4.1. *[46] For all $i \in \mathcal{N}$ there exists a constant $c^i_H \geq 1$ given by*

$$c^i_H = \min_{c^i} \quad c^i \tag{4.9}$$

$$s.t. \quad c^i(H^i - H^i_c(H^{\setminus i})^{-1}(H^i_c)^T) \succeq H^i \tag{4.10}$$

Proof. Applying the Schur complement [15] to (4.8) and considering $H \succ 0$ it directly follows that $H^i \succ 0$, $H^{\setminus i} \succ 0$ and $H^i - H^i_c(H^{\setminus i})^{-1}(H^i_c)^T \succ 0$. It follows that there exists $c^i \geq 0$ such that (4.10) holds. Because of $H^i_c(H^{\setminus i})^{-1}(H^i_c)^T \succeq 0$ it holds that $H^i - H^i_c(H^{\setminus i})^{-1}(H^i_c)^T \preceq H^i$, and it follows that $c^i_H \geq 1$. □

The value c^i_H may be interpreted as the strength of coupling between the local optimization problems, e.g. $H^i_c = 0$ (i.e. no coupling in the Hessian) implies $c^i_H = 1$, while $c^i_H \to \infty$ as the strength of coupling increases. In Figure 4.1 the multipliers c^i_H, $i = \{1, 2\}$ and level sets $u^T H(a_H)u \leq 1$ are shown for

$$\mathbf{u}_k = \begin{bmatrix} u^1_k \\ u^2_k \end{bmatrix}, \quad H(a_H) = \begin{bmatrix} 1 & a_H \\ a_H & 2 \end{bmatrix}, \quad a_H \in \left[0, \sqrt{(2)}\right). \tag{4.11}$$

The level sets and $H(a_H)$ illustrate that the value c^i_H can be interpreted as the strength of coupling, i.e. for $a_H = 0$ one obtains $c^i_H = 1$ and $H(a_H)$ is a diagonal matrix. Increasing a_H results in an increase in c^i_H and, at the same time, $(\mathbf{u}_k)^T H(a_H)\mathbf{u}_k$ more strongly depends on the cost term that contains both u^1_k and u^2_k.

Next, let $\Delta\mathbf{u}_{k,p} := \mathbf{u}^*_k - \mathbf{u}_{k,p}$ denote the difference between the global optimizer and current iterate, and let $\Delta\mathbf{u}^i_{k,p} := (\mathbf{0}; \dots ; \mathbf{u}^{i*}_k - \mathbf{u}^i_{k,p}; \dots ; \mathbf{0}) \in \mathbb{R}^{Nm}$ denote the difference to the global optimizer for controller $\mathcal{C}^i$ and consider the following multipliers $\bar{\mu}^i_{k,p}$ defined by

$$\bar{\mu}^i_{k,p} := \frac{\nabla V(x_k, \mathbf{u}_{k,p})^T \Delta\mathbf{u}^i_{k,p}}{\nabla V(x_k, \mathbf{u}_{k,p})^T \Delta\mathbf{u}_{k,p}}, \quad \text{if } \Delta\mathbf{u}_{k,p} \neq 0, \tag{4.12}$$

and any $0 \leq \bar{\mu}^i_{k,p} \leq 1$ with $\sum_{i\in\mathcal{N}} \bar{\mu}^i_{k,p} = 1$ if $\Delta\mathbf{u}_{k,p} = 0$. The values involved are shown in Figure 4.2, and it can be seen that $\nabla V(x_k, \mathbf{u}_{k,p})^T \Delta\mathbf{u}^i_{k,p}$ is not necessarily smaller than zero. However, due to strong convexity of $V(x_k, \mathbf{u}_{k,p})$ it holds that $V(x_k, \mathbf{u}^*_k) > V(x_k, \mathbf{u}_{k,p}) + \nabla V(x_k, \mathbf{u}_{k,p})^T \Delta\mathbf{u}_{k,p}$ if $\Delta\mathbf{u}_{k,p} \neq 0$ (cf. Definition 2.4), and it directly follows from $V(x_k, \mathbf{u}^*_k) - V(x_k, \mathbf{u}_{k,p}) < 0$ for $\Delta\mathbf{u}_{k,p} \neq 0$ that $\nabla V(x_k, \mathbf{u}_{k,p})^T \Delta\mathbf{u}_{k,p} < 0$ if $\Delta\mathbf{u}_{k,p} \neq 0$.

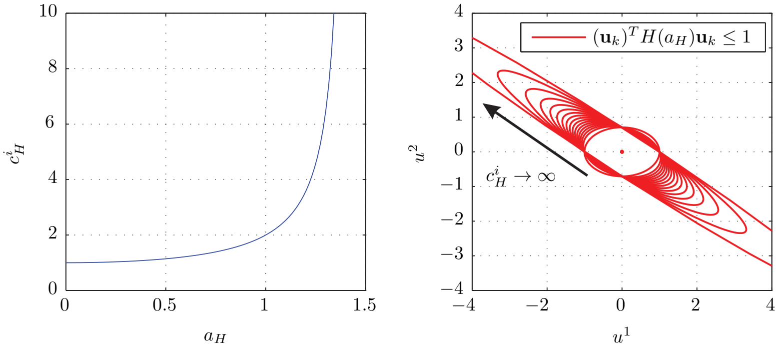

Figure 4.1.: Multipliers c_H^i and level sets of $(\mathbf{u}_k)^T H(a_H)\mathbf{u}_k \leq 1$ for $H(a_H)$ according to (4.11).

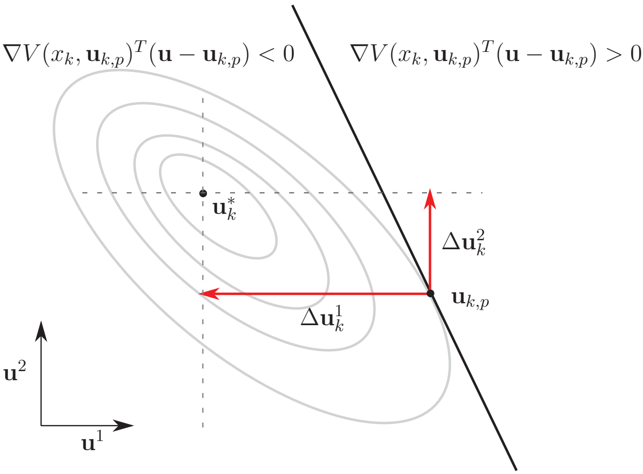

Figure 4.2.: Level sets of $V(x_k, \mathbf{u}_{k,p})$ (gray), sets $\nabla V(x_k, \mathbf{u}_{k,p})^T(\mathbf{u} - \mathbf{u}_{k,p}) > 0$, $\nabla V(x_k, \mathbf{u}_{k,p})^T(\mathbf{u} - \mathbf{u}_{k,p}) < 0$, and the distances $\Delta\mathbf{u}_{k,p}^i$ to the optimizer.

Proposition 4.2. *[46] Given $\bar{\mu}_{k,p}^i$ according to (4.12) it holds that $\sum_{i\in\mathcal{N}} \bar{\mu}_{k,p}^i = 1$, and for all $i \in \mathcal{N}$ with $\bar{\mu}_{k,p}^i \geq 0$ it holds that*

$$\nabla V(x_k, \mathbf{u}_{k,p})^T \Delta\mathbf{u}_{k,p}^i \leq -\bar{\mu}_{k,p}^i V_d(x_k, \mathbf{u}_{k,p}). \tag{4.13}$$

Proof. Considering

$$\textstyle\sum_{i\in\mathcal{N}} \nabla V(x_k, \mathbf{u}_{k,p})^T \Delta \mathbf{u}_{k,p}^i = \nabla V(x_k, \mathbf{u}_{k,p})^T \Delta \mathbf{u}_{k,p},$$

it can be seen that $\sum_{i\in\mathcal{N}} \bar{\mu}_{k,p}^i = 1$. Next, applying Taylor's Theorem to $V(x_k, \mathbf{u}_{k,p})$ developed in $\mathbf{u}_{k,p}$ and noting that $\nabla^2 V(x_k, \mathbf{u}_{k,p}) = 2H$ results in

$$V(x_k, \mathbf{u}_{k,p}^*) = V(x_k, \mathbf{u}_{k,p}) + \nabla V(x_k, \mathbf{u}_{k,p})^T (\mathbf{u}_k^* - \mathbf{u}_{k,p}) + (\mathbf{u}_k^* - \mathbf{u}_{k,p})^T H (\mathbf{u}_k^* - \mathbf{u}_{k,p}),$$

and it directly follows that

$$\nabla V(x_k, \mathbf{u}_{k,p})^T (\mathbf{u}_k^* - \mathbf{u}_{k,p}) = -V_d(x_k, \mathbf{u}_{k,p}) - (\mathbf{u}_k^* - \mathbf{u}_{k,p})^T H (\mathbf{u}_k^* - \mathbf{u}_{k,p}). \tag{4.14}$$

Substituting (4.14) into (4.12) results in

$$\nabla V(x_k, \mathbf{u}_{k,p})^T \Delta \mathbf{u}_{k,p}^i = -\bar{\mu}_{k,p}^i (V_d(x_k, \mathbf{u}_{k,p}) + (\mathbf{u}_k^* - \mathbf{u}_{k,p})^T H (\mathbf{u}_k^* - \mathbf{u}_{k,p})), \tag{4.15}$$

and considering $(\Delta \mathbf{u}_{k,p})^T H (\Delta \mathbf{u}_{k,p})^T \geq 0$ and $\bar{\mu}_{k,p}^i \geq 0$ inequality (4.13) directly follows. □

Proposition 4.3. *[46] Given c_H^i according to* (4.10) *it holds that for every state x_k, and every input iterate $\mathbf{u}_{k,p}$ there exists multipliers*

$$\mu_{k,p}^i \geq 0, \quad \textstyle\sum_{i\in\mathcal{N}} \mu_{k,p}^i = 1,$$

such that the bound

$$\frac{(\mu_{k,p}^i)^2}{4c_H^i} V_d(x_k, \mathbf{u}_{k,p}) \leq V(x_k, \mathbf{u}_{k,p}) - V(x_k, \bar{\mathbf{u}}_{k,p}^i) \tag{4.16}$$

holds.

Proof. Applying Taylor's Theorem to $V_d(x_k, \mathbf{u}_{k,p})$ developed in $\mathbf{u}_{k,p}^*$ results in

$$V_d(x_k, \mathbf{u}_{k,p}) = \nabla V(x_k, \mathbf{u}_k^*)^T (\mathbf{u}_{k,p} - \mathbf{u}_k^*) + \frac{1}{2} (\mathbf{u}_{k,p} - \mathbf{u}_k^*)^T \nabla^2 V(x_k, \mathbf{u}_k^*) (\mathbf{u}_{k,p} - \mathbf{u}_k^*)$$

Theorem 2.3 and optimality of $\mathbf{u}_k^*$ imply that $\nabla V(x_k, \mathbf{u}_k^*)^T (\mathbf{u}_{k,p} - \mathbf{u}_k^*) \geq 0$, and it directly follows that

$$V_d(x_k, \mathbf{u}_{k,p}) \geq (\Delta \mathbf{u}_{k,p})^T H \Delta \mathbf{u}_{k,p} \tag{4.17}$$

holds. Next, H can be rearranged and partitioned as in (4.8) to obtain

$$\begin{aligned} V_d(x_k, \mathbf{u}_{k,p}) &\geq \inf_{\mathbf{u}_{k,p}^{\setminus i} \in \mathbb{U}^{\setminus i}(x_k)} \begin{bmatrix} \Delta \mathbf{u}_{k,p}^i \\ \Delta \mathbf{u}_{k,p}^{\setminus i} \end{bmatrix}^T \begin{bmatrix} H^i & H_c^i \\ (H_c^i)^T & H^{\setminus i} \end{bmatrix} \begin{bmatrix} \Delta \mathbf{u}_{k,p}^i \\ \Delta \mathbf{u}_{k,p}^{\setminus i} \end{bmatrix}, \\ &\geq \inf_{\Delta \mathbf{u}_{k,p}^{\setminus i}} \begin{bmatrix} \Delta \mathbf{u}_{k,p}^i \\ \Delta \mathbf{u}_{k,p}^{\setminus i} \end{bmatrix}^T \begin{bmatrix} H^i & H_c^i \\ (H_c^i)^T & H^{\setminus i} \end{bmatrix} \begin{bmatrix} \Delta \mathbf{u}_{k,p}^i \\ \Delta \mathbf{u}_{k,p}^{\setminus i} \end{bmatrix}, \end{aligned}$$

where $\Delta\mathbf{u}_{k,p}^{\setminus i} \in \mathbb{R}^{N(m-m^i)}$ contains the difference to the optimum for all controllers $j \neq i$. With the minimizer $\Delta\mathbf{u}_{k,p}^{\setminus i} = -(H^{\setminus i})^{-1}(H_c^i)\Delta\mathbf{u}_{k,p}^i$ it holds that

$$V_d(x_k, \mathbf{u}_{k,p}) \geq (\Delta\mathbf{u}_{k,p}^i)^T \left(H^i - H_c^i(H^{\setminus i})^{-1}(H_c^i)^T\right) \Delta\mathbf{u}_{k,p}^i.$$

Considering Proposition (4.1) it holds that

$$c_H^i V_d(x_k, \mathbf{u}_{k,p}) \geq (\Delta\mathbf{u}_{k,p}^i)^T H^i \Delta\mathbf{u}_{k,p}^i. \tag{4.18}$$

Next, a candidate input sequence $\boldsymbol{\vartheta}_{k,p}^i$ is parametrized by $\boldsymbol{\vartheta}_{k,p}^i = \mathbf{u}_{k,p}^i + \theta_{k,p}^i(\mathbf{u}_k^{*i} - \mathbf{u}_{k,p}^i)$ with a step size $\theta_{k,p}^i \in [0,1]$. Because of $\mathbf{u}_{k,p}^i \in \mathbf{U}^i(x_k^i)$ and $\mathbf{u}_k^{*i} \in \mathbf{U}^i(x_k^i)$ it holds that $\boldsymbol{\vartheta}_{k,p}^i \in \mathbf{U}^i(x_k^i)$. Applying Taylor's Theorem to $V(x_k, \mathbf{u}_k)$ results in

$$V(x_k, \tilde{\boldsymbol{\vartheta}}_{k,p}^i) = V(x_k, \mathbf{u}_{k,p}) + \theta_{k,p}^i \nabla V(x_k, \mathbf{u}_{k,p})^T \Delta\mathbf{u}_{k,p}^i + \frac{(\theta_{k,p}^i)^2}{2}(\Delta\mathbf{u}_{k,p}^i)^T H \Delta\mathbf{u}_{k,p}^i, \tag{4.19}$$

where $\tilde{\boldsymbol{\vartheta}}_{k,p}^i := (\mathbf{u}_{k,p}^1, \ldots, \boldsymbol{\vartheta}_{k,p}^i, \ldots, \mathbf{u}_{k,p}^{N_s})$. Considering (4.18) and (4.13), it holds for all $i \in \mathcal{N}$ with $\bar{\mu}_{k,p}^i \geq 0$ that

$$V(x_k, \tilde{\boldsymbol{\vartheta}}_{k,p}^i) \leq V(x_k, \mathbf{u}_{k,p}) + \left((\theta_{k,p}^i)^2 c_H^i - \theta_{k,p}^i \bar{\mu}_{k,p}^i\right) V_d(x_k, \mathbf{u}_{k,p}). \tag{4.20}$$

Based on this, the following step size $\theta_{k,p}^i$ is chosen:

$$\theta_{k,p}^i := \begin{cases} 0 & \text{if } \bar{\mu}_{k,p}^i < 0, \\ \frac{\bar{\mu}_{k,p}^i}{2c_H^i} & \text{if } 0 \leq \bar{\mu}_{k,p}^i \leq 1, \\ \frac{1}{2c_H^i} & \text{if } \bar{\mu}_{k,p}^i > 1. \end{cases} \tag{4.21}$$

Substituting $\theta_{k,p}^i$ into (4.19) and (4.20) it can be verified that

$$V(x_k, \tilde{\boldsymbol{\vartheta}}_{k,p}^i) \leq V(x_k, \mathbf{u}_{k,p}) - \frac{(\check{\mu}_{k,p}^i)^2}{4c_H^i} V_d(x_k, \mathbf{u}_{k,p})$$

holds with the following multipliers:

$$\check{\mu}_{k,p}^i := \begin{cases} 0 & \text{if } \bar{\mu}_{k,p}^i < 0, \\ \bar{\mu}_{k,p}^i & \text{if } 0 \leq \bar{\mu}_{k,p}^i \leq 1, \\ 1 & \text{if } \bar{\mu}_{k,p}^i > 1, \end{cases} \tag{4.22}$$

where the last case follows from $(\frac{1}{2c_H^i})^2 c_H^i - \frac{\bar{\mu}_{k,p}^i}{2c_H^i} \leq -\frac{1}{4c_H^i}$ for all $\bar{\mu}_{k,p}^i > 1$. Finally, with $\mu_{k,p}^i := \frac{\check{\mu}_{k,p}^i}{\sum_{i \in \mathcal{N}} \check{\mu}_{k,p}^i}$ and $\sum_{i \in \mathcal{N}} \check{\mu}_{k,p}^i \geq 1$ it follows that $(\mu_{k,p}^i)^2 \leq (\check{\mu}_{k,p}^i)^2$, and it holds that

$$V(x_k, \tilde{\boldsymbol{\vartheta}}_{k,p}^i) \leq V(x_k, \mathbf{u}_{k,p}) - \frac{(\mu_{k,p}^i)^2}{4c_H^i} V_d(x_k, \mathbf{u}_{k,p}).$$

Considering $V(x_k, \bar{\mathbf{u}}_{k,p}^i) \leq V(x_k, \tilde{\boldsymbol{\vartheta}}_{k,p}^i)$, the proposition follows. □

This result implies that if $V_d(x_k, \mathbf{u}_{k,p}) > 0$, there always exist local candidate input sequences $\bar{\mathbf{u}}^i_{k,p}$ which will result in a decrease of the global cost. Furthermore, it can be seen that the definition of the step size $\theta^i_{k,p}$ and multipliers $\mu^i_{k,p}$ may lead to some conservatism in this formulation if there exists $i \in \mathcal{N}$ such that $\bar{\mu}^i_{k,p} > 1$. However, there exist initial conditions where $\mu^i_{k,p} = \bar{\mu}^i_{k,p}$ holds, i.e. in general no further conservatism is introduced. The next results states that a bound on the convergence rate can be established which depends on the weights $w^i_{k,p}$, the coupling strength c^i_H, and on the multipliers $\mu^i_{k,p}$.

Theorem 4.1. *[44] The difference to the optimal cost $V_d(x_k, \mathbf{u}_{k,p})$ converges according to*

$$V_d(x_k, \mathbf{u}_{k,p+1}) \leq \beta_{k,p} V_d(x_k, \mathbf{u}_{k,p}), \tag{4.23}$$

with rate

$$0 \leq \beta_{k,p} \leq 1 - \frac{1}{4} \sum\nolimits_{i\in\mathcal{N}} w^i_{k,p} \frac{(\mu^i_{k,p})^2}{c^i_H} \tag{4.24}$$

and it holds that $\beta_{k,p} < 1$ for all $w^i_{k,p} > 1$ and $\sum_{i\in\mathcal{N}} w^i_{k,p} = 1$.

Proof. Considering the update of the inputs (4.4), the cost function at $(k, p+1)$ is given by

$$V(x_k, \mathbf{u}_{k,p+1}) = V\left(x_k, \sum\nolimits_{i\in\mathcal{N}} w^i_{k,p} \bar{\mathbf{u}}^i_{k,p}\right). \tag{4.25}$$

It follows from convexity of $V(x_k, \mathbf{u}_{k,p+1})$ that

$$V(x_k, \mathbf{u}_{k,p+1}) \leq \sum\nolimits_{i\in\mathcal{N}} w^i_{k,p} V(x_k, \bar{\mathbf{u}}^i_{k,p}). \tag{4.26}$$

Considering Proposition 4.3, the following upper bound on the optimal cost of the local minimization problem (4.3) is obtained:

$$V(x_k, \bar{\mathbf{u}}^i_{k,p}) \leq V(x_k, \mathbf{u}_{k,p}) - \frac{(\mu^i_{k,p})^2}{4c^i_H} V_d(x_k, \mathbf{u}_{k,p}). \tag{4.27}$$

Finally, substituting (4.27) back into (4.26) yields

$$\begin{aligned} V(x_k, \mathbf{u}_{k,p+1}) &\leq \sum_{i\in\mathcal{N}} w^i_{k,p} V(x_k, \mathbf{u}_{k,p}) - w^i_{k,p} \frac{(\mu^i_{k,p})^2}{4c^i_H} V_d(x_k, \mathbf{u}_{k,p}) \\ &\leq V(x_k, \mathbf{u}_{k,p}) - \sum_{i\in\mathcal{N}} w^i_{k,p} \frac{(\mu^i_{k,p})^2}{4c^i_H} V_d(x_k, \mathbf{u}_{k,p}) \end{aligned}$$

Subtracting $V(x_k, \mathbf{u}^*_{k|k})$ on both sides results in (4.23) and

$$\beta_{k,p} \leq 1 - \sum_{i\in\mathcal{N}} w^i_{k,p} \frac{(\mu^i_{k,p})^2}{4c^i_H}.$$

Furthermore, for any $\mu^i_{k,p} \geq 0$, $\sum_{i\in\mathcal{N}} \mu^i_{k,p} = 1$ and any choice of $w^i_{k,p} > 0$, $\sum_{i\in\mathcal{N}} w^i_{k,p} = 1$, it directly follows that $\beta_{k,p} < 1$. □

It can be seen that a smaller $\beta_{k,p}$ implies faster convergence and that $\beta_{k,p}$ may strongly depend on the choice of weights $w^i_{k,p}$, which may be time-varying.

4.3. Optimized Parameters for Fast Convergence

In the work on the cooperative distributed MPC Algorithm 4.1 no results are given on how to choose the weights $w^i_{k,p}$ to achieve fast convergence. Based on the convergence rate derived in the previous section, the impact of the weights on the convergence of the overall algorithm can be analyzed, and optimized weights can be derived. In this section, bounds on the convergence rate $\beta_{k,p}$ for different choices of $w^i_{k,p}$ are discussed. Specifically, time-invariant weights and weights which are optimized online are considered.

4.3.1. Optimized Time-Invariant Weights

The convergence rate $\beta_{k,p}$ can be bounded by $\beta_{k,p} \leq \bar{\beta}$:

$$\bar{\beta} = \min_{w^i_{k,p}} \max_{\mu^i_{k,p}} 1 - \frac{1}{4} \sum\nolimits_{i\in\mathcal{N}} w^i_{k,p} \frac{(\mu^i_{k,p})^2}{c^i_H}, \tag{4.28}$$

i.e. $\bar{\beta}$ is guaranteed a-priori for all $\mu^i_{k,p}$ (worst-case). In the following problem (4.28) will first be solved for time-invariant weights w^i, i.e. time-invariant weights which minimize the bound $\beta_{k,p}$ given in Theorem 4.1 are computed. Subsequently, it is investigated if optimizing the weights in each time step and iteration results in an improved bound.

Theorem 4.2. *[44] The optimal solution of* (4.28) *is given by* $w^i_{k,p} = \sqrt{c^i_H} c_r^{-1}$ *with* $c_r = \sum_{i\in\mathcal{N}} \sqrt{c^i_H}$. *The resulting bound on the convergence rate is*

$$\bar{\beta} = 1 - \frac{1}{4} \left(\sum\nolimits_{i\in\mathcal{N}} \sqrt{c^i_H} \right)^{-2}. \tag{4.29}$$

Proof. Let $\mu_{k,p} := (\mu^1_{k,p}; \ldots ; \mu^{N_s}_{k,p})$ and $w_{k,p} := (w^1_{k,p}; \ldots ; w^{N_s}_{k,p})$ denote the vectors of multipliers $\mu^i_{k,p}$ and weights $w^i_{k,p}$. Instead of solving the inner maximization problem in (4.28), the maximizer can be computed by solving the following problem:

$$\min_{\mu_{k,p}} \sum\nolimits_{i\in\mathcal{N}} w^i_{k,p} \frac{(\mu^i_{k,p})^2}{c^i_H}, \text{ s.t. } \sum\nolimits_{i\in\mathcal{N}} \mu^i_{k,p} = 1,\ \mu_{k,p} \geq 0. \tag{4.30}$$

The corresponding KKT conditions (cf. Section 4.3.2) with multipliers λ_i and ν are given by

$$\frac{2w_{k,p}^i}{c_H^i}\mu_{k,p}^i - \lambda_i + \nu = 0, \quad \forall i \in \{1,\dots,N_s\} \tag{4.31}$$

$$\lambda_i \geq 0, \quad \forall i \in \{1,\dots,N_s\} \tag{4.32}$$

$$\lambda_i(-\mu_{k,p}^i) = 0, \quad \forall i \in \{1,\dots,N_s\} \tag{4.33}$$

$$\mu_{k,p}^i \geq 0, \quad \forall i \in \{1,\dots,N_s\} \tag{4.34}$$

$$\textstyle\sum_{i\in\mathcal{N}} \mu_{k,p}^i = 1. \tag{4.35}$$

It can be verified that the KKT conditions hold for $\nu = -2c_z$, $\lambda_i = 0$, and

$$\mu_{k,p}^i = c_z \frac{c_H^i}{w_{k,p}^i}, \quad c_z = \left(\textstyle\sum_{i\in\mathcal{N}} \frac{c_H^i}{w_{k,p}^i}\right)^{-1}.$$

Since the problem is convex, satisfaction of the KKT conditions is sufficient to establish optimality. Therefore, the parametric minimizer is given by $\mu_{k,p}^i(w_{k,p}^i, c_H^i) = c_z \frac{c_H^i}{w_{k,p}^i}$. Substituting $\mu_{k,p}^i = c_z \frac{c_H^i}{w_{k,p}^i}$ into (4.28) results in

$$\bar{\beta} = \min_{w_{k,p}} 1 - \frac{1}{4} z^2 \textstyle\sum_{i\in\mathcal{N}} \frac{c_H^i}{w_{k,p}^i}, \tag{4.36}$$

$$\bar{\beta} = \min_{w_{k,p}} 1 - \left(4 \textstyle\sum_{i\in\mathcal{N}} \frac{c_H^i}{w_{k,p}^i}\right)^{-1}. \tag{4.37}$$

It remains to find weights which solve this minimization problem. The minimizer $w_{k,p}^*$ of (4.37) is identical to

$$w_{k,p}^* = \arg\min_{w_{k,p}} \textstyle\sum_{i\in\mathcal{N}} \frac{c_H^i}{w_{k,p}^i}, \text{s.t. } \sum_{i\in\mathcal{N}} w_{k,p}^i = 1,\ w_{k,p} > 0. \tag{4.38}$$

The solution to this minimization problem can again be computed explicitly by means of the KKT conditions given by

$$-\frac{c_H^i}{(w_{k,p}^i)^2} - \lambda_i + \nu = 0, \quad \forall i \in \{1,\dots,N_s\} \tag{4.39}$$

$$\lambda_i \geq 0, \quad \forall i \in \{1,\dots,N_s\} \tag{4.40}$$

$$\lambda_i(-w_{k,p}^i) = 0, \quad \forall i \in \{1,\dots,N_s\} \tag{4.41}$$

$$w_{k,p}^i > 0, \quad \forall i \in \{1,\dots,N_s\} \tag{4.42}$$

$$\textstyle\sum_{i\in\mathcal{N}} w_{k,p}^i = 1. \tag{4.43}$$

The KKT conditions are satisfied for $w_{k,p}^i = \sqrt{c_H^i} c_r^{-1}$ with $c_r = \sum_{i\in\mathcal{N}} \sqrt{c_H^i}$, $\lambda_i = 0$ and $\nu = c_r^2$. Substituting into (4.37) gives (4.29). □

4.3.2. Online Optimization of Weights

To improve performance, one may optimize over $w^i_{k,p}$ online after exchanging the planned input of each subsystem, which also allows to relax the condition $w^i_{k,p} > 0$ to $w^i_{k,p} \geq 0$. However, optimizing the weights online requires either additional computation by all controllers or additional information exchange, even if the subproblems are fully decoupled. In such an implementation, weights $w^i_{k,p}$ are optimized for given candidate input sequences $\bar{\mathbf{u}}^i_{k,p}$ in every iteration:

$$w^*_{k,p} = \arg\min_{w_{k,p}} V\left(x_k, \textstyle\sum_{i\in\mathcal{N}} w^i_{k,p}\bar{\mathbf{u}}^i_{k|k,p}\right). \tag{4.44}$$

For the time invariant weights the condition $w^i_{k,p} > 0$ is required to ensure convergence for all initial conditions. If the optimal weights are chosen in each iteration depending on the initial condition the constraint $w^i_{k,p} > 0$ can be relaxed to $w^i_{k,p} \geq 0$. The following result states that optimizing the weights at each time (k,p) in general does not result in an improved bound $\bar{\beta}$.

Theorem 4.3. *[44] When optimizing the weights $w^i_{k,p}$ in every iteration, the convergence rate $\beta_{k,p}$ is bounded by:*

$$\beta_{k,p} \leq 1 - \frac{1}{4}\left(\textstyle\sum_{i\in\mathcal{N}} \sqrt{c^i_H}\right)^{-2} = \bar{\beta},$$

for all $p \geq 0$.

Proof. In terms of the convergence rate $\beta_{k,p}$, optimizing the weights online results in the following problem:

$$\min_{w_{k,p}} 1 - \frac{1}{4}\sum_{i\in\mathcal{N}} w^i_{k,p}\frac{(\mu^i_{k,p})^2}{c^i_H}, \text{ s.t. } \textstyle\sum_{i\in\mathcal{N}} w^i_{k,p} = 1,\ w^i_{k,p} \geq 0,$$

which can be used to compute the parametric minimizer $w^i_{k,p}(\mu^i_{k,p}, c^i_H)$, i.e. weights which depend on the current multipliers $\mu^i_{k,p}$, for all $i \in \mathcal{N}$. The corresponding KKT conditions are given by

$$-\frac{(\mu^i_{k,p})^2}{c^i_H} - \lambda_i + \nu = 0, \quad \forall i \in \{1,\ldots,N_s\} \tag{4.45}$$

$$\lambda_i(-w^i_{k,p}) = 0, \quad \forall i \in \{1,\ldots,N_s\} \tag{4.46}$$

$$w^i_{k,p} \geq 0, \quad \forall i \in \{1,\ldots,N_s\} \tag{4.47}$$

$$\textstyle\sum_{i\in\mathcal{N}} w^i_{k,p} = 1. \tag{4.48}$$

These conditions hold for $j^*_{k,p} = \arg\max_i (\mu^i_{k,p})^2(c^i_H)^{-1}$, $w^{j^*_{k,p}} = 1$, $\lambda_{j^*_{k,p}} = 0$, and

$$\nu = \left(\mu^{j^*_{k,p}}_{k,p}\right)^2\left(c^{j^*_{k,p}}_H\right)^{-1}, \quad \lambda_s = -(\mu^s_{k,p})^2(c^s_H)^{-1} + \left(\mu^{j^*_{k,p}}_{k,p}\right)^2\left(c^{j^*_{k,p}}_H\right)^{-1}.$$

Since the problem is convex, satisfaction of the KKT conditions is sufficient to establish optimality. Substituting into (4.28) yields

$$\bar{\beta} = 1 - \frac{1}{4} \min_{\mu_{k,p}} \max_{i} \frac{(\mu^i_{k,p})^2}{c^i_H}, \text{ s.t. } \textstyle\sum_{i\in\mathcal{N}} \mu^i_{k,p} = 1, \mu^i_{k,p} \geq 0. \tag{4.49}$$

Next, let υ^i denote a variable used to parameterize feasible solutions $\mu^i_{k,p}(\upsilon^i) = \sqrt{c^i_H} c_r^{-1} + \upsilon^i$, for all $\sum_i \upsilon^i = 0$. For $\upsilon^i = 0$ for all $i \in \mathcal{N}$ it follows that

$$(\mu^1_{k,p}(0))^2 (c^1_H)^{-1} = \ldots = (\mu^{N_s}_{k,p}(0))^2 (c^{N_s}_H)^{-1} = c_r^{-2} > 0.$$

This implies:

$$\max\left\{\frac{(\mu^1_{k,p}(0))^2}{c^1_H}, \ldots, \frac{(\mu^{N_s}_{k,p}(0))^2}{c^{N_s}_H}\right\} \leq \max\left\{\frac{(\mu^1_{k,p}(\upsilon^1))^2}{c^1_H}, \ldots, \frac{(\mu^{N_s}_{k,p}(\upsilon^{N_s}))^2}{c^{N_s}_H}\right\}.$$

In other words, $\mu^i_{k,p} = \sqrt{c^i_H} c_r^{-1}$ is the optimal solution. Substituting into (4.49) results in (4.29). □

This bound can be slightly improved if more than one iteration is performed per time-step. In this scenario, the convergence rate of the weights using online optimization is given by:

Theorem 4.4. *When optimizing the weights $w^i_{k,p}$ in every iteration, the convergence rate $\beta_{k,p}$ for $p > 1$ is bounded by:*

$$\beta_{k,p} \leq 1 - \frac{1}{4} \left(\textstyle\sum_{i\in\mathcal{N}\setminus j^{\min}} \sqrt{c^i_H}\right)^{-2}, \tag{4.50}$$

where $j^{\min} = \arg\min_{i\in\mathcal{N}} \sqrt{c^i_H}$.

Proof. Convexity of $V(x_k, \mathbf{u}_{k,p})$ as well as optimality of $\mathbf{u}^*_k$ imply that

$$\nabla V(x_k, \mathbf{u}_{k,p})^T \Delta \mathbf{u}_{k,p} \leq 0. \tag{4.51}$$

On the other hand if $j^*_{k,p-1} = i$ it holds that $w^i_{k,p} = 1$ and optimality of $\mathbf{u}^i_{k,p} = \boldsymbol{\rho}^i_{k,p-1}$ with respect to problem (4.3) implies that

$$\nabla V(x_k, \mathbf{u}_{k,p})^T (\mathbf{0}_{n^1\times 1}; \ldots; \mathbf{u}^{i*}_k - \boldsymbol{\rho}^i_{k,p-1}; \ldots; \mathbf{0}_{n^{N_s}\times 1}) \geq 0, \tag{4.52}$$

and it follows that $\bar{\mu}^i_{k,p} \leq 0$ and $\mu^i_{k,p} = 0$. Using this additional constraint in (4.49) and the same arguments as in the proof of the previous theorem, results in

$$\beta_{k,p} \leq 1 - \frac{1}{4} \left(\textstyle\sum_{i\in\mathcal{N}\setminus j^*_{k,p}} \sqrt{c^i_H}\right)^{-2}. \tag{4.53}$$

With $-\left(\sum_{i\in\mathcal{N}\setminus j^*_{k,p}} \sqrt{c^i_H}\right)^{-2} \leq -\left(\sum_{i\in\mathcal{N}\setminus j^{\min}} \sqrt{c^i_H}\right)^{-2}$ the theorem follows. □

This result implies that if more than one iteration is performed the convergence may be improved by optimizing the weights online. If the algorithm is applied in closed-loop with the threshold $\epsilon > 0$, strict convexity of $V(x_k, \mathbf{u}_{k,p})$ implies that $V_d(x_k, \mathbf{u}_{k,p}) > 0$ and it follows that there exists $j \in \mathcal{N}$ such that $\mu^j_{k,p} > 0$. Because of $\mu^i_{k,p} = 0$ for $i = j^*_{k,p-1}$ it holds that $j^*_{k,p} \neq j^*_{k,p-1}$. Therefore, Theorem 4.4 implies that a sequential algorithm in which only one controller updates its input sequence in each iteration, and no controller updates its input sequence in two consecutive iterations, guarantees faster convergence than using the time-invariant weights given in Theorem 4.2 if more than one iteration is performed.

However, this sequence is not known a-priori and finding this sequence either requires knowledge of the values of the multipliers $\mu^i_{k,p}$, which are unknown even online because they depend on $\mathbf{u}^*_k$, or requires the solution of the problem (4.44) restricted to weights that are either 0 or 1. This requires the solutions of all local optimization problems (4.3) and is a non-convex problem. It follows that such a sequence can only be obtained after all local problems have been solved and the solutions have been exchanged between all controllers.

Contrary to this, in typical sequential algorithms the sequence is fixed a-priori and computation and communication are performed according to this sequence. However, the best sequence cannot be chosen a-priori and, in the worst case, only offers slightly faster (or as fast if $p_{\max} = 1$) convergence than using time-invariant weights. Furthermore, fixing any sequence a-priori may, without further assumptions, result in extremely slow convergence. For instance, if the sequence does not contain all subsystems, a sequential algorithm may not progress towards the optimum at all. This suggests that in a sequential algorithm with a-priori fixed sequence at least N_s iterations need to be performed per time-step to guarantee any cost improvement. In particular, choosing any weight $w^i_{k,p} = 1$ a-priori may not result in any convergence, for instance if $V(x_k, \mathbf{u}_{k,p}) = V(x_k, \bar{\mathbf{u}}^i_{k,p})$.

Benefits of Online Optimization

The results in the previous section show that optimizing the weights $w^i_{k,p}$ online does, in general, not result in an improved bound $\bar{\beta}$. However, the convergence speed may be improved by online optimization of the weights $w^i_{k,p}$ if the state x_k and input sequence $\mathbf{u}_{k,p}$, which are encoded by the multipliers $\mu^i_{k,p}$, allow for fast convergence. This is made precise in the following two theorems, in which the bound on the convergence rate is minimized with respect to $\mu^i_{k,p}$.

Theorem 4.5. *[44] Given time-invariant weights $w^i_{k,p} = \sqrt{c^i_H} c_r^{-1}$, the multipliers $\mu^{j^*}_{k,p} = 1$ with*

$$j^* = \arg\max_{j \in \mathcal{N}} \left(c_r \sqrt{c^j_H} \right)^{-1},$$

and $\mu^j_{k,p} = 0$ *for all* $j \in \mathcal{N} \setminus j^*$ *minimize* $\bar{\beta}$ *and the minimum is given by*

$$\bar{\beta}_{\min} = 1 - \frac{1}{4}\left(c_r \min_{i\in\mathcal{N}} \sqrt{c^i_H}\right)^{-1}. \tag{4.54}$$

Proof. Substituting $w^i_{k,p} = \sqrt{c^i_H} c_r^{-1}$ into (4.24) and minimizing the bound $\bar{\beta}$ for the multipliers $\mu^i_{k,p}$ instead of maximizing results in

$$\mu^*_{k,p} = \arg\max_{\mu_{k,p}} \sum_{i\in\mathcal{N}} \frac{(\mu^i_{k,p})^2}{c_r\sqrt{c^i_H}}, \text{ s.t. } \textstyle\sum_{i\in\mathcal{N}} \mu^i_{k,p} = 1, \mu_{k,p} \geq 0, \tag{4.55}$$

i.e. the maximization of a convex function, which is a non-convex problem. If the feasible set is a bounded polytope, the optimal value is attained at a vertex of the feasible set[1]. Only the vertices of the box $0 \leq \mu^i_{k,p} \leq 1$ that also lie in the subspace $\sum_{i\in\mathcal{N}} \mu^i_{k,p} = 1$ are vertices of the feasible set of (4.55). It directly follows that the vertices are given by vectors $\mathfrak{v}_j = (\mathfrak{v}^1_j, \ldots, \mathfrak{v}^i_j, \ldots, \mathfrak{v}^{N_s}_j) \in \mathbb{R}^{N_s}$ and $\mathfrak{v}^i_j$ is defined as follows for all $j \in \mathcal{N}$:

$$\mathfrak{v}^i_j = \begin{cases} 0 & \text{if } i \neq j, \\ 1 & \text{otherwise.} \end{cases}$$

Therefore, the maximum is given by $\mu^{j^*}_{k,p} = 1$ with $j^* = \arg\max_{i\in\mathcal{N}} \left(c_r\sqrt{c^i_H}\right)^{-1}$ and substituting into (4.55) results in (4.54). □

Similarly, one can minimize the bound on the convergence rate in the case of online optimization of the weights.

Theorem 4.6. *[44] If the weights are optimized online, the bound* $\bar{\beta}$ *is minimized by* $\mu^{j_m}_{k,p} = 1$ *with* $j_m = \arg\min_{i\in\mathcal{N}} c^i_H$ *and* $\mu^i_{k,p} = 0$ *for all* $i \in \mathcal{N} \setminus j_m$*. The minimum is given by*

$$\bar{\beta}_{\min} = 1 - \frac{1}{4}\left(\min_{i\in\mathcal{N}} c^i_H\right)^{-1}. \tag{4.56}$$

Proof. In the proof of Theorem 4.3 it was shown that the optimal weights are $w_s = 0$, for all $s \in \mathcal{N} \setminus j^*$, $\lambda_{j^*} = 0$, $w_{j^*} = 1$ and $j^* = \arg\max_i (\mu^i_{k,p})^2 (c^i_H)^{-1}$ (cf. (4.45)-(4.48)). Substituting into (4.24) and minimizing for the multipliers $\mu^i_{k,p}$ results in

$$\bar{\beta}_{\min} = 1 - \frac{1}{4} \max_{\mu^i_{k,p}} \max_i \frac{(\mu^i_{k,p})^2}{c^i_H} \text{ s.t. } \textstyle\sum_{i\in\mathcal{N}} \mu^i_{k,p} = 1, \mu_{k,p} \geq 0.$$

It can be seen, that the optimal solution is given by $\mu^{j_m}_{k,p} = 1$ with $j_m = \arg\min_i c^i_H$. Substituting into (4.24) completes the proof. □

[1] Convexity of the cost function implies that at any point $\bar{\mathfrak{v}}$ on a line segment between two vertices $\mathfrak{v}_i$ and $\mathfrak{v}_j$ the cost is lower or equal than at $\mathfrak{v}_i$ and $\mathfrak{v}_j$.

4.4. Numerical Example

To compare the convergence rates obtained in the previous sections to the actual convergence rate, the distributed MPC given in Algorithm 4.1 is applied to distributed MPC problems with decoupled dynamics, local constraints, and potentially fully coupled costs using the time-invariant weights derived in Theorem 4.2 and the optimized weights given in (4.44). Specifically, the subsystems are modeled by double-integrator dynamics

$$x_{k+1}^i = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} x_k^i + \begin{bmatrix} 0.5 \\ 1 \end{bmatrix} u_k^i,$$

with local input and state constraints $u_k^i \in [-1, 1]$, $x_k^i \in [-1, 1] \times [-1, 1]$, and randomly generated fully coupled costs $V(x_k, \mathbf{u}_{k,p})$. Note that regardless of which case is considered the quadratic program solved by the cooperative distributed MPC always has this structure and, more importantly, any randomly generated coupling by costs or dynamics will result in a cost function with the same properties as the ones randomly generated here. Thus, this comparison is valid for all three cases considered in this section.

To evaluate the influence of the number of subsystems, Algorithm 4.1 was used to solve problems ranging from $N_s = 2$ to $N_s = 25$ subsystems. For each value of N_s (i.e. each number of subsystems) 40 problems were generated based on 10 random cost functions and 4 random initial conditions. The distributed MPC algorithm was initialized with a decentralized solution, i.e. by ignoring all interconnections, and 20 iterations were performed for each problem. The actual rate of convergence $\beta_{k,p} = \frac{V_d(x_k, \mathbf{u}_{k|k,p+1})}{V_d(x_k, \mathbf{u}_{k|k,p})}$ was computed and compared to the bound $\bar{\beta}$ given in the previous sections. In Table 4.1 the mean, minimal and maximal ratio (taken over all random examples and iterations) between the actual rate of convergence $\gamma_{k,p}$ and bound $\bar{\beta}$ is given.

These results highlight that the difference between the actual convergence rate and bound $\bar{\beta}$ strongly depends on the number of subsystems. For instance, for $N_s = 2$ the algorithm converges approximately twice as fast as indicated by Theorem 4.2. However, as the number of subsystems increases, the actual convergence approaches the bound $\bar{\beta}$. This could be expected because for a large number of subsystems the weights $w_{k,p}^i$ of individual subsystems become small and may limit the progress of the algorithm per iteration. Furthermore, it can be seen that optimizing the weights online on average only results in slightly faster convergence for the random examples considered here.

4.5. Discussion

The results in the previous sections reveal a strong connection between the weights $w_{k,p}^i$, the strength of the coupling between subsystems indicated by c_H^i, and the

N_s	Optimized time-invariant weights			Online optimization of weights		
	max	mean	min	max	mean	min
2	2.335	2.072	1.944	20.98	2.519	1.944
3	1.533	1.418	1.294	1.665	1.446	1.322
4	1.354	1.322	1.298	1.446	1.362	1.317
5	1.263	1.225	1.174	1.299	1.242	1.177
7	1.176	1.135	1.099	1.191	1.142	1.101
10	1.121	1.101	1.071	1.128	1.105	1.073
15	1.066	1.059	1.052	1.073	1.060	1.053
20	1.051	1.043	1.034	1.058	1.044	1.035
25	1.044	1.039	1.030	1.049	1.040	1.030

Table 4.1.: Ratio $\frac{\bar{\beta}}{\beta_{k,p}}$ between the bound $\bar{\beta}$ and actual convergence rate $\beta_{k,p}$.

convergence rate $\beta_{k,p}$. In particular, due to the summation over $c_H^i > 1$ in (4.29) the bounds on the convergence rate $\bar{\beta}$ directly depends on the number of subsystems N_s. This could be expected, because the condition $\sum_{i \in \mathcal{N}} w_{k,p}^i = 1$ results in small weights $w_{k,p}^i$ if N_s is large. In turn, this implies that the progress of each of the local input variable updates (4.5) becomes small if N_s is large.

With respect to optimizing the weights online, it can be seen that, in general, this does not result in improved a-priori guarantees for the convergence rate compared to the optimized time-invariant weights. However, faster convergence may be achieved by optimizing $w_{k,p}^i$ online depending on the number of iterations and initial conditions. For instance, if the inputs of only a few controllers are far away from the optimum, the convergence of the distributed MPC algorithm may no longer depend on the number of subsystems when optimizing the weights online. This is made precise in Theorem 4.6.

It has to be noted that the results for the best case scenarios are somewhat conservative, especially in the case of Theorem 4.6. Assuming that there exists a subsystem j with $c_H^j = 1$ (i.e. no coupling), then $\beta_{k,p} \leq 0.75$. At the same time, if only the input of this subsystem is suboptimal, then the cost will actually converge after one step. The best weight for this case, $w_{k,p}^j = 1$ and $w_{k,p}^s = 0$ for all $s \neq j$, is correctly identified in the proofs of Theorem 4.3 and Theorem 4.6. However, the possible improvement of the cost by controller $\mathcal{C}^j$ in this case is underestimated through (4.27) by a factor of 4. Thus, the reason for the discrepancy in the convergence rate in this case are the bounds established in the proof of Proposition 4.3.

Overall it can be seen that the communication requirements of this algorithm may be problematic because communication between all controllers is required in every iteration and every time step. Nonetheless, the cooperative distributed MPC with full communication is a good starting point for an event-based algorithm since it can be analyzed how the communicated information affects convergence and therefore control performance. Furthermore, because the communication topology is not restricted a-priori as in [103], this algorithm allows to communicate between any pair of controllers if necessary. While there is no robustness with respect to uncertain communication, a possible solution to compensate for communication delays $\tau_k \ll \Delta t$ is to only optimize the input sequence for $u_{k+1|k}$ to $u_{k+N-1|k}$, and to use the additional constraint $u_{k|k} = u_{k|k-1}$ to ensure consistent application of the control inputs.

Results on closed-loop application of the cooperative distributed MPC algorithm with full communication can be readily obtained from the more general results given in Section 5.3, where the stability properties of this algorithm in conjunction with event-based communication are analyzed.

The results in this chapter imply that the primal cost converges at least with $\mathcal{O}(c^p)$ and $0 < c < 1$. Recently accelerated gradient methods and dual decomposition have been used to solve mixed $\mathscr{L}_1/\mathscr{L}_2$-problems, which result from MPC problems with sparse coupling by costs, constraints and dynamics [36]. These algorithms achieve a convergence rate of $\mathcal{O}\left(\frac{1}{p^2}\right)$, with iteration number p, for the dual cost and the distance between the current primal iterate and optimal primal solution. However, the implications for the primal cost before the termination of the algorithm are not directly clear, because the dual cost does not provide any insight into the suboptimality of the associated primal solution. In particular, the primal cost may increase from one iteration to the next, which may be problematic if the optimization has to be terminated before the optimum is reached. This is relevant because, in practice, closed-loop distributed MPC is almost always terminated early due to the time constraints and communication constraints. Some results concerned with early stopping of distributed MPC based on dual decomposition have been developed recently and can be found in [34]. However, these results only apply to distributed MPC without terminal constraint, the number of iterations may be quite large, and the impact of coupling between the subsystems as well as communication between the controllers is hard to analyze. In contrast, the algorithm analyzed in this chapter can be terminated at any iteration if a suitable terminal constraint is used (see e.g. [104] and Section 5.3). Furthermore, as shown in the next chapters, the methodology presented in this chapter can be used to analyze how communication affects convergence and to derive triggering rules for communication and distributed stopping criteria.

Part III.

DMPC with Event-Based Communication

5. Event-Based Communication Based on Cost Decrease

In the cooperative distributed MPC algorithm discussed in the previous chapter every controller communicates with every other controller in each iteration. This results in a large number of messages which may not be required. Specifically, often the local solution does not change much between iterations, or the resulting change in the cost function is small. In this chapter, a triggering condition for communication is presented which eliminates communication in these cases and can be directly used as a distributed stopping criterion for the iterative cooperative distributed MPC algorithm. Subsequently, the convergence rate and suboptimality of the algorithm with event-based communication are analyzed by using the framework presented in the previous chapter. The results on event-based communication, convergence and suboptimality have been submitted for publication in [46]. The stability properties of the closed-loop are analyzed using the stability results on input-to-state stability derived in Section 2.3.

5.1. Communication Events and Distributed MPC Algorithm

The cooperative distributed MPC with event-based communication relies on the same local optimization used in the cooperative distributed MPC with full time-triggered communication and throughout this chapter it is assumed that Assumption 4.1 holds. The difference between the two algorithms lies in the update of the variables, which involves communication.

The generator for a communication event considered in this chapter is obtained by locally comparing the cost $V(x_k, \mathbf{u}_{k,p})$ before and the cost $V(x_k, \bar{\mathbf{u}}^i_{k,p})$ after the local optimization (4.3). Specifically, if the difference $V(x_k, \mathbf{u}_{k,p}) - V(x_k, \bar{\mathbf{u}}^i_{k,p})$ is small, this indicates that the local optimization of $\mathcal{C}^i$ did not significantly improve the closed-loop performance. In this case, the result of the local optimization is not communicated and is discarded. This results in the following triggering mechanism for communication:

$$\mathcal{T}_{k,p} := \left\{ i \in \mathcal{N} \,\middle|\, V(x_k, \mathbf{u}_{k,p}) - V(x_k, \bar{\mathbf{u}}^i_{k,p}) > \gamma_i \right\}, \tag{5.1}$$

where $\mathcal{T}_{k,p}$ denotes the index set of controllers which communicate at time (k, p), and $\gamma_i \in \mathbb{R}_{>0}$ is the threshold for triggering a communication event. Specifically, if a communication event is triggered for $\mathcal{C}^i$ the result of the local optimization is communicated to all other controllers. This results in a time-varying communication graph $\mathscr{C}_{k,p}$ given by $\mathcal{E}_{k,p} := (\mathcal{T}_{k,p} \times \mathcal{N}) \setminus \cup_{i\in\mathcal{N}}(i,i)$.

Next, the iteration (4.4) is modified such that only the candidate input sequences $\bar{\mathbf{u}}^i_{k,p}$ of controllers $\mathcal{C}^i$, $i \in \mathcal{T}_{k,p}$ which communicated in time (k,p) are considered, i.e. if $\mathcal{T}_{k,p} \neq \emptyset$ the iteration (4.4) becomes:

$$\mathbf{u}_{k,p+1} := \sum\nolimits_{i\in\mathcal{T}_{k,p}} w^i_{k,p}\bar{\mathbf{u}}^i_{k,p}, \tag{5.2}$$

and if $\mathcal{T}_{k,p} = \emptyset$ the iterative distributed MPC scheme with event-based communication given in Algorithm 5.1 terminates. As discussed above, local updates which do not sufficiently improve the cost are not communicated and discarded. The stopping criterion $\mathcal{T}_{k,p} = \emptyset$ can be readily checked by every controller by checking if any data was received.

Assumption 5.1. *It is assumed that the weights $w^i_{k,p} \in \mathbb{R}_{>0}$ are chosen such that $\sum_{i\in\mathcal{T}_{k,p}} w^i_{k,p} = 1$ holds if $\mathcal{T}_{k,p} \neq \emptyset$.*

The following result establishes that the cost is strictly decreasing if communication takes place. It will be used to bound the number of iterations required for the stopping criterion $\mathcal{T}_{k,p} = \emptyset$ to hold.

Proposition 5.1. *[46] For each iteration, the cost $V(x_k, \mathbf{u}_{k,p})$ is decreasing and decreases by at least $\gamma^{\min} := \min_{i\in\mathcal{N}} \gamma^i$ if, and only if, $\mathcal{T}_{k,p} \neq \emptyset$.*

Proof. Considering (5.2) and $\mathcal{T}_{k,p} \neq \emptyset$, the cost function at iterate $p+1$ is given by

$$V(x_k, \mathbf{u}_{k,p+1}) = V\left(x_k, \sum\nolimits_{i\in\mathcal{T}_{k,p}} w^i_{k,p}\bar{\mathbf{u}}^i_{k,p}\right). \tag{5.3}$$

It follows from convexity of $V(x_k, \mathbf{u}_{k,p+1})$ that

$$\begin{aligned} V(x_k, \mathbf{u}_{k,p+1}) &\leq \sum\nolimits_{i\in\mathcal{T}_{k,p}} w^i_{k,p} V(x_k, \bar{\mathbf{u}}^i_{k,p}), \\ V(x_k, \mathbf{u}_{k,p+1}) &\leq V(x_k, \mathbf{u}_{k,p}) - \sum\nolimits_{i\in\mathcal{T}_{k,p}} w^i_{k,p}\left(V(x_k, \mathbf{u}_{k,p}) - V(x_k, \bar{\mathbf{u}}^i_{k,p})\right). \end{aligned} \tag{5.4}$$

Next, based on the triggering condition in (5.1) it can be verified that

$$V(x_k, \mathbf{u}_{k,p+1}) \leq V(x_k, \mathbf{u}^i_{k,p}) - \sum\nolimits_{i\in\mathcal{T}_{k,p}} w^i_{k,p}\gamma^i \tag{5.5}$$

Finally, considering Assumption 5.1 and $\gamma^{\min} > 0$ it directly follows that

$$V(x_k, \mathbf{u}_{k,p+1}) \leq V(x_k, \mathbf{u}^i_{k,p}) - \gamma^{\min}, \tag{5.6}$$

if $\mathcal{T}_{k,p} \neq \emptyset$. If $\mathcal{T}_{k,p} = \emptyset$, the algorithm terminates with $u_k = u_{k|k,p}$. □

The next result investigates sufficient conditions for the number of iterations required such that the stopping criterion $\mathcal{T}_{k,p} = \emptyset$ holds.

Algorithm 5.1: Event-based distributed MPC iterations in k for each $\mathcal{C}^i$

Given k, x_k, $\mathbf{u}_{k,0}$, $p = 0$, $p_{\max}$, and γ_i:
while $p \leq p_{\max}$ **do**
    Solve (4.3) and check if $i \in \mathcal{T}_{k,p}$ according to (5.1)
    If $i \in \mathcal{T}_{k,p}$ communicate $\boldsymbol{\rho}^i_{k,p}$ to all controllers
    Construct $\mathcal{T}_{k,p}$ based on the received information
    if $\mathcal{T}_{k,p} = \emptyset$ **then**
        break
    else
        Compute $\mathbf{u}_{k,p+1}$ according to (5.2) and set $p := p + 1$
    end if
end while
Apply $u^i_{k|k,p}$ to the system (4.1)

Theorem 5.1. *[46] If $V_d(x_k, \mathbf{u}_{k,0}) \leq \gamma^{\min}$ it holds that $\Omega_{k,p} = \emptyset$. If $V_d(x_k, \mathbf{u}_{k,0}) \geq \gamma^{\min}$, there exists $p \leq \tilde{p}_k$ with*

$$\tilde{p}_k = \lfloor (\gamma^{\min})^{-1} V_d(x_k, \mathbf{u}_{k,0}) \rfloor,$$

such that the stopping condition $\Omega_{k,p} = \emptyset$ holds.

Proof. Because of $V(x_k, \mathbf{u}^*_k) \leq V(x_k, \bar{\mathbf{u}}^i_{k,\tilde{p}_k})$, it directly follows that $V_d(x_k, \mathbf{u}_{k,p}) < \gamma^{\min}$ implies $V(x_k, \mathbf{u}_{k,p}) - V(x_k, \bar{\mathbf{u}}^i_{k,p}) < \gamma^{\min}$ and $\Omega_{k,p} = \emptyset$.

Next, assume that for $V_d(x_k, \mathbf{u}_{k,0}) \geq \gamma^{\min}$ no number of iterations $p \leq \tilde{p}_k - 1$ exists such that $\Omega_{k,p} = \emptyset$ holds. By Proposition 5.1 the cost decreases by $\gamma^{\min} > 0$ in each iteration if $\Omega_{k,p} \neq \emptyset$. Therefore, after $\tilde{p}_k$ iterations it holds that $V_d(x_k, \mathbf{u}_{k,\tilde{p}_k}) \leq \gamma^{min}$, which implies $\Omega_{k,\tilde{p}_k} = \emptyset$, and the theorem follows. ☐

This result shows that the algorithm terminates after a finite number of iterations and the stopping condition $\mathcal{T}_{k,p} = \emptyset$ may hold before reaching the global optimum. However, the stopping condition may hold earlier then $V_d(x_k, \mathbf{u}_{k,p}) < \gamma^{\min}$, i.e. Theorem 5.1 gives a sufficient condition for the number of iterations p such that $\mathcal{T}_{k,p} = \emptyset$ holds.

Another important aspect is how to chose the thresholds γ^i and the suboptimality of the input sequence when the algorithm terminates. From the results of Theorem 5.1 it can be seen that smaller thresholds γ^i result in more iterations. Intuitively this should also result in a solution with smaller suboptimality. This is made precise in the following theorem which shows that the thresholds γ^i can be chosen to guarantee a bound $V_d(x_k, \mathbf{u}_{k,p}) \leq \bar{V}_d$ on the suboptimality at termination.

Theorem 5.2. *[46] Given a bound $\bar{V}_d > 0$ and c^i_H according to Proposition 4.1, a finite number of iterations $\tilde{p}_k$ exists such that the event-based scheme with thresholds*

$\gamma^i = \frac{\bar{V}_d}{4{N_s}^2 c_H^i}$ *terminates after* $p_k \leq \tilde{p}_k$ *iterations with*

$$\tilde{p}_k = \left\lfloor 4N_s^2 c_H^{\max} \frac{V_d(x_k, \mathbf{u}_{k,0})}{\bar{V}_d} \right\rfloor, \tag{5.7}$$

and $V_d(x_k, \mathbf{u}_{k,p_k}) \leq \bar{V}_d$.

Proof. Theorem 5.1 shows that there exists $p_k \leq \tilde{p}_k$ such that $\mathcal{T}_{k,p_k} = \emptyset$ holds. Together with (5.1) this implies $V(x_k,\mathbf{u}_{k,p_k}) - V(x_k,\bar{\mathbf{u}}^i_{k,p_k}) < \gamma^i$. With the local thresholds $\gamma^i := \frac{\bar{V}_d {N_s}^{-2}}{4c_H^i}$ it follows from Proposition 4.3 that

$$\frac{(\mu^i_{k,p_k})^2}{4c_H^i} V_d(x_k, \mathbf{u}_{k,p_k}) \leq \frac{\bar{V}_d {N_s}^{-2}}{4c_H^i}, \quad \forall i \in \mathcal{N}. \tag{5.8}$$

Summation over all $i \in \mathcal{N}$ gives:

$$\textstyle\sum_{i\in\mathcal{N}} (\mu^i_{k,p_k})^2 V_d(x_k, \mathbf{u}_{k,p_k}) \leq \sum_{i\in\mathcal{N}} \bar{V}_d {N_s}^{-2}, \tag{5.9}$$

and it directly follows that

$$V_d(x_k, \mathbf{u}_{k,p_k}) \leq \frac{\bar{V}_d {N_s}^{-1}}{\sum_{i\in\mathcal{N}} (\mu^i_{k,p_k})^2}. \tag{5.10}$$

Therefore, an upper bound on $V_d(x_k, \mathbf{u}_{k,p_k})$ can be obtained by maximizing the right hand side of (5.10) with respect to $\mu^i_{k,p}$:

$$\min_{\mu^i_{k,p}} \textstyle\sum_{i\in\mathcal{N}} (\mu^i_{k,p})^2, \text{ s.t. } \ \mu^i_{k,p} > 0, \sum_{i\in\mathcal{N}} \mu^i_{k,p} = 1. \tag{5.11}$$

The corresponding KKT conditions with multipliers λ_i for inequality constraints, and the multiplier ν for the equality constraints are given by

$$2\mu^i_{k,p} - \lambda_i + \nu = 0, \quad \forall i \in \{1, \dots, N_s\} \tag{5.12}$$

$$\lambda_i \geq 0, \quad \forall i \in \{1, \dots, N_s\} \tag{5.13}$$

$$\lambda_i(-\mu^i_{k,p}) = 0, \quad \forall i \in \{1, \dots, N_s\} \tag{5.14}$$

$$\mu^i_{k,p} \geq 0, \quad \forall i \in \{1, \dots, N_s\} \tag{5.15}$$

$$\textstyle\sum_{i\in\mathcal{N}} \mu^i_{k,p} = 1. \tag{5.16}$$

It can be verified that the KKT conditions hold for $\mu^i_{k,p} = {N_s}^{-1}$, $\lambda_i = 0$ and $\nu = -2{N_s}^{-1}$. Since the problem is convex, the KKT conditions are sufficient for optimality. Substituting into (5.10) results in $V_d(x_k, \mathbf{u}_{k,p_k}) \leq \bar{V}_d$, and the theorem follows. Finally, substituting $\gamma^{\min} = \frac{\bar{V}_d {N_s}^{-2}}{4c_H^{\max}}$ into the results of Theorem 5.1 results in (5.7). □

Depending on the suboptimality $V_d(x_k, \mathbf{u}_{k,0})$ at initialization, three different cases have to be considered.

Theorem 5.3. *[46] The cost difference $V_d(x_k, \mathbf{u}_{k,p})$ converges*

(I) to

$$V_d(x_k, \mathbf{u}_{k,p_k}) \leq \bar{V}_d$$

if $\bar{V}_d \leq V_d(x_k, \mathbf{u}_{k,0})$ and $\mathcal{T}_{k,p_k} = \emptyset$,

(II) to

$$V_d(x_k, \mathbf{u}_{k,p_k}) \leq V_d(x_k, \mathbf{u}_{k,0})$$

if $\frac{\bar{V}_d N_s{}^{-2}}{4c_H^{\max}} \leq V_d(x_k, \mathbf{u}_{k,0}) \leq \bar{V}_d$ and $\mathcal{T}_{k,p_k} = \emptyset$,

(III) and for $V_d(x_k, \mathbf{u}_{k,0}) \leq \frac{\bar{V}_d N_s{}^{-2}}{4c_H^{\max}}$ the algorithm terminates immediately with $p_k = 0$.

Proof. (I) directly follows from Theorem 5.1, Theorem 5.2, and $\gamma^{\min} = \frac{\bar{V}_d N_s{}^{-2}}{4c_H^{\max}}$, where $c_H^{\max} := \max_{i\in\mathcal{N}} c_H^i \geq 1$. To show (II), note that by Proposition 5.1, the cost is decreasing. Finally (III) directly follows from Theorem 5.1. □

5.2. Rate of Convergence and Optimized Weights

In order to draw comparisons to cooperative distributed MPC with full communication it is of interest how the event-based communication and choice of weights $w_{k,p}^i$ affects the convergence rate of the algorithm. The convergence rate can be computed based on the framework presented in the previous chapter. Theorem 5.3 implies that the algorithm may terminate if $V_d(x_k, \mathbf{u}_{k,p}) \leq \bar{V}_d$. Thus, a rate of convergence can only be provided for the case $V_d(x_k, \mathbf{u}_{k,p}) > \bar{V}_d$.

Theorem 5.4. *If $V_d(x_k, \mathbf{u}_{k,p}) > \bar{V}_d$ the difference to the optimal cost $V_d(x_k, \mathbf{u}_{k,p})$ converges according to*

$$V_d(x_k, \mathbf{u}_{k,p+1}) \leq \beta_{k,p} V_d(x_k, \mathbf{u}_{k,p}), \tag{5.17}$$

with the convergence rate

$$\beta_{k,p} = 1 - \frac{1}{4} \sum\nolimits_{i\in\mathcal{T}_{k,p}} w_{k,p}^i \frac{(\mu_{k,p}^i)^2}{c_H^i}. \tag{5.18}$$

Proof. By substituting (4.16) into (5.4) and considering $\sum_{i\in\mathcal{T}_{k,p}} w^i_{k,p} = 1$, it holds that

$$V(x_k,\mathbf{u}_{k,p+1}) \leq V(x_k,\mathbf{u}_{k,p}) - \sum_{i\in\mathcal{T}_{k,p}} w^i_{k,p} \frac{(\mu^i_{k,p})^2}{4c^i_H} V_d(x_k,\mathbf{u}_{k,p})$$

Subtracting $V(x_k,\mathbf{u}^*_k)$ on both sides results in

$$V_d(x_k,\mathbf{u}_{k,p+1}) \leq \beta_{k,p} V_d(x_k,\mathbf{u}_{k,p+1}), \tag{5.19}$$

with $\beta_{k,p} = \left(1 - \sum_{i\in\mathcal{T}_{k,p}} w^i_{k,p} \frac{(\mu^i_{k,p})^2}{4c^i_H}\right)$. □

In the case of communication between all controllers ($\mathcal{T}_{k,p} = \mathcal{N}$) this becomes (cf. Theorem 4.1)

$$\beta_{k,p} = 1 - \frac{1}{4} \sum_{i\in\mathcal{N}} w^i_{k,p} \frac{(\mu^i_{k,p})^2}{c^i_H}. \tag{5.20}$$

In other words, the convergence of the scheme may be slowed down by the event-based approach because not all controllers participate in the scheme in each iteration p. On the other hand, in the case of full communication the weights $w^i_{k,p}$ have to satisfy $\sum_{i\in\mathcal{N}} w^i_{k,p} = 1$ to ensure that the costs are decreasing, but in the event-based case only $\sum_{i\in\mathcal{T}_{k,p}} w^i_{k,p} = 1$ is required (cf. Proposition 5.1). This may lead to faster convergence.

The following result is concerned with establishing an upper bound $\beta_{k,p} \leq \bar{\beta}_{k,p}$ and corresponding time-invariant weights $w^i_{k,p} := f^i_w(\mathcal{T}_{k,p})$, which minimize the bound.

Theorem 5.5. *[46] If $V_d(x_k,\mathbf{u}_{k,p}) > \bar{V}_d$, the rate of convergence $\beta_{k,p}$ is bounded by:*

$$\beta_{k,p} \leq \bar{\beta}_{k,p} = 1 - \frac{\kappa^2_{k,p}}{4} \left(\sum_{i\in\mathcal{T}_{k,p}} \sqrt{c^i_H}\right)^{-2} < 1, \tag{5.21}$$

with $0 < \kappa_{k,p} \leq 1$. The corresponding weights are given by

$$w^i_{k,p} = \begin{cases} \sqrt{c^i_H} f_{\mathcal{T}}(\mathcal{T}_{k,p})^{-1} & \forall i \in \mathcal{T}_{k,p}, \\ 0 & \forall i \notin \mathcal{T}_{k,p}, \end{cases} \tag{5.22}$$

with $f_{\mathcal{T}}(\mathcal{T}_{k,p}) = \sum_{i\in\mathcal{T}_{k,p}} \sqrt{c^i_H}$.

Proof. Considering Proposition 4.3, the triggering condition (5.1), and $\gamma^i := \frac{\bar{V}_d N_s^{-2}}{4c^i_H}$, inequality (5.8) holds for all $i \in \mathcal{N} \setminus \mathcal{T}_{k,p}$, and it directly follows that

$$\sqrt{{N_s}^{-2} \bar{V}_d V_d(x_k,\mathbf{u}_{k,p})^{-1}} \geq \mu^i_{k,p}, \quad \forall i \in \mathcal{N} \setminus \mathcal{T}_{k,p}. \tag{5.23}$$

Summation over $i \in \mathcal{N} \setminus \mathcal{T}_{k,p}$ gives:

$$N_{k,p}^{nc} {N_s}^{-1} \sqrt{\bar{V}_d V_d(x_k, \mathbf{u}_{k,p})^{-1}} \geq \textstyle\sum_{i \in \mathcal{N} \setminus \mathcal{T}_{k,p}} \mu_{k,p}^i, \tag{5.24}$$

where $N_{k,p}^{nc} := \operatorname{card}(\mathcal{N} \setminus \mathcal{T}_{k,p})$ is the cardinality of $\mathcal{N} \setminus \mathcal{T}_{k,p}$. Next, letting

$$\kappa_{k,p} := 1 - N_{k,p}^{nc} N^{-1} \sqrt{\bar{V}_d V_d(x_k, \mathbf{u}_{k,p})^{-1}},$$

it directly follows from $\sum_{i \in \mathcal{N}} \mu_{k,p}^i = 1$ that

$$\kappa_{k,p} \leq \sum_{i \in \mathcal{N}} \mu_{k,p}^i - \sum_{i \in \mathcal{N} \setminus \mathcal{T}_{k,p}} \mu_{k,p}^i = \sum_{i \in \mathcal{T}_{k,p}} \mu_{k,p}^i, \tag{5.25}$$

and $0 < \kappa_{k,p} \leq 1$ holds for $V_d(x_k, \mathbf{u}_{k,p}) \geq \bar{V}_d$. In order to obtain an upper bound on the convergence rate, one has to solve:

$$\bar{\beta}_{k,p} = \min_{w_{k,p}^i} \max_{\mu_{k,p}^i} 1 - \frac{1}{4} \textstyle\sum_{i \in \mathcal{T}_{k,p}} w_{k,p}^i \frac{(\mu_{k,p}^i)^2}{c_H^i}, \tag{5.26}$$

subject to $\mu_{k,p}^i \geq 0$, $\sum_{i \in \mathcal{T}_{k,p}} \mu_{k,p}^i \geq \kappa_{k,p}$, $w_{k,p}^i \geq 0$, and $\sum_{i \in \mathcal{T}_{k,p}} w_{k,p}^i = 1$. The KKT conditions for the inner maximization, which is parametric with respect to $w_{k,p}^i$ and $\kappa_{k,p}$, are given by

$$\frac{2 w_{k,p}^i}{c_H^i} \mu_{k,p}^i - \lambda_i + \nu = 0, \quad \forall i \in \{1, \ldots, N_s\} \tag{5.27}$$

$$\lambda_i \geq 0, \quad \forall i \in \mathcal{T}_{k,p} \tag{5.28}$$

$$\lambda_i(-\mu_{k,p}^i) = 0, \quad \forall i \in \mathcal{T}_{k,p} \tag{5.29}$$

$$\mu_{k,p}^i \geq 0, \quad \forall i \in \mathcal{T}_{k,p} \tag{5.30}$$

$$\textstyle\sum_{i \in \mathcal{T}_{k,p}} \mu_{k,p}^i = \kappa_{k,p}. \tag{5.31}$$

It can be verified that the KKT conditions hold for $\mu_{k,p}^i = c_z \frac{c_H^i}{w_{k,p}^i} \kappa_{k,p}$ and $c_z = \left(\sum_{i \in \mathcal{T}_{k,p}} \frac{c_H^i}{w_{k,p}^i} \right)^{-1}$, $\nu = -2 c_z \kappa_{k,p}$ and $\lambda_i = 0$. Because the problem is convex, satisfaction of the KKT conditions implies optimality of the solution. Substituting $\mu_{k,p}^i = c_z \frac{c_H^i}{w_{k,p}^i} \kappa_{k,p}$ into (5.26) results in

$$\bar{\beta}_{k,p} = \min_{w_{k,p}^i} 1 - \frac{\kappa_{k,p}^2}{4 \sum_{i \in \mathcal{T}_{k,p}} \frac{c_H^i}{w_{k,p}^i}}, \tag{5.32}$$

$$\bar{\beta}_{k,p} = 1 - \frac{\kappa_{k,p}^2}{4 \min_{w_{k,p}^i} \sum_{i \in \mathcal{T}_{k,p}} \frac{c_H^i}{w_{k,p}^i}}. \tag{5.33}$$

The KKT conditions for this problem are given by

$$-\frac{c_H^i}{(w_{k,p}^i)^2} - \lambda_i + \nu = 0, \quad \forall i \in \mathcal{T}_{k,p} \tag{5.34}$$

$$\lambda_i \geq 0, \quad \forall i \in \mathcal{T}_{k,p} \tag{5.35}$$

$$\lambda_i(-w_{k,p}^i) = 0, \quad \forall i \in \mathcal{T}_{k,p} \tag{5.36}$$

$$w_{k,p}^i > 0, \quad \forall i \in \mathcal{T}_{k,p} \tag{5.37}$$

$$\textstyle\sum_{i\in\mathcal{T}_{k,p}} w_{k,p}^i = 1. \tag{5.38}$$

The KKT conditions hold for $w_{k,p}^i = \sqrt{c_H^i} f_{\mathcal{T}}(\mathcal{T}_{k,p})^{-1}$ with $f_{\mathcal{T}}(\mathcal{T}_{k,p}) = \sum_{i\in\mathcal{T}_{k,p}} \sqrt{c_H^i}$, $\lambda_i = 0$ and $\nu = f_{\mathcal{T}}(\mathcal{T}_{k,p})^2$ for all $i \in \mathcal{T}_{k,p}$, and $w_{k,p}^j = 0$ for $j \in \mathcal{N} \setminus \mathcal{T}_{k,p}$. In other words, the optimal weights in the case of event-based communication depend on the communication topology given by $\mathcal{T}_{k,p}$. Substituting $w_{k,p}^i = \sqrt{c_H^i} f_{\mathcal{T}}(\mathcal{T}_{k,p})^{-1}$ for all $i \in \mathcal{T}_{k,p}$ into (5.32) results in the following bound:

$$\bar{\beta} = 1 - \frac{\kappa_{k,p}^2}{4}\left(\textstyle\sum_{i\in\mathcal{T}_{k,p}} \sqrt{c_H^i}\right)^{-2}. \tag{5.39}$$

□

In contrast to the case of full time-triggered communication, the optimized weights and the corresponding bound on the convergence rate for the case of event-based communication depend on the set of communicating controllers. Furthermore, because of $\kappa \to 0$ as $N_{k,p}^{nc} \to N_s$ and $V_d \to \bar{V}_d$ the convergence of the algorithm with event-based communication slows down close to $\bar{V}_d$. In contrast, the bound on the convergence rate in case of full communication (cf. Theorem 4.2) is time-invariant:

$$\bar{\beta} = 1 - \frac{1}{4}\left(\textstyle\sum_{i\in\mathcal{N}} \sqrt{c_H^i}\right)^{-2}. \tag{5.40}$$

Nonetheless, the convergence of the event-based scheme may be faster. Assuming that $\mathcal{T}_{k,p} \subset \mathcal{N}$, it holds that

$$\left(\textstyle\sum_{i\in\mathcal{T}_{k,p}} \sqrt{c_H^i}\right)^{-2} > \left(\textstyle\sum_{i\in\mathcal{N}} \sqrt{c_H^i}\right)^{-2},$$

which may lead to improved convergence in some cases. This applies, for example, if $c_H^i \gg c_H^j$ for $i \neq j$ and all $j \in \mathcal{N} \setminus i$ and $i \notin \mathcal{T}_{k,p}$, i.e. subproblem i is strongly interconnected but $\bar{\mathbf{u}}_{k,p}^i$ did not significantly improve the cost.

5.3. Stability Analysis

In this section stability of the distributed MPC algorithm with event-based communication in closed-loop with (2.2) is investigated. For Case 2.2 (i.e. coupled dynamics and costs, no state constraints) and Case 2.3 (i.e. decoupled dynamics, coupled costs, decoupled constraints) a terminal constraint is used and asymptotic stability can be established. In Case 2.4 (i.e. coupled dynamics and costs, no state constraints) no terminal constraint or cost are used and only practical stability can be established. In all cases, the ISpS framework presented in Section 2.3 is used to deal with suboptimal solutions. To specify the control objective the set Σ from (2.4), with the additional assumption that Γ_u is chosen such that $R := \Gamma_u^T \Gamma_u \succ 0$, is used. The weight $Q = Q^T \succeq 0$ is again given by $Q := \Gamma_x^T \Gamma_x$.

Stability Using Terminal Costs and Terminal Constraints

Assumption 5.2. *For the cases 2.2 and 2.3, it is assumed that the terminal cost P, terminal set $\mathbb{T}$ (cf. cases 2.2 and 2.3) and terminal controller $u_k = K_{\mathbb{T}} x_k$ are chosen such that the following conditions hold for all $x_k \in \mathbb{T}$:*

1. $K_{\mathbb{T}} x_k \in \mathbb{U}$,
2. $x_{k+1} = (A + BK_{\mathbb{T}})x_k \in \mathbb{T}$,
3. $\|Ax_k + Bu_k\|_P^2 = 0,\ \forall (x_k, u_k) \in \Sigma$
4. $\|Ax_k + BK_{\mathbb{T}} x_k\|_P^2 - \|x_k\|_P^2 \leq -\|x_k\|_Q^2 - \|Kx_k\|_R^2$.

The first two conditions imply feasibility of the terminal control law, the third condition ensures that the terminal cost is zero at the control goal, and the fourth condition encodes that the terminal control law is stabilizing. If $\Sigma = \{0\}$ such a terminal constraint, terminal cost, and controller can always be constructed for Case 2.2 (cf. [104]) and Case 2.3 (see Appendix A.2). In the general case $\Sigma \neq \{0\}$ it may not be possible to satisfy Assumption 5.2 depending on the dynamics and the construction of the set Σ.

Since the cost is decreasing with the iterations p, it is sufficient to show that the initialization $\mathbf{u}_{k+1,0}$ for the next time step stabilizes the system. Let $\bar{p}_k$ with $\bar{p}_k \leq \tilde{p}_k$ denote the number of iterations performed in time k. For the cases 2.2 and 2.3, the terminal control law $K_{\mathbb{T}}$ is used to initialize the algorithm for the next time step, i.e. the following initialization is performed locally by each controller $\mathcal{C}^i$:

$$\mathbf{u}_{k+1,0} := (u_{k+1|k,\bar{p}_k}; \ldots; u_{k+N-1|k,\bar{p}_k}; K_{\mathbb{T}} x_{k+N|k,\bar{p}_k}). \tag{5.41}$$

The following result states that asymptotic stability of the closed-loop can be established for the extended state vector $\mathbf{z}_{k,p} = (z_{k|k,p}; \ldots; z_{k+N-1|k,p})$ with respect to corresponding set $\boldsymbol{\Sigma} := \Sigma \times \ldots \times \Sigma$. To this end, let $z_{k+l|k,p} = (x_{k+l|k,p}; u_{k+l|k,p})$ for all $l \in \{0, \ldots, N-1\}$ denote the states and inputs planned for time $k+l$ at (k,p), and let $\boldsymbol{\mathcal{Z}}$ denote the feasible set of (2.6) for $\mathbf{z}_{k,p}$. Furthermore, $\mathbf{V}_z(\mathbf{z}_{k,p})$ denotes the cost formulated with respect to $\mathbf{z}_{k,p}$, i.e. by substituting the dynamics for $x_{k+N|k}$ such that $\mathbf{V}_z(\mathbf{z}_{k,p}) = V(x_k, \mathbf{u}_{k|k,p})$.

Theorem 5.6. *Given a feasible initialization* $\mathbf{z}_{0,0} \in \boldsymbol{\mathcal{Z}}$*, the following holds for the distributed MPC in closed-loop with* (2.2) *at any iteration* $p \geq 0$ *if Assumption 5.2 holds:*

(I) the distributed MPC problem is feasible for all $k \geq 0$*,*

(II) the closed-loop is asymptotically stable with respect to $\boldsymbol{\Sigma}$*.*

for the cases 2.2 and 2.3.

Proof. To show (I), note that a feasible initialization at $k = 0$ implies $x_{N|0,0} \in \mathbb{T}$. Using this solution as initialization for $k = 1$ according to (5.41) implies $x_{N|1,0} \in \mathbb{T}$. By Assumption 5.2 it holds that $K_{\mathbb{T}} x_{N|1} \in \mathbb{U}$ and $(A + BK_{\mathbb{T}})x_{N|1,0} \in \mathbb{T}$ and it follows that the initialization for $k = 1$ given by (5.41) is feasible. Statement (I) follows by induction over k and noting that feasibility is preserved at any iteration because of the purely local constraints in (4.3).

To establish (II) the case that no iterations take place is considered first, i.e. $\bar{p}_{k+l} = 0$ for all $l \in \mathbb{N}_0$. With the extended state vector $\mathbf{z}_{k,p}$ and considering (5.41) it holds for any $\bar{p}_k$ that

$$\begin{aligned}\mathbf{V}_z(\mathbf{z}_{k+1,0}) - \mathbf{V}_z(\mathbf{z}_{k,\bar{p}_k}) \leq & \|x_{k+N|k+1,0}\|_Q^2 + \|u_{k+N|k+1,0}\|_R^2 + \|x_{k+N+1|k+1,0}\|_P^2 \\ & - \|x_{k|k,\bar{p}_k}\|_Q^2 - \|u_{k|k,\bar{p}_k}\|_R^2 - \|x_{k+N|k,\bar{p}_k}\|_P^2.\end{aligned}$$

Considering Assumption 5.2 it follows that the cost is decreasing:

$$\mathbf{V}_z(\mathbf{z}_{k+1,0}) - \mathbf{V}_z(\mathbf{z}_{k,\bar{p}_k}) \leq -\|x_{k|k,\bar{p}_k}\|_Q^2 - \|u_{k|k,\bar{p}_k}\|_R^2. \tag{5.42}$$

Applying this inequality and the initialization (5.41) for $N-1$ time steps results in

$$\mathbf{V}_z(\mathbf{z}_{k+N,0}) - \mathbf{V}_z(\mathbf{z}_{k,\bar{p}_k}) \leq -\sum_{l=0}^{N-1}(\|x_{k+l|k,\bar{p}_k}\|_Q^2 + \|u_{k+l|k,\bar{p}_k}\|_R^2), \tag{5.43}$$

because applying (5.41) implies that $u_{k+l|k+l,0} = u_{k+l|k,0}$, $x_{k+l|k+l,0} = x_{k+l|k,0}$ for all $l \in \{0, \ldots, N-1\}$. Next, Proposition 5.1 then implies that $\mathbf{V}_z(\mathbf{z}_{k+N,\bar{p}_{k+N}}) \leq \mathbf{V}(\mathbf{z}_{k+N,0})$ and it holds that

$$\mathbf{V}_z(\mathbf{z}_{k+N,\bar{p}_{k+N}}) - \mathbf{V}_z(\mathbf{z}_{k,\bar{p}_k}) \leq -\sum_{l=0}^{N-1}(\|x_{k|k,\bar{p}_k}\|_Q^2 + \|u_{k|k,\bar{p}_k}\|_R^2). \tag{5.44}$$

It directly follows from the construction of Q and R that there exists $\alpha_3(\|\mathbf{z}_{k,\bar{p}_k}\|_{\boldsymbol{\Sigma}})$ such that $\mathbf{V}_z(\mathbf{z}_{k+N,\bar{p}_{k+N}}) - \mathbf{V}_z(\mathbf{z}_{k,\bar{p}_k}) \leq -\alpha_3(\|\mathbf{z}_{k,\bar{p}_k}\|_{\boldsymbol{\Sigma}})$. Furthermore, the definition of the costs (2.5) implies that there exists $\alpha_1(\|\mathbf{z}\|_{\boldsymbol{\Sigma}})$ such that $\alpha_1(\|\mathbf{z}\|_{\boldsymbol{\Sigma}}) \leq \mathbf{V}_z(\mathbf{z})$ holds. Finally, because $\mathbf{V}_z(\mathbf{z})$ is a continuous quadratic function of $\mathbf{z}$ and $\mathbf{V}_z(\mathbf{z}) = 0$ if $\mathbf{z} \in \boldsymbol{\Sigma}$ it follows that there exists $\alpha_2(\|\mathbf{z}\|_{\boldsymbol{\Sigma}})$ such that $\mathbf{V}_z(\mathbf{z}) \leq \alpha_2(\|\mathbf{z}\|_{\boldsymbol{\Sigma}})$.

In other words $\mathbf{V}_z(\mathbf{z}_{k,p})$ can be used as ISpS-Lyapunov function and the conditions of Theorem 2.1 hold for $L = N$, $d_1 = d_2 = 0$, and $\omega_k = 0$ for all $k \in \mathbb{N}_0$ which implies that the closed-loop is asymptotically stable. □

The result given above establishes asymptotic stability with respect to $\boldsymbol{\Sigma}$ for both Case 2.2 and Case 2.3 based on Assumption 5.2. In contrast in [104], the terminal constraint $\mathbb{T}$ has to be choosen such that all unstable modes of the system are zero at the end of the prediction horizon (i.e. Assumption 5.2 holds with $K_{\mathbb{T}} = 0$ and $\Sigma = \{0\}$) and an additional explicit "stability" constraint is used to establish exponential stability.

Stability Without Terminal Costs or Terminal Constraints

In Case 2.4 no terminal constraint or terminal cost is employed. In general, this implies that no terminal control law is known. Thus, the following initialization for $k = k + 1$ will be used:

$$\mathbf{u}^i_{k+1,0} := (u^i_{k+1|k,\bar{p}_k}; \ldots; u^i_{k+N-1|k,\bar{p}_k}; 0). \tag{5.45}$$

In the following it is shown that the system is practically stable by using the cost $V(x_k, \mathbf{u}_{k|k,\bar{p}_k})$ at termination (i.e. $\mathcal{T}_{k,\bar{p}_k} = \emptyset$) as ISpS-Lyapunov function. To this end, we require the following assumption with respect to the optimal cost $V^*(x_k)$ of the centralized MPC (2.6):

Assumption 5.3. *For Case 2.4 it is assumed that the prediction horizon N is chosen such that there exists a set $\mathcal{X} = \{x_k \in \mathbb{R}^n \,|\, V^*(x_k) \leq c_{Lev}\}$ with $c_{Lev} \in \mathbb{R}_{>0}$ and a constant $0 \leq \eta < 1$, such that*

$$V^*(Ax_k + Bu^*(x_k)) - V^*(x_k) \leq -\eta\,(\|x_k\|^2_Q + \|u^*(x_k)\|^2_R), \tag{5.46}$$

and

$$V^*(Ax_k + Bu^*(x_k)) + \bar{V}_d \leq c_{Lev} \tag{5.47}$$

holds for all $x_k \in \mathcal{X}$.

The reader is referred to [49] and [35] for further discussions on how to verify (5.46). The contraction of the level sets of $V^*(x_k)$ due to (5.46) may then be used to establish (5.47), which ensures invariance of $\mathcal{X}$ with respect to the dynamics (2.2) in closed-loop with the suboptimal distributed MPC (cf. [12]). Because the implications of this assumption are not clear if $\Sigma \neq \{0\}$, only the case $\Sigma = \{0\}$ is considered here.

Theorem 5.7. *Given $x_k \in \mathcal{X}$, $\mathcal{X}$ according to Assumption 5.3, and a feasible initial solution at time $k = 0$, the distributed MPC applied to Case 2.4 with stopping criterion $\mathcal{T}_{k,\bar{p}_k} = \emptyset$:*

(I) is recursively feasible,

(II) renders the dynamics (2.2) practically stable in $\mathcal{X}$ with respect to the origin.

Proof. The re-initialization at $k = k+1$ according to (5.41) implies $\mathbf{u}_{k+1|k+1,0} \in \mathbf{U}$. Statement (I) follows by induction over k and noting that feasibility is preserved at any iteration p as shown in the proof of Proposition 4.3.

To establish (II), note that $V^*(x_k)$ is a continuous function and $V^*(x_k) = 0$ if $x_k = 0$ (cf. [6]). It directly follows that there exists $\alpha_2(\|x_k\|)$ such that

$$\alpha_1(\|x_k\|) \leq V^*(x_k) \leq \alpha_2(\|x_k\|) \tag{5.48}$$

holds for all $x_k \in \mathcal{X}$ and $\alpha_1(\|x_k\|) := \lambda_{\min}(Q)\|x_k\|^2$. Next, Theorem 5.3 implies that $V^*(x_k) \leq V(x_k, \mathbf{u}_{k,\bar{p}_k}) \leq V^*(x_k) + \bar{V}_d$ and (5.48) becomes:

$$\alpha_1(\|x_k\|) \leq V(x_k, \mathbf{u}_{k,\bar{p}_{k+1}}) \leq \alpha_2(\|x_k\|) + d_1,$$

where $d_1 = \bar{V}_d$. It follows from (5.47), Theorem 5.3, and $x_{k+1} = Ax_k + Bu_{k|k,\bar{p}_k}$ that

$$V(x_{k+1}, \mathbf{u}_{k+1,\bar{p}_k}) \leq V^*(Ax_k + Bu^*(x_k)) + \bar{V}_d \leq c_{\text{Lev}} \tag{5.49}$$

holds, i.e. $x_{k+1} \in \mathcal{X}$ for all $x_k \in \mathcal{X}$ and the state remains in $\mathcal{X}$ for all $k \in \mathbb{N}_0$. For all $x_k \in \mathcal{X}$ Assumption 5.3 implies that

$$V^*(Ax_k + Bu^*(x_k)) - V^*(x_k) \leq -\eta\left(\|x_k\|_Q^2 + \|u^*(x_k)\|_R^2\right) \tag{5.50}$$

holds. Finally, considering $V^*(x_k) \leq V(x_k, \mathbf{u}_{k,\bar{p}_k})$ and (5.49) it directly follows that:

$$\begin{aligned} V(x_{k+1}, \mathbf{u}_{k+1,\bar{p}_{k+1}}) - V(x_k, \mathbf{u}_{k,\bar{p}_k}) &\leq V(x_{k+1}, \mathbf{u}_{k+1,\bar{p}_{k+1}}) - V^*(x_k) && (5.51)\\ &\leq V^*(Ax_k + Bu^*(x_k)) + \bar{V}_d - V^*(x_k) && (5.52)\\ &\leq -\alpha_3(\|x_k\|) + d_2, && (5.53) \end{aligned}$$

with $\alpha_3(\|x_k\|) := \lambda_{\min}(Q)\|x_k\|^2$, $d_2 = \bar{V}_d$, and the conditions of Theorem 2.1 hold with $L = 1$, $d_1 = d_2 = \bar{V}_d$, $\omega_k = 0$ for all $k \in \mathcal{N}_0$. It directly follows that the closed-loop dynamics are practically stable in $\mathcal{X}$ with respect to $\Sigma = \{0\}$. □

It follows from Theorem 2.1 that the state x_k converges to a bounded neighborhood of the origin, i.e. $\lim_{k\to\infty} x_k \in \mathcal{B}_{d_c}^r(0)$, and d_c can be made arbitrarily small by choosing an appropriate bound $\bar{V}_d$ (cf. Theorem 2.1 for the relationship between d_1,d_2 and d_c). Because no state or terminal constraints are present and the input constraints are assumed to be decoupled, a feasible initialization can directly be obtained locally by ignoring all interactions.

Alternatively, ISS for Case 2.4 may be established by rewriting the closed-loop dynamics into a perturbed system which uses the input of the centralized MPC (2.6), and the difference between the input computed by the suboptimal distributed MPC and centralized MPC is considered as a bounded disturbance ω_k. In this case a weaker assumption on the set $\mathcal{X}$ may be used, but it is not obvious how to verify this assumption (cf. [46]). In contrast Theorem 5.7 provides a tighter bound $\mathcal{B}_{d_c}^r(0)$ and Assumption 5.3 can be verified as outlined above.

5.4. Numerical Examples

To evaluate Algorithm 4.1 and Algorithm 5.1, both are applied to distributed MPC problems with decoupled double-integrator dynamics, local input and state constraints, $N_s = 5$ to $N_s = 30$ subsystems. For each number of subsystems 20 random, potentially fully coupled costs were generated and each resulting problem was solved for four random initial conditions and an initialization obtained by ignoring all interconnections. To define suitable thresholds γ^i, the optimal costs $V^*(x_k)$ were computed, $\bar{V}_d = c_V V^*(x_k)$ was defined, and simulations were performed for $c_V := \{0.1, 0.5, 1\}$. The event-based algorithm was applied until the stopping criterion $\mathcal{T}_{k,\bar{p}_k} = \emptyset$ holds, and problems which did not require any communication were discarded. Afterwards, Algorithm 4.1 was applied for $\bar{p}_k$ iterations starting from the same initial conditions.

Table 5.1 gives the mean over all problem instances for the number of iterations and the number of communication events of the event-based algorithm e_k relative to the number of communication events $e_k^f := N_s \bar{p}_k$ of the algorithm with time-triggered communication. For the event-based algorithm the number of communication events triggered for each controller $\mathcal{C}^i$ and each time step is given by $e_k^i := \sum_{p=0}^{\bar{p}_k} e_{k,p}^i$, $e_{k,p}^i = 1$ if $i \in \mathcal{T}_{k,p}$, and $e_{k,p}^i = 0$ if $i \notin \mathcal{T}_{k,p}$. In other words $e_{k,p}^i$ indicates whether or not a communication event was triggered for $\mathcal{C}^i$ at (k, p). The overall number of communication events triggered by the event-based algorithm is then given by $e_k := \sum_{i=1}^{N_s} e_k^i$. The number of communication events by the event-based algorithm as percentage of the number of messages of the algorithm with full communication is given by e_k / e_k^f. It should be noted that communication events for initialization are not included because they are identical for both algorithms and are only required once at $(k, p) = (0, 0)$ if the algorithm is applied in closed loop.

The value e_k / e_k^f directly shows the reduction in communication. Specifically, a larger choice of $\bar{V}_d$ (i.e. larger c_V) results in fewer messages and iterations, but

Table 5.1.: Averages for number of iterations, ratio of communication events.

	$\bar{p}_k$			e_k / e_k^f [%]		
N_s	$c_V = 0.1$	0.5	1	0.1	0.5	1
5	6.2	3.3	2.3	0.51	0.40	0.31
7	10.3	5.0	3.3	0.52	0.39	0.32
10	14.5	7.4	5.1	0.52	0.36	0.30
15	27.0	15.0	10.3	0.55	0.42	0.35
20	34.9	19.1	13.5	0.53	0.40	0.33
25	48.7	28.3	19.8	0.55	0.43	0.37
30	59.1	34.2	25.2	0.57	0.44	0.37

may also result in stronger suboptimality. For instance, for $c_V = 0.1$ and $N_s = 30$ the number of messages was reduced by 43% compared to the algorithm with full communication, and for $c_V = 1$ the number of messages was reduced by 63% The resulting suboptimality of the solutions is shown in Table 5.2, where it can be seen that a larger c_V results in larger suboptimality at convergence.

Table 5.2 gives, for each number of subsystems N_s, the mean over all problem instances of the relative suboptimality at convergence (i.e. $\mathcal{T}_{k,p} = \emptyset$). The relative suboptimality for each initial value is given by

$$V_{d,\text{rel}}(x_k) := \frac{V_d(x_k, \mathbf{u}_{k,\bar{p}_k})}{V^*(x_k)}.$$

With the bounds $\bar{V}_d = c_V V^*(x_k)$ it directly follows from Theorem 5.3 that $V_{d,\text{rel}}(x_k) \leq c_V$ holds. The results show that the relative suboptimality $V_{d,\text{rel}}(x_k)$ at convergence is in fact much smaller than c_V. This is especially true if the number of subsystems N_s is large and can be explained by the fact that the condition $\mathcal{T}_{k,p} = \emptyset$ used for the suboptimality bound holds for all subsystems with values just below their threshold γ^i. However, as the number of subsystems increases this scenario is very unlikely and typically only the triggering rules of a few subsystems are close to γ^i.

Furthermore, it can be seen that due to the smaller thresholds (i.e. smaller c_V) more iterations are performed if N_s is large. Finally, the event-based approach reduces the number of messages and converges faster, i.e. after the same number of iterations, the suboptimality is lower for the event-based algorithm. The reason for this is that the weights are based on the communicated information which may allow for faster convergence as discussed below Theorem 5.4 and Theorem 5.5.

To illustrate the proposed cooperative distributed MPC algorithm with event-based communication in closed loop, the following numerical example motivated by a system of coupled water tanks with pumps and outlets (see Figure 5.1) taken

Table 5.2.: Average of the relative suboptimality $\frac{V_d(x_k, \mathbf{u}_{k,\bar{p}_k})}{V^*(x_k)}$ at convergence.

	Event-based communication			Full communication		
N_s	$c_V = 0.1$	0.5	1	0.1	0.5	1
5	0.00481	0.01709	0.03512	0.01422	0.03802	0.05278
7	0.00245	0.01167	0.02256	0.00813	0.02581	0.03766
10	0.00170	0.00787	0.01374	0.00655	0.01964	0.02838
15	0.00117	0.00532	0.01027	0.00474	0.01532	0.02453
20	0.00089	0.00399	0.00731	0.00358	0.01184	0.01828
25	0.00076	0.00329	0.00653	0.00311	0.01054	0.01770
30	0.00064	0.00289	0.00525	0.00266	0.00924	0.01458

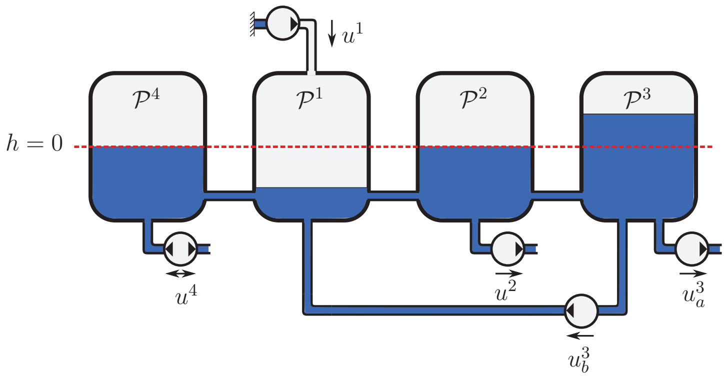

Figure 5.1.: Coupled water tanks [41].

from [41] is used. Specifically, Algorithm 4.1 and Algorithm 5.1 are applied to the case without terminal constraint. Both Algorithms are initialized to a decentralized solution at $k = 0$, and the input sequence $\mathbf{u}_{k+1,0}$ for $k+1$ is initialized according to (5.45). The tanks are modeled as second order systems, where the states $x = (x_k^1; x_k^2; x_k^3; x_k^4) \in \mathbb{R}^8$ with $x_k^i = (h_k^i; \dot{h}_k^i) \in \mathbb{R}^2$ correspond to the water level h_k^i and time-derivative $\dot{h}_k^i$ of the water level. The inputs $u = (u^1; u^2; u_a^3; u_b^3; u^4)$ control the pumps and outlets to fill/release water. The interconnected dynamics of the tanks are given by

$$A = \begin{bmatrix} 0.95 & 0.3718 & 0.05 & 0 & 0 & 0 & 0.005 & 0 \\ 0 & 0.5353 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0.05 & 0 & 0.9 & 0.3718 & 0.05 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.5353 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.05 & 0 & 0.95 & 0.3718 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.5353 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0.3718 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.5353 \end{bmatrix}, \tag{5.54}$$

and

$$B = \begin{bmatrix} 0.1282 & 0 & 0 & -0.1282 & 0 \\ 0.4647 & 0 & 0 & -0.4647 & 0 \\ 0 & 0.1282 & 0 & 0 & 0 \\ 0 & 0.4647 & 0 & 0 & 0 \\ 0 & 0 & 0.1282 & 0.1282 & 0 \\ 0 & 0 & 0.4647 & 0.4647 & 0 \\ 0 & 0 & 0 & 0 & 0.1282 \\ 0 & 0 & 0 & 0 & 0.4647 \end{bmatrix}. \tag{5.55}$$

The overall goal is to control the water levels h_k^i to zero while using the outlets as little as possible. This is formulated by a cost function with:

$$Q = \text{diag}(0.5, 0.1, 0.5, 0.1, 0.5, 0.1, 0.5, 0.1), \quad R = \text{diag}(2, 40, 40, 1, 10).$$

The input constraints are given by $0 \leq u_k^1 \leq 10$, $-10 \leq u_k^4 \leq 10$, and $-10 \leq u_k^i \leq 0$ for $i \in \{2, 3\}$. Simulations were performed for an initial water level of $h_0 = (-20; 0; 10; 0)$, a prediction horizon of $N = 20$, and a maximum number of iterations per time step of $p_{\max} = 20$. For Algorithm 5.1, the bound $\bar{V}_d = 25$ was chosen to reduce the number of messages without strongly degrading performance and, in this example, the tolerance $\epsilon = 10^{-3}$ was used for Algorithm 4.1. Simulation results for the water levels and inputs are shown in Figure 5.2.

It can be seen that, due to the high cost for the control inputs u^2 and u_a^3, only the inputs u^1 and u_b^3 are used. The enlarged area shows slight changes in the inputs if a communication event is triggered. Overall, the states converge close to

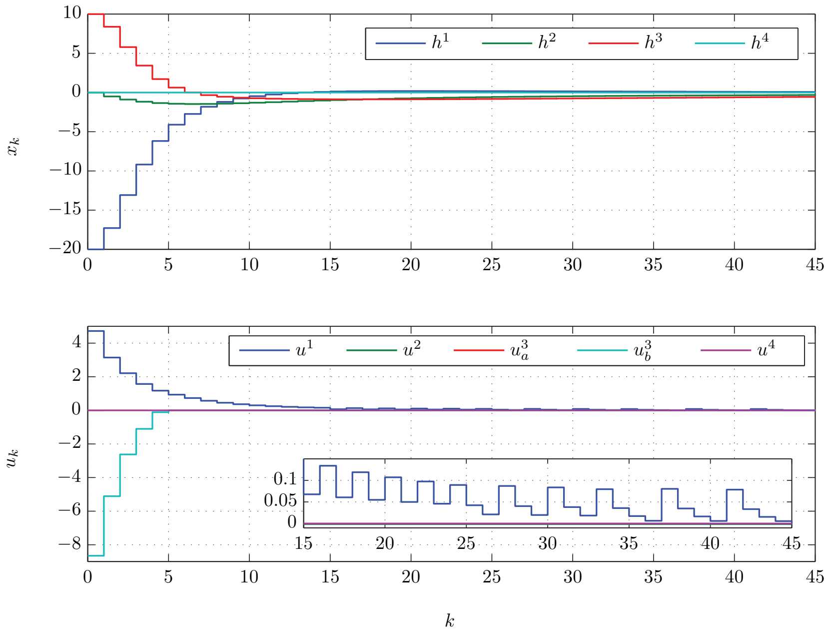

Figure 5.2.: Simulation results for the coupled water tanks using the event-based algorithm (inset: enlarged input sequence).

the control goal despite the event-based communication, highlighting the practical stability property of the closed loop.

The number of communication events e_k^i per controller $\mathcal{C}^i$ and time step k, the number of iterations per time step, and suboptimality $V_d(x_k, \mathbf{u}_{k|k,p})$ are shown in Figure 5.3. It can be seen that the number of events rapidly decreases after a couple of time steps and events only occur occasionally. Furthermore, the stopping criterion $\mathcal{T}_{k,\bar{p}_k} = \emptyset$ holds after a few iterations. In contrast, the number of communication events $e_k^f = N_s\bar{p}_k$ per time step for Algorithm 4.1 remains relatively high. Over all iterations and time steps, 115 communication events are generated by Algorithm 5.1 compared to 2636 for Algorithm 4.1. Finally, the last plot in Figure 5.3 shows the suboptimality $V_d(x_k, \mathbf{u}_{k,p})$ resulting from Algorithm 4.1 and Algorithm 5.1. For comparison, the absolute costs are $V(x_0, \mathbf{u}_{0,0}) \approx 1100$ and $V(x_{45}, \mathbf{u}_{45,0}) \approx 2$). It can be seen that Algorithm 4.1 requires both more iterations and communication while not providing substantially improved performance.

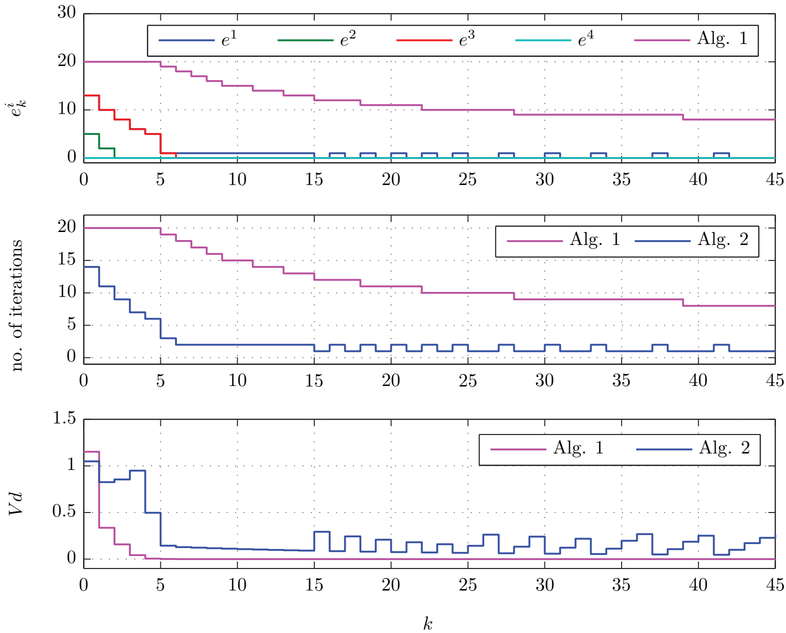

Figure 5.3.: Number of communication events e_k^i, number of iterations per time-step, and suboptimality $V_d(x_k, \mathbf{u}_{k,\bar{p}_k})$.

5.5. Discussion

In this chapter, the distributed MPC algorithm analyzed in Chapter 4 is combined with a suitable triggering function for communication events. In each iteration the controllers $\mathcal{C}^i$ compute candidate input sequences by solving the same problem used in Chapter 4, but the controllers only exchange information if a communication event is triggered. This approach significantly reduces the load on the communication network compared to communicating in every time step and iteration, while often achieving comparable performance. Specifically, a communication event is triggered only if the optimized input sequence computed by $\mathcal{C}^i$ sufficiently improves the closed-loop performance of the overall system. The cooperative distributed MPC algorithm with event-based communication terminates if no communication occurs in an iteration. This simple distributed stopping condition can be easily checked and ensures that the suboptimality of the input sequence is below a prescribed bound $\bar{V}_d$. This bound can be made arbitrarily small at the expense of triggering more communication events and performing more iterations. Therefore, $\bar{V}_d$ allows for a trade-off of closed-loop performance and the load on the communication network. Additionally, the convergence properties of this distributed MPC algorithm are analyzed and bounds on the number of iterations and on the convergence rate are established. Based on the bounds on the convergence rate, the algorithm with time-triggered communication analyzed in Chapter 4 can be compared with the algorithm utilizing event-based communication. On the one hand, the bounds on the convergence rate suggest that the convergence of the algorithm with event-based communication slows down the closer it gets to termination because only few controllers may participate in the algorithm. On the other hand it may converge faster than the cooperative distributed MPC algorithm with time-triggered communication because the weights are adjusted in each iteration based on the set of communicating controllers $\mathcal{T}_{k,p}$ in order to achieve fast convergence. Simulation results show that on average the event-based approach offers slightly faster convergence for a set of randomly created examples.

Overall the algorithm with event-based communication is well suited for closed-loop application because the initialization computed at each time step often already provides good control performance and only few additional communication events are required to further optimize the input sequence. Furthermore, stability results for different types of coupling between subsystems and for distributed MPC with and without terminal constraints are provided based on the initialization computed at each time step, and the suboptimality of the input sequences resulting from applying the cooperative distributed MPC. In contrast to parallel algorithms based on dual decomposition the algorithm can be terminated at any iteration in Case 2.2 and Case 2.3 while retaining all advantages of utilizing parallel optimization.

The event-based approach to communication achieves comparable performance to the case of time-triggered communication analyzed in Chapter 4 and mitigates the drawback of utilizing too many iterations or too much communication. However, two aspects remain open. First, if a communication event is triggered the corresponding controller $\mathcal{C}^i$ has to communicate information to all other controllers, even if the local optimization of a controller $\mathcal{C}^j$ only weakly (or not at all) depends on the communicated information. Therefore, it should be possible to reduce the load on the communication network even further by employing triggering rules which do not only decide when to communicate, but also indicate with which controller information needs to be exchanged.

Secondly, it is not clear how uncertain communication affects the cooperative distributed algorithms with time-triggered or event-based communication. While the algorithm may be modified to cope with small delays $\tau_k \ll \Delta t$ by restricting the optimization to $u_{k+1|k}$ to $u_{k+N-1|k}$, this does not provide an answer to the question of how uncertain communication affects the performance of the algorithm.

These questions will be investigated in the next chapter based on analyzing the sensitivity of the solutions of the local optimization problems (4.3) with respect to the communicated information.

6. Event-Based Communication Based on Sensitivity Analysis

In the event-based distributed MPC algorithm developed in the previous chapter a controller $\mathcal{C}^i$ sends information to all other controllers $\mathcal{C}^j$, $j \in \mathcal{N} \setminus i$ when a communication event is triggered for $\mathcal{C}^i$. If the number of subsystems is large, this results in large number of messages, which may not be required. The approach presented in this chapter aims to reduce the load on the communication network further by analyzing between which controllers communication is required. To this end, the main question is how the local optimizer $\boldsymbol{\rho}^i_{k,p}$ of (4.3) changes if the input sequences $\mathbf{u}^j_{k,p}$, $j \in \mathcal{N} \setminus i$ are only known approximately. For instance, if the sensitivity of the optimizer $\boldsymbol{\rho}^i_{k,p}$ of $\mathcal{C}^i$ with respect to the input sequence $\mathbf{u}^j_{k,p}$ of $\mathcal{P}^j$ is low, the controller $\mathcal{C}^j$ should only rarely need to communicate information to controller $\mathcal{C}^i$.

In the following, an event-based communication scheme for the exchange of information between controllers is developed, in which each controller $\mathcal{C}^i$ only communicates information to another controller $\mathcal{C}^j$ if a communication event is triggered. In other words, the triggering rules do not only decide when to communicate, but also indicate to which controller information needs to be communicated. The main idea behind the algorithm developed in this chapter is to apply Algorithm 4.1 using approximate information about other subsystems' states and input sequences. In this algorithm, event-based communication is used to ensure that the information assumed by a controller $\mathcal{C}^i$ for the state and inputs of $\mathcal{P}^j$ does not differ from the true values too much. Specifically, the implicit function theorem is applied to the local optimizers $\boldsymbol{\rho}^i_{k,p}$ to compute their sensitivity with respect to the communicated information. Based on this analysis triggering conditions for communication events are derived which ensure that the possible cost increase, resulting from using assumed values instead of the true values in the local optimization, is bounded. This allows to analyze the rate of convergence, suboptimality, and closed-loop properties of the algorithm, as well as the impact of communication delays and packet loss.

The approach used in the sensitivity analysis is conceptually similar to approaches used to compute the optimizer of a linear centralized MPC problem as explicit function of the input data (cf. [6]). In fact, if $\boldsymbol{\rho}^i_{k,p}$ is known explicitly, bounds on the sensitivity of the optimal solution with respect to the states and input sequences of other subsystems can be easily obtained. However, for medium to large scale systems computing the explicit solution is not feasible due to the high computational complexity of this task. Even computing the optimal solution $\boldsymbol{\rho}^i_{k,p}$ of the local problems (4.3) explicitly is, in general, a challenging problem, because the explicit solution has to be parametrized in the states and input sequences of all subsystems. Therefore, a high dimensional search space has to be explored to compute the optimizer explicitly. This typically results in a very high computational complexity (cf. [6]). In this chapter, this issue is resolved by analytically computing bounds on the sensitivity with respect to the communicated information without computing the solution for $\boldsymbol{\rho}^i_{k,p}$ explicitly.

In [95] a so called "sensitivity based" algorithm for distributed MPC has been proposed, but the methodology is very different from the approach considered in this chapter. In particular, in [95] the "sensitivity" interpretation of the Lagrange multipliers (cf. [15], Section 5.6.2) and dual decomposition are used by each controller $\mathcal{C}^i$ to construct linear approximations of the costs of an interconnected controller $\mathcal{C}^j$. Furthermore, every controller communicates with every other controller in every time step and iteration, the cost function is assumed to be separable, and no results on closed-loop stability are given.

In contrast, in this chapter the goal is to minimize communication and establish convergence as well as practical stability of the closed-loop. To this end, rigorous bounds on the sensitivity of the local optimizer are derived and used to define triggering functions for communication events. The stability properties of the closed-loop are analyzed for Case 2.3 (coupled costs, decoupled dynamics and constraints) and Case 2.4 (coupled costs and dynamics, no state or terminal constraints). Because each controller only knows the states and inputs of interconnected subsystems and controllers approximately, it is not possible to ensure that the terminal equality constraint for Case 2.2 (coupled costs and dynamics, no state constraints) holds. Therefore, the algorithm proposed in this chapter is not applicable to Case 2.2. A distributed MPC algorithm based on similar ideas has been previously published by the author in [41] without any formal proofs of convergence, stability of the closed-loop, or rigorous sensitivity analysis.

6.1. Distributed MPC Using Approximate Information

The algorithm developed in this chapter is based on Algorithm 4.1 and uses event-based communication, but does not discard information that is not communicated. To clearly differentiate between the two Algorithms the input sequence optimized by the distributed MPC in this chapter is denoted by $\mathbf{v}_{k,p}$, while $\mathbf{u}_{k,p+1}$ denotes the input sequence obtained by performing one iteration of Algorithm 4.1 starting from the inputs $\mathbf{v}_{k,p}$.

Updating the local input sequence $\mathbf{v}^i_{k,p}$ without communication may lead to a mismatch between the values used by different controllers. To this end, let $\hat{\mathbf{v}}^{i,j}_{k,p}$

and $\hat{x}_k^{i,j}$ denote the values assumed by controller $\mathcal{C}^i$ for the inputs and states of $\mathcal{P}^j$ for all $j \in \mathcal{N} \setminus i$, and let $\delta\hat{\mathbf{v}}_{k,p}^{i,j} := \hat{\mathbf{v}}_{k,p}^{i,j} - \mathbf{v}_{k,p}^{j}$ and $\delta\hat{x}_k^{i,j} := \hat{x}_k^{i,j} - x_k^j$ denote the corresponding errors. The value assumed by $\mathcal{C}^i$ for the global state is denoted by $\hat{x}_{k,p}^i := (\hat{x}_{k,p}^{i,1}; \dots; \hat{x}_{k,p}^{i,N_s})$, with $\hat{x}_{k,p}^{i,i} := x_k^i$. Based on these values, the local optimization problem (4.3) becomes:

$$\begin{aligned} \boldsymbol{\phi}_{k,p}^i := \arg\min\nolimits_{\mathbf{v}_k^i} \quad & V(\hat{x}_{k,p}^i, \mathbf{v}_k) \\ \text{s.t.} \quad & \mathbf{v}_k^i \in \mathbf{U}^i(x_k^i),\ \mathbf{v}_k^j = \hat{\mathbf{v}}_{k,p}^{i,j}, \forall j \in \mathcal{N} \setminus i. \end{aligned} \tag{6.1}$$

In contrast, the local optimization using exact information is given by

$$\begin{aligned} \boldsymbol{\rho}_{k,p}^i := \arg\min\nolimits_{\mathbf{v}_k^i} \quad & V(x_{k,p}, \mathbf{v}_k) \\ \text{s.t.} \quad & \mathbf{v}_k^i \in \mathbf{U}^i(x_k^i),\ \mathbf{v}_k^j = \mathbf{v}_{k,p}^{i,j}, \forall j \in \mathcal{N} \setminus i, \end{aligned} \tag{6.2}$$

and locally the following iterations (cf. (4.5)) with $w^i = \sqrt{c_H^i c_r^{-1}}$ (see Section 4.3) are performed by each controller $\mathcal{C}^i$, $i \in \mathcal{N}$:

$$\mathbf{v}_{k,p+1}^i := (1 - w^i)\mathbf{v}_{k,p}^i + w^i \boldsymbol{\phi}_{k,p}^i. \tag{6.3}$$

Using $\bar{\mathbf{v}}_{k,p}^i := (\mathbf{v}_{k,p}^1; \dots; \boldsymbol{\phi}_{k,p}^i; \dots; \mathbf{v}_{k,p}^{N_s})$, and considering that $V(x_k, \mathbf{v}_{k,p})$ one obtains:

$$\mathbf{v}_{k,p+1} := \sum\nolimits_{i \in \mathcal{N}} w^i \bar{\mathbf{v}}_{k,p}^i, \quad V(x_k, \mathbf{v}_{k,p+1}) \le \sum\nolimits_{i \in \mathcal{N}} w^i\, V(x_k, \bar{\mathbf{v}}_{k,p}^i). \tag{6.4}$$

In order to compare the difference between the two optimization problems (6.1) and (6.2), let $\mathbf{u}_{k,p}$ denote the input generated based on (6.2), such that for each each controller $\mathcal{C}^i$, $i \in \mathcal{N}$:

$$\mathbf{u}_{k,p+1}^i := (1 - w^i)\mathbf{v}_{k,p}^i + w^i \boldsymbol{\rho}_{k,p}^i. \tag{6.5}$$

With $\bar{\mathbf{u}}_{k,p}^i := (\mathbf{v}_{k,p}^1; \dots; \boldsymbol{\rho}_{k,p}^i; \dots; \mathbf{v}_{k,p}^{N_s})$ this results in

$$\mathbf{u}_{k,p+1} := \sum\nolimits_{i \in \mathcal{N}} w^i \bar{\mathbf{u}}_{k,p}^i, \quad V(x_k, \mathbf{u}_{k,p+1}) \le \sum\nolimits_{i \in \mathcal{N}} w^i\, V(x_k, \bar{\mathbf{u}}_{k,p}^i). \tag{6.6}$$

It can be seen that the difference $V(x_k, \mathbf{v}_{k,p}) - V(x_k, \mathbf{u}_{k,p})$ between the costs using inexact information and full communication depends on the optimizers $\boldsymbol{\rho}_{k,p}^i$ and $\boldsymbol{\phi}_{k,p}^i$ given by (6.2) and (6.1), which implicitly depend on the communicated information. Therefore, to gain insight into how $\delta\hat{\mathbf{v}}_{k,p}^{i,j}$ and $\delta\hat{x}_k^{i,j}$ affect the overall control performance, the sensitivity of $\boldsymbol{\phi}_{k,p}^i$ with respect to $\delta\hat{\mathbf{v}}_{k,p}^{i,j}$ and $\delta\hat{x}_k^{i,j}$ is analyzed in the next section.

6.2. Sensitivity Analysis

In order to anaylze the impact of using inexact values in the local optimization on the closed loop, it is crucial to analyze how the optimizer (6.1) changes with regard to the communicated information. To reformulate the problem to a more suitable form, H and F are partitioned according to the subsystems $\mathcal{P}^i$:

$$H = \begin{bmatrix} H^{1,1} & \cdots & H^{1,N_s} \\ \vdots & \ddots & \vdots \\ H^{N_s,1} & \cdots & H^{N_s,N_s} \end{bmatrix}, \quad F = \begin{bmatrix} F^{1,1} & \cdots & F^{1,N_s} \\ \vdots & \ddots & \vdots \\ F^{N_s,1} & \cdots & F^{N_s,N_s} \end{bmatrix}.$$

Furthermore, $\varphi^{\backslash i}_{k,p} = (\varphi^{i}_{k,p}; \ldots; \varphi^{i-1}_{k,p}; \varphi^{i+1}_{k,p}; \ldots, \varphi^{N_s}_{k,p}) \in \mathbb{R}^{t^i}$ denotes the parameter vector containing the information about subsystems $j \neq i$ with $t^i := \sum_{j \in \mathcal{N}^i} m^j + n^j$, and $\varphi^i_{k,p} = (x^i_k; \mathbf{v}^i_{k,p})$ denotes the actual state and input sequence of subsystem $\mathcal{P}^i$. Based on this, the following local cost function is obtained where the term $\mathscr{R}^i(x^i_k, \varphi^{\backslash i}_{k,p})$ collects terms which do not depend on $\mathbf{v}^i_k$ (i.e. are constant in the local optimization), such that $V^i(x^i_k, \mathbf{v}^i_{k,p}, \varphi^{\backslash i}_{k,p}) = V(x_k, \mathbf{v}_{k,p})$, and

$$V^i(x^i_k, \mathbf{v}^i_k, \varphi^{\backslash i}_{k,p}) := (\mathbf{v}^i_k)^T H^{i,i} \mathbf{v}^i_k + (x^i_k)^T F^{i,i} \mathbf{v}^i_k + (\varphi^{\backslash i}_{k,p})^T F^{\backslash i}_c \mathbf{v}^i_k + \mathscr{R}^i(x^i_k, \varphi^{\backslash i}_{k,p}), \quad (6.7)$$

where $F^{\backslash i}_c := (F^{1,i}_c; \ldots; F^{i-1,i}_c; F^{i+1,i}_c; \ldots; F^{N_s,i}_c;)$ and $F^{j,i}_c = (F^{j,i}; 2H^{j,i})$. Based on this local cost function, the optimization problem (6.1) can be rewritten such that

$$\boldsymbol{\phi}^i_{k,p} := \arg\min_{\mathbf{v}^i_k} V^i(x^i_k, \mathbf{v}^i_k, \varphi^{\backslash i}_{k,p}), \text{ s.t. } f^i_{\text{in}}(\mathbf{v}^i_k, x^i_k) \leq 0, \quad (6.8)$$

where $f^i_{\text{in}}(\mathbf{v}^i_k, x^i_k) = \mathbf{C}^i_{\mathbf{U}} \mathbf{v}^i_k - \mathbf{b}^i_{\mathbf{U}}(x^i_k) \leq 0$ (see Appendix A.1), and $f^i_{\text{in},j}(\mathbf{v}^i_k, x^i_k)$ with $j \in \{1, \ldots, h^i_{\text{in}}\}$ denotes the j-th row of $f^i_{\text{in}}(\mathbf{v}^i_k, x^i_k)$. Furthermore, $\mathcal{N}^i$ denotes the index set of subsystems interconnected with subsystem i:

$$\mathcal{N}^i := \left\{ j \in \mathcal{N} \setminus i \,\middle|\, F^{j,i} \neq 0,\ H^{j,i} \neq 0 \right\}. \quad (6.9)$$

The Lagrangian of (6.8) is defined as follows:

$$\mathcal{L}(x^i_k, \mathbf{v}^i_k, \varphi^{\backslash i}_{k,p}, \lambda^i) := V^i(x^i_k, \mathbf{v}^i_k, \varphi^{\backslash i}_{k,p}) + (\lambda^i)^T f^i_{\text{in}}(\mathbf{v}^i_k, x^i_k),$$

where $\lambda = (\lambda^i_1; \ldots; \lambda^i_{h^i_{\text{in}}})$ is the vector of Lagrange multipliers.

In the following the main idea is to consider the solution of the local optimization problem (6.8) as implicit function of the communicated information and to compute the dependency of the optimal value on the communicated information. While the optimal solution of the local optimization problem is a continuous function of the communicated values (cf. [6]) it is not continuously differentiable, and it is not clear how the implicit function theorem could be applied directly. In [32] a method for sensitivity analysis is presented which resolves this issue by applying the implicit function theorem to the KKT optimality conditions (cf. Section 2.4) with respect to the parameters $\varphi^i_{k,p}$. This allows to analyze how the optimal pair $(\boldsymbol{\phi}^i_{k,p}, \lambda^{i*}_{k,p})$ changes if the parameters $\varphi^i_{k,p}$ change. The approach here is inspired by that in [32], but the results in [32] are stated for a rather general class of non-convex problems and require a number of assumptions which are not required for the results presented in subsequent parts of this chapter.

Assumption 6.1. *It is assumed that for all feasible parameters* $\varphi_{k,p}^{\backslash i}$ *and* x_k^i *of* (6.8) *a unique optimal pair* $(\boldsymbol{\phi}_{k,p}^i, \lambda_{k,p}^{i*})$ *exists such that:*

1. *the KKT conditions* (2.31) *to* (2.35) *hold,*
2. *linear independence constraint qualification (LICQ) holds,*
3. *strict complementary slackness holds, i.e.:* $\lambda_{j,k,p}^{i*} > 0$ *if* $f_{in,j}^i(\boldsymbol{\phi}_{k,p}^i, x_k^i) = 0$.

Assumption 6.1.1 implies that the following optimality conditions hold:

$$\nabla_{\boldsymbol{\phi}_{k,p}^i} \mathcal{L}(x_k^i, \boldsymbol{\phi}_{k,p}^i, \varphi_{k,p}^{\backslash i}, \lambda_{k,p}^{i*}) = 0, \tag{6.10}$$

$$(\lambda_{k,p}^{i*})^T f_{\text{in}}^i(\boldsymbol{\phi}_{k,p}^i, x_k^i) = 0. \tag{6.11}$$

Assumption 6.1.2 and Assumption 6.1.3 ensure that the solution to (6.8) is non-degenerate. If these assumptions do not hold for parameters $(\varphi_{k,p}^{\backslash i}, x_k^i)$ there are redundant active constraints which can be removed from the problem, such that Assumption 6.1 holds without changing the optimizer $\boldsymbol{\phi}_{k,p}^i$. Note that the assumption does not require that a solver finds an optimal pair which satisfies Assumption 6.1 online, but only that such a pair exists for each state x_k^i and each vector $\varphi_{k,p}^{\backslash i}$.

The variation of the KKT pair $(\boldsymbol{\phi}_{k,p}^i, \lambda^{i*})$ with respect to the communicated information $\varphi_{k,p}^i$ can be analyzed by applying the implicit-function theorem (cf. [84], Theorem 9.1 and Theorem 9.2) to the optimality conditions (6.10) and (6.11) (cf. [32], Theorem 3.2.2 to Corollary 3.2.4).

Let $J_{\mathbf{v}_{k,p}^i}^i$ denote the Jacobian of (6.10) with respect to $\mathbf{v}_k^i, \lambda_{k,p}^{i*}$ evaluated at $x_k^i, \boldsymbol{\phi}_{k,p}^i, \lambda_{k,p}^{i*}$ and $\varphi_{k,p}^{\backslash i}$. Furthermore, $J_{\varphi_{k,p}^{\backslash i}}^i$ denotes the Jacobian of (6.11) with respect to $\varphi_{k,p}^{\backslash i}$ evaluated at $x_k^i, \boldsymbol{\phi}_{k,p}^i, \lambda_{k,p}^{i*}$ and $\varphi_{k,p}^{\backslash i}$:

$$J_{\mathbf{v}_{k,p}^i}^i := \begin{bmatrix} \nabla_{\mathbf{v}_{k,p}^i}^2 \mathcal{L} & \nabla_{\mathbf{v}_{k,p}^i} f_{\text{in},1}^i & \cdots & \nabla_{\mathbf{v}_{k,p}^i} f_{\text{in},h_{\text{in}}^i}^i \\ \lambda_1^i \left(\nabla_{\mathbf{v}_{k,p}^i} f_{\text{in},1}^i\right)^T & f_{\text{in},1}^i & & 0 \\ \vdots & & \ddots & \\ \lambda_{h_{\text{in}}^i}^i \left(\nabla_{\mathbf{v}_{k,p}^i} f_{\text{in},h_{\text{in}}^i}^i\right)^T & 0 & & f_{\text{in},h_{\text{in}}^i}^i \end{bmatrix}, \tag{6.12}$$

$$J_{\varphi_{k,p}^{\backslash i}}^i := \left(\nabla_{\varphi_{k,p}^{\backslash i}} \nabla_{\mathbf{v}_{k,p}^i} \mathcal{L}; \lambda_1^i (\nabla_{\varphi_{k,p}^{\backslash i}} f_{\text{in},1}^i)^T; \ldots; \lambda_{h_{\text{in}}^i}^i (\nabla_{\varphi_{k,p}^{\backslash i}} f_{\text{in},h_{\text{in}}^i}^i)^T\right). \tag{6.13}$$

If $J_{\mathbf{v}_{k,p}^i}^i$ is non-singular, the implicit function theorem applied to (6.10) and (6.11) results in

$$\begin{bmatrix} \nabla_{\varphi_{k,p}^{\backslash i}} \boldsymbol{\phi}_{k,p}^i \\ \nabla_{\varphi_{k,p}^{\backslash i}} \lambda_{k,p}^{i*} \end{bmatrix} = (J_{\mathbf{v}_{k,p}^i}^i)^{-1} J_{\varphi_{k,p}^{\backslash i}}^i . \tag{6.14}$$

Under the assumption that $J^i_{\mathbf{v}^i_{k,p}}$ is non-singular, the following two piecewise affine functions are obtained:

$$\begin{bmatrix}\boldsymbol{\phi}^i(\varphi^{\setminus i})\\ \lambda^{i*}(\varphi^{\setminus i})\end{bmatrix} = \begin{bmatrix}\boldsymbol{\phi}^i_{k,p}(\varphi^{\setminus i}_{k,p})\\ \lambda^{i*}_{k,p}(\varphi^{\setminus i}_{k,p})\end{bmatrix} + (J^i_{\mathbf{v}^i_{k,p}})^{-1} J^i_{\varphi^{\setminus i}_{k,p}} (\varphi^{\setminus i}_{k,p} - \varphi^{\setminus i}), \quad \forall \varphi^{\setminus i} \in \mathcal{R}(\varphi^{\setminus i}_{k,p}). \tag{6.15}$$

For a given state x^i_k this function is valid in a polytopic region $\mathcal{R}(\varphi^{\setminus i}_{k,p})$. In particular, it is well known that the optimizer of problem (6.1) is a continuous, piecewise affine function of the parameters $\varphi^{\setminus i}_{k,p}$ (cf. [6]). By substituting (6.15) into the constraints and optimality conditions it can be verified that (6.15) is the solution for (6.1) on a polytopic neighborhood of $\varphi^i_{k,p}$ given by:

$$\mathcal{R}(\varphi^{\setminus i}_{k,p}) := \left\{ \varphi^{\setminus i} \in \mathbb{R}^{t^i} \;\middle|\; \begin{array}{l} f_{\text{in}}(\boldsymbol{\phi}^i(\varphi^{\setminus i}), x^i_k) \leq 0 \\ \lambda^{i*}(\varphi^{\setminus i}) \geq 0 \\ (\lambda^{i*}(\varphi^{\setminus i})^T f^i_{\text{in}}(\boldsymbol{\phi}^i(\varphi^{\setminus i}), x^i_k) = 0 \end{array} \right\}$$

Let $\partial\mathcal{S} := \mathcal{S} \setminus \text{int}(\mathcal{S})$ denote the boundary of a set $\mathcal{S}$, and $\varphi^{\setminus i}_s$ with $s \in \{1, \dots, N^i_{\mathcal{R}}\}$ denotes parameters such that for every pair of parameters s_j and s_i solving (6.1) results in different sets of active constraints. The following proposition summarizes some well known properties of the regions $\mathcal{R}(\varphi^{\setminus i}_s)$ (see e.g. [6] for a proof):

Proposition 6.1. *The regions $\mathcal{R}(\varphi^{\setminus i}_s)$ for parameters $\varphi^{\setminus i}_s$ with different sets of active constraints are non-empty, do not overlap, and the boundary of two neighboring regions belongs to both regions. In other words $\varphi^{\setminus i} \in int(\mathcal{R}(\varphi^{\setminus i}_{s_1}))$ implies $\varphi^{\setminus i} \notin int(\mathcal{R}(\varphi^{\setminus i}_{s_2}))$ for all $s_1 \neq s_2$ and for neighboring regions there exists $\varphi^{\setminus i}$ such that $\varphi^{\setminus i} \in \partial\mathcal{R}(\varphi^{\setminus i}_{s_2})$, and $\varphi^{\setminus i} \in \partial\mathcal{R}(\varphi^{\setminus i}_{s_1})$.*

Applying (6.15) to the problem at hand leads to:

$$J^i_{\mathbf{v}^i_{k,p}} = \begin{bmatrix} 2H^{i,i} & (\mathbf{C}^i_{\mathbf{U}})^T \\ \tilde{\lambda}^i_{k,p}\mathbf{C}^i_{\mathbf{U}} & \tilde{f}^i_{\text{in},k,p} \end{bmatrix}, \quad J^i_{\varphi^i,k,p} = \begin{bmatrix} (F^{\setminus i}_c)^T \\ \mathbf{0}_{r^i \times t^i} \end{bmatrix}, \tag{6.16}$$

where $\tilde{\lambda}^i_{k,p} = \text{diag}(\lambda^{i*}_{k,p})$, $\tilde{f}^i_{\text{in},k,p} = \text{diag}(\mathbf{C}^i_{\mathbf{U}}\boldsymbol{\phi}^i_{k,p} - \mathbf{b}^i_{\mathbf{U}}(x^i_k))$ and $\mathbf{0}_{t^i \times r^i} \in \mathbb{R}^{t^i \times r^i}$ is a zero matrix.

Proposition 6.2. *The matrix $J^i_{\mathbf{v}^i_{k,p}}$ is non-singular for every optimal pair $(\boldsymbol{\phi}^i_{k,p}, \lambda^{i*})$ which satisfies Assumption 6.1.*

Proof. For every optimal pair $(\boldsymbol{\phi}^i_{k,p}, \lambda^{i*})$ there exists a permutation matrix $T^i_{k,p} \in \{0,1\}^{r^i \times r^i}$, such that $T^i_{k,p}(T^i_{k,p})^T = I$ and

$$T^i_{k,p}\tilde{f}^i_{\text{in},k,p}(T^i_{k,p})^T = \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \breve{f}^i_{\text{in},k,p} \end{bmatrix}, \quad T^i_{k,p}\tilde{\lambda}^i_{k,p}(T^i_{k,p})^T = \begin{bmatrix} \check{\lambda}^i_{k,p} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}, \tag{6.17}$$

where $\mathbf{0}$ denotes a zero matrix of appropriate dimension, $\breve{f}^i_{\text{in},k,p}$ denotes a diagonal matrix containing the inactive constraints $f^i_{\text{in},j} < 0$, and $\check{\lambda}^i_{k,p}$ is a diagonal matrix containing the multipliers of the active constraints. Furthermore, $\check{\mathbf{C}}^i_{\mathbf{U}}$ and $\breve{\mathbf{C}}^i_{\mathbf{U}}$ denote the half spaces of the active and inactive constraints, such that $T^i_{k,p}\mathbf{C}^i_{\mathbf{U}} = (\check{\mathbf{C}}^i_{\mathbf{U}}, \breve{\mathbf{C}}^i_{\mathbf{U}})$. Considering $T^i_{k,p}(T^i_{k,p})^T = I$, and $\breve{T}^i_{k,p} := \text{blkdiag}(I, T^i_{k,p})$ it holds that

$$\begin{aligned}\breve{T}^i_{k,p} J^i_{\mathbf{v}^i_{k,p}} (\breve{T}^i_{k,p})^T &= \begin{bmatrix} 2H^{i,i} & (\mathbf{C}^i_{\mathbf{U}})^T (T^i_{k,p})^T \\ T^i_{k,p}\tilde{\lambda}^i_{k,p}(T^i_{k,p})^T T^i_{k,p}\mathbf{C}^i_{\mathbf{U}} & T^i_{k,p}\tilde{f}^i_{\text{in},k,p}(T^i_{k,p})^T \end{bmatrix}, \\ &= \begin{bmatrix} 2H^{i,i} & & (\check{\mathbf{C}}^i_{\mathbf{U}})^T & (\breve{\mathbf{C}}^i_{\mathbf{U}})^T \\ \check{\lambda}^i_{k,p}\check{\mathbf{C}}^i_{\mathbf{U}} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \breve{f}^i_{\text{in},k,p} \end{bmatrix}.\end{aligned}$$

Assumption 6.1.2 implies that the rows of $\check{\mathbf{C}}^i_{\mathbf{U}}$ are linearly independent, and because of Assumption 6.1.3 it holds that $\check{\lambda}^i_{k,p}$ and $\breve{f}^i_{\text{in},k,p}$ are full rank. Together with $H^{i,i} \succ 0$ it follows that $\breve{T}^i_{k,p} J^i_{\mathbf{v}^i_{k,p}} (\breve{T}^i_{k,p})^T$ and $J^i_{\mathbf{v}^i_{k,p}}$ are non-singular. □

Considering the structure of $J^i_{\varphi^i,k,p}$, only the upper left block of $(J^i_{\mathbf{v}^i_{k,p}})^{-1}$ is required to analyze the variation of $\boldsymbol{\phi}^i_{k,p}$. To this end $(J^i_{\mathbf{v}^i_{k,p}})^{-1}$ is partitioned such that

$$(J^i_{\mathbf{v}^i_{k,p}})^{-1} = \begin{bmatrix} K^i(\varphi^i_{k,p}) & (J^i_{\mathbf{v}^i_{k,p}})^{-1}_{1,2} \\ (J^i_{\mathbf{v}^i_{k,p}})^{-1}_{2,1} & (J^i_{\mathbf{v}^i_{k,p}})^{-1}_{2,2} \end{bmatrix},$$

where $K^i(\varphi^i_{k,p}) \in \mathbb{R}^{m^i \times m^i}$. The following result is concerned with establishing properties of $K^i(\varphi^i_{k,p})$, which are used to study the sensitivity of the local optimization problems.

Theorem 6.1. *For every feasible parameter vector $\varphi^i_{k,p}$ and corresponding optimal pair $(\boldsymbol{\phi}^i_{k,p}, \lambda^{i*})$ according to Assumption 6.1, it holds that $0 \preceq K^i(\varphi^i_{k,p}) \preceq \Xi^i$, where $\Xi^i := (2H^{i,i})^{-1}$.*

Proof. Performing blockwise inversion for $J^i_{\mathbf{v}^i_{k,p}}$ according to (6.16) results in

$$\begin{aligned}K^i(\varphi^i_{k,p}) =& \Xi^i - \Xi^i (C^i_{\mathbf{U}})^T \left(-\tilde{f}^i_{\text{in},k,p} + \tilde{\lambda}^i_{k,p} C^i_{\mathbf{U}} \Xi^i (C^i_{\mathbf{U}})^T\right)^{-1} \tilde{\lambda}^i_{k,p}\Xi^i, \\ =& \Xi^i - \Xi^i (C^i_{\mathbf{U}})^T (T^i_{k,p})^T T^i_{k,p} \left(-\tilde{f}^i_{\text{in},k,p} + \tilde{\lambda}^i_{k,p} C^i_{\mathbf{U}} \Xi^i (C^i_{\mathbf{U}})^T\right)^{-1} \\ & (T^i_{k,p})^{-1} T^i_{k,p} \tilde{\lambda}^i_{k,p} (T^i_{k,p})^T T^i_{k,p} \Xi^i,\end{aligned}$$

where $\Xi^i = (2H^{i,i})^{-1} = (\Xi^i)^T \succ 0$. Considering (6.17), $(T^i_{k,p})^{-1} = (T^i_{k,p})^T$, $T^i_{k,p}\mathbf{C}^i_{\mathbf{U}} = (\check{\mathbf{C}}^i_{\mathbf{U}}, \breve{\mathbf{C}}^i_{\mathbf{U}})$, and $(T^i_{k,p}\Xi^i(T^i_{k,p})^T)^{-1} = T^i_{k,p}(\Xi^i)^{-1}(T^i_{k,p})^T$ it follows that

$$K^i(\varphi^i_{k,p}) = \Xi^i - \Xi^i \begin{bmatrix} \check{\mathbf{C}}^i_{\mathbf{U}} \\ \breve{\mathbf{C}}^i_{\mathbf{U}} \end{bmatrix}^T \begin{bmatrix} \check{\lambda}^i_{k,p}\check{\mathbf{C}}^i_{\mathbf{U}}\Xi^i(\check{\mathbf{C}}^i_{\mathbf{U}})^T & \check{\lambda}^i_{k,p}\check{\mathbf{C}}^i_{\mathbf{U}}\Xi^i(\breve{\mathbf{C}}^i_{\mathbf{U}})^T \\ \mathbf{0} & -\breve{f}^i_{\text{in},k,p} \end{bmatrix}^{-1} \begin{bmatrix} \check{\lambda}^i_{k,p} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \check{\mathbf{C}}^i_{\mathbf{U}} \\ \breve{\mathbf{C}}^i_{\mathbf{U}} \end{bmatrix} \Xi^i.$$

Performing blockwise inversion it follows that

$$K^i(\varphi^i_{k,p}) = \Xi^i - \Xi^i \begin{bmatrix} \check{\mathbf{C}}^i_{\mathbf{U}} \\ \check{\mathbf{C}}^i_{\mathbf{U}} \end{bmatrix}^T \begin{bmatrix} (\check{\mathbf{C}}^i_{\mathbf{U}} \Xi^i (\check{\mathbf{C}}^i_{\mathbf{U}})^T)^{-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \check{\mathbf{C}}^i_{\mathbf{U}} \\ \check{\mathbf{C}}^i_{\mathbf{U}} \end{bmatrix} \Xi^i,$$

and

$$K^i(\varphi^i_{k,p}) = \Xi^i - \Xi^i (\check{\mathbf{C}}^i_{\mathbf{U}})^T (\check{\mathbf{C}}^i_{\mathbf{U}} \Xi^i (\check{\mathbf{C}}^i_{\mathbf{U}})^T)^{-1} \check{\mathbf{C}}^i_{\mathbf{U}} \Xi^i.$$

Because $\Xi^i = (\Xi^i)^T \succ 0$ and $\check{\mathbf{C}}^i_{\mathbf{U}}$ is full rank it holds that $(\check{\mathbf{C}}^i_{\mathbf{U}} \Xi^i (\check{\mathbf{C}}^i_{\mathbf{U}})^T)^{-1} \succ 0$. This results in

$$\Xi^i (\check{\mathbf{C}}^i_{\mathbf{U}})^T (\check{\mathbf{C}}^i_{\mathbf{U}} \Xi^i (\check{\mathbf{C}}^i_{\mathbf{U}})^T)^{-1} \check{\mathbf{C}}^i_{\mathbf{U}} \Xi^i \succ 0,$$

and it directly follows that $K^i(\varphi^i_{k,p}) \preceq \Xi^i$. It remains to show that

$$\Xi^i - \Xi^i (\check{\mathbf{C}}^i_{\mathbf{U}})^T (\check{\mathbf{C}}^i_{\mathbf{U}} \Xi^i (\check{\mathbf{C}}^i_{\mathbf{U}})^T)^{-1} \check{\mathbf{C}}^i_{\mathbf{U}} \Xi^i \succeq 0. \tag{6.18}$$

It follows from the Schur complement [15] that (6.18) holds if and only if

$$\begin{bmatrix} \Xi^i & \Xi^i (\check{\mathbf{C}}^i_{\mathbf{U}})^T \\ \check{\mathbf{C}}^i_{\mathbf{U}} \Xi^i & \check{\mathbf{C}}^i_{\mathbf{U}} \Xi^i (\check{\mathbf{C}}^i_{\mathbf{U}})^T \end{bmatrix} \succeq 0, \tag{6.19}$$

holds. Multiplying (6.19) from the left by $T^i_\Xi := \left[\begin{smallmatrix} I & 0 \\ -\check{\mathbf{C}}^i_{\mathbf{U}} & I \end{smallmatrix}\right]$ and from the right by $(T^i_\Xi)^T$ results in

$$\begin{bmatrix} \Xi^i & 0 \\ 0 & 0 \end{bmatrix} \succeq 0. \tag{6.20}$$

This holds because $\Xi^i \succ 0$ and it follows that (6.18) holds and the theorem follows. □

Based on this result, the next theorem bounds the change in the difference between two primal optimizers obtained for different parameter vectors.

Theorem 6.2. *Given two feasible parameter vectors $\varphi_1^{\setminus i}$ and $\varphi_l^{\setminus i}$ it holds that*

$$\|(H^{i,i})^{\frac{1}{2}}(\boldsymbol{\phi}^i(\varphi_l^{\setminus i}) - \boldsymbol{\phi}^i(\varphi_1^{\setminus i}))\| \leq \|0.5(H^{i,i})^{-\frac{1}{2}}(F_c^{\setminus i})^T(\varphi_l^{\setminus i} - \varphi_1^{\setminus i})\|. \tag{6.21}$$

Proof. If $\varphi_l^{\setminus i} \in \mathcal{R}(\varphi_1^{\setminus i})$ then it holds that

$$\boldsymbol{\phi}^i(\varphi_l^{\setminus i}) - \boldsymbol{\phi}^i(\varphi_1^{\setminus i}) = K^i(\varphi_1^{\setminus i})(F_c^{\setminus i})^T(\varphi_l^{\setminus i} - \varphi_1^{\setminus i}). \tag{6.22}$$

If the parameter vectors $\varphi_l^{\setminus i}$, $\varphi_1^{\setminus i}$ result in different active constraint combinations they are not in the same polytopic region $\mathcal{R}$ of the piecewise affine control law

(6.15). The difference in the optimizer $\boldsymbol{\phi}^i$ for the two parameter vectors can be obtained by considering more than one region. Let $\varphi_s^{\setminus i} = \varphi_1^{\setminus i} + \theta_s(\varphi_l^{\setminus i} - \varphi_1^{\setminus i})$ with $s \in \{2, \dots, l-1\}$ and $\theta_1 = 0 < \theta_2 < \dots < \theta_{l-1} < \theta_l = 1$ denote points on the boundary between every two neighboring polytopic regions intersected by the line $\varphi_1^{\setminus i} + \theta(\varphi_l^{\setminus i} - \varphi_1^{\setminus i})$ with $0 \leq \theta \leq 1$. Because the regions are convex and non-empty, it holds that $\varphi_s^{\setminus i} \in \partial\mathcal{R}(0.5(\varphi_{s-1}^{\setminus i} + \varphi_s^{\setminus i}))$, $\varphi_s^{\setminus i} \in \partial\mathcal{R}(0.5(\varphi_s^{\setminus i} + \varphi_{s+1}^{\setminus i}))$ for all $s \in \{2, \dots, l-1\}$, and it holds that

$$\boldsymbol{\phi}^i(\varphi_l^{\setminus i}) - \boldsymbol{\phi}^i(\varphi_1^{\setminus i}) = \sum\nolimits_{s=1}^{l-1} K^i(0.5(\varphi_s^{\setminus i} + \varphi_{s+1}^{\setminus i}))(F_c^{\setminus i})^T(\varphi_{s+1}^{\setminus i} - \varphi_s^{\setminus i}). \tag{6.23}$$

In other words, the optimizer can be obtained by constructing a line between $\varphi_1^{\setminus i}$ and $\varphi_l^{\setminus i}$ and considering the matrices K^i of the regions $\mathcal{R}$ intersected by that line. Furthermore, it directly follows from (6.23) and the triangle inequality that

$$\begin{aligned}
\|(H^{i,i})^{\frac{1}{2}}(\boldsymbol{\phi}^i(\varphi_l^{\setminus i}) - \boldsymbol{\phi}^i(\varphi_1^{\setminus i}))\| &= \|\sum_{s=1}^{l-1}(H^{i,i})^{\frac{1}{2}}K^i(0.5(\varphi_s^{\setminus i} + \varphi_{s+1}^{\setminus i}))(F_c^{\setminus i})^T(\varphi_{s+1}^{\setminus i} - \varphi_s^{\setminus i})\|, \\
\leq \sum\nolimits_{s=1}^{l-1}\|(H^{i,i})^{\frac{1}{2}}K^i(0.5(\varphi_s^{\setminus i} + \varphi_{s+1}^{\setminus i}))(H^{i,i})^{\frac{1}{2}}\| &\|(H^{i,i})^{-\frac{1}{2}}(F_c^{\setminus i})^T(\varphi_{s+1}^{\setminus i} - \varphi_s^{\setminus i})\|.
\end{aligned}$$

Because $K^i(\varphi^{\setminus i})$ and $(H^{i,i})^{\frac{1}{2}}$ are symmetric and positive semidefinite (see the proof of Theorem 6.1), it follows that for all feasible $\varphi^{\setminus i}$ the norm is given by the largest eigenvalue (cf. [55]):

$$\|(H^{i,i})^{\frac{1}{2}}K^i(\varphi^{\setminus i})(H^{i,i})^{\frac{1}{2}}\| = \lambda_{\max}((H^{i,i})^{\frac{1}{2}}K^i(\varphi^{\setminus i})(H^{i,i})^{\frac{1}{2}}).$$

Based on Theorem 6.1 it holds that $K^i(\varphi^{\setminus i}) \preceq \Xi^i$ for all feasible $\varphi^{\setminus i}$, and with $\Xi^i = (2H^{i,i})^{-1}$ it follows that

$$\|(H^{i,i})^{\frac{1}{2}}K^i(\varphi^{\setminus i})(H^{i,i})^{\frac{1}{2}}\| \leq \|(H^{i,i})^{\frac{1}{2}}\Xi^i(H^{i,i})^{\frac{1}{2}}\| \leq 0.5.$$

Overall this results in

$$\begin{aligned}
&\|(H^{i,i})^{\frac{1}{2}}(\boldsymbol{\phi}^i(\varphi_l^{\setminus i}) - \boldsymbol{\phi}^i(\varphi_1^{\setminus i}))\| \leq 0.5\sum\nolimits_{s=1}^{l-1}\|(H^{i,i})^{-\frac{1}{2}}(F_c^{\setminus i})^T(\varphi_{s+1}^{\setminus i} - \varphi_s^{\setminus i})\|, \\
&\quad \leq 0.5\sum\nolimits_{s=1}^{l-1}\|(H^{i,i})^{-\frac{1}{2}}(F_c^{\setminus i})^T(\varphi_{s+1}^{\setminus i} - \varphi_s^{\setminus i})\|, \\
&\quad \leq 0.5\sum\nolimits_{s=1}^{l-1}\|(H^{i,i})^{-\frac{1}{2}}(F_c^{\setminus i})^T(\varphi_1^{\setminus i} + \theta_{s+1}(\varphi_l^{\setminus i} - \varphi_1^{\setminus i}) - \varphi_1^{\setminus i} - \theta_s(\varphi_l^{\setminus i} - \varphi_1^{\setminus i}))\|, \\
&\quad \leq 0.5\sum\nolimits_{s=1}^{l-1}\|(H^{i,i})^{-\frac{1}{2}}(F_c^{\setminus i})^T((\theta_{s+1} - \theta_s)(\varphi_l^{\setminus i} - \varphi_1^{\setminus i}))\|, \\
&\quad \leq 0.5\sum\nolimits_{s=1}^{l-1}(\theta_{s+1} - \theta_s)\|(H^{i,i})^{-\frac{1}{2}}(F_c^{\setminus i})^T(\varphi_l^{\setminus i} - \varphi_1^{\setminus i})\|, \\
&\quad \leq \|0.5(H^{i,i})^{-\frac{1}{2}}(F_c^{\setminus i})^T(\varphi_l^{\setminus i} - \varphi_1^{\setminus i})\|.
\end{aligned}$$

□

This results directly allows to analyze the sensitivity of the local optimizer with respect to different parameter vectors $\varphi_l^{\setminus i}$ and $\varphi_1^{\setminus i}$. In particular, given the difference between $\varphi_l^{\setminus i}$ and $\varphi_1^{\setminus i}$ the difference in the resulting optimizers $\boldsymbol{\phi}^i(\varphi_l^{\setminus i})$ and $\boldsymbol{\phi}^i(\varphi_1^{\setminus i})$ can be bounded without explicitly computing the optimizers. For instance, if $\varphi_l^{\setminus i}$ is the true value of the parameter vector and $\varphi_1^{\setminus i}$ is the value used by controller $\mathcal{C}^i$ the previous result allows to give a bound on the resulting difference in the local optimizer. In the next Section, this result is used to derive a triggering function for communication events which ensures that the difference in the locally optimized input sequences remains below a given bound.

6.3. Communication Events

Based on the sensitivity analysis in the previous section, triggering conditions for a communication event between a pair of controllers can be constructed which guarantee that the impact of the event-based communication on the convergence and closed-loop performance of the cooperative distributed MPC is bounded. To this end let $\hat{\varphi}_{k,p}^{i,j}$ denote the value assumed by $\mathcal{C}^j$ for $\varphi_{k,p}^j$ and let $\delta\hat{\varphi}_{k,p}^{i,j} := \hat{\varphi}_{k,p}^{i,j} - \varphi_{k,p}^j$ denote the corresponding difference. At the end of each iteration (k,p), the following initialization is performed for $(k,p+1)$:

$$(\hat{\varphi}_{k,p+1}^{i,j,-}; \xi_{k,p+1}^{j,i,-}) = (\hat{\varphi}_{k,p}^{i,j}; \xi_{k,p}^{j,i}), \tag{6.24}$$

where $(\cdot)^-$ denotes a value before an update through communication. Because $\hat{\varphi}_{k,p}^{i,j}$ is a local value of $\mathcal{C}^i$, the vector $\xi_{k,p}^{j,i}$ is used by $\mathcal{C}^j$ to keep track of $\hat{\varphi}_{k,p}^{i,j}$. In other words, if a communication event is triggered both $\xi_{k,p}^{j,i}$ and $\hat{\varphi}_{k,p}^{i,j}$ are updated such that $\xi_{k,p}^{i,j} = \hat{\varphi}_{k,p}^{i,j}$, i.e. $\mathcal{C}^j$ stores the last value communicated to $\mathcal{C}^i$. However, if the communication network induces uncertainties, $\xi_{k,p}^{i,j} = \hat{\varphi}_{k,p}^{i,j}$ may no longer hold (see Section 6.5).

At the beginning of each iteration (k,p), i.e. before the local optimization problem is solved, the current values are communicated from controller $\mathcal{C}^j$ to controller $\mathcal{C}^i$ if the triggering condition $\|\mathscr{M}^{i,j}(\xi_{k,p}^{j,i,-} - \varphi_{k,p}^j)\| > \gamma^i$ holds:

$$(\hat{\varphi}_{k,p}^{i,j}; \xi_{k,p}^{j,i}) := \begin{cases} (\varphi_{k,p}^j; \varphi_{k,p}^j) & \text{if } \|\mathscr{M}^{i,j}(\xi_{k,p}^{j,i,-} - \varphi_{k,p}^j)\| > \gamma^i \\ (\hat{\varphi}_{k,p}^{i,j,-}; \xi_{k,p}^{j,i,-}) & \text{otherwise} \end{cases}, \tag{6.25}$$

where $\mathscr{M}^{i,j} = 0.5(H^{i,i})^{-\frac{1}{2}}(F_c^{j,i})^T$. In other words, if $\hat{\varphi}_{k,0}^{i,j,-} = \xi_{k,0}^{j,i,-}$ and the communication network does not induce any uncertainties it holds for all $p \in \mathbb{N}_0$ that $\hat{\varphi}_{k,p}^{i,j} = \xi_{k,p}^{j,i}$. In terms of the errors, this results in $\|\mathscr{M}^{i,j}\delta\hat{\varphi}_{k,p}^{i,j}\| = 0$ if communication occurs and $\|\mathscr{M}^{i,j}\delta\hat{\varphi}_{k,p}^{i,j}\| \leq \gamma^i$ otherwise. Thus, as shown in the following theorem the difference between the costs resulting from applying the iteration (6.5) (i.e. using full time-triggered communication) and the iteration (6.3) (i.e. using the communication events (6.25)) is bounded.

Theorem 6.3. *Using the communication events* (6.25) *with thresholds* $\gamma^i = \frac{\gamma}{card(\mathcal{N}^i)}$, $\gamma \in \mathbb{R}_{\geq 0}$, *and update* (6.25), *it holds for all* $p > 0$, *all* $x_k \in \mathcal{X}$, *and compact sets* $\mathcal{X}$ *and* $\mathbf{U}$ *that*

$$V(x_k, \bar{\mathbf{v}}^i_{k,p}) - V(x_k, \bar{\mathbf{u}}^i_{k,p}) \leq \bar{V}_e := c_e\gamma + \gamma^2,$$

where $c_e \in \mathbb{R}_{>0}$ *is given by*

$$c_e \geq \max_{x\in\mathcal{X},\mathbf{u}\in\mathbf{U}} \|x^T(F^{1,i};\dots;F^{N_s,i})(H^{i,i})^{-\frac{1}{2}} + 2\mathbf{u}^T(H^{1,i};\dots;H^{N_s,i})(H^{i,i})^{-\frac{1}{2}}\|$$

Proof. Applying Theorem 6.2 and the triangle inequality to $\|(H^{i,i})^{\frac{1}{2}}(\boldsymbol{\phi}^i_{k,p} - \boldsymbol{\rho}^i_{k,p})\|$ results in

$$\|(H^{i,i})^{\frac{1}{2}}(\boldsymbol{\phi}^i_{k,p} - \boldsymbol{\rho}^i_{k,p})\| \leq \textstyle\sum_{j\in\mathcal{N}^i}\|\mathscr{M}^{i,j}\delta\hat{\varphi}^{i,j}_{k,p}\| \leq \gamma, \tag{6.26}$$

where the last inequality follows from the updates (6.25) and $\gamma^i = \frac{\gamma}{\mathrm{card}(\mathcal{N}^i)}$. Applying Taylor's theorem to $V(x_k, \bar{\mathbf{u}}_{k,p})$ results in

$$\begin{aligned}
V(x_k, \bar{\mathbf{v}}^i_{k,p}) - V(x_k, \bar{\mathbf{u}}^i_{k,p}) &= \nabla V(x_k, \bar{\mathbf{u}}^i_{k,p})^T(\bar{\mathbf{v}}^i_{k,p} - \bar{\mathbf{u}}^i_{k,p}) + (\bar{\mathbf{v}}^i_{k,p} - \bar{\mathbf{u}}^i_{k,p})^T H(\bar{\mathbf{v}}^i_{k,p} - \bar{\mathbf{u}}_{k,p}),\\
&= \nabla V(x_k, \bar{\mathbf{v}}^i_{k,p})^T(0;\dots;\boldsymbol{\phi}^i_{k,p} - \boldsymbol{\rho}^i_{k,p};\dots;0) + (\boldsymbol{\phi}^i_{k,p} - \boldsymbol{\rho}^i_{k,p})^T H^{i,i}(\boldsymbol{\phi}^i_{k,p} - \boldsymbol{\rho}^i_{k,p}),\\
&\leq c_e\|(H^{i,i})^{\frac{1}{2}}(\boldsymbol{\phi}^i_{k,p} - \boldsymbol{\rho}^i_{k,p})\| + \|(H^{i,i})^{\frac{1}{2}}(\boldsymbol{\phi}^i_{k,p} - \boldsymbol{\rho}^i_{k,p})\|^2,\\
&\leq c_e\gamma + \gamma^2 =: \bar{V}_e,
\end{aligned}$$

for all $p > 0$. Furthermore, the maximizer c_e is attained because $\mathcal{X}$ and $\mathbf{U}$ are assumed to be compact (cf. Theorem 2.2). □

It should be noted that $\bar{V}_e$ can be made arbitrarily small by choosing a suitable threshold $\gamma \in \mathbb{R}_{\geq 0}$. In particular, a smaller threshold γ results in a smaller bound $\bar{V}_e$, but in general also results in a higher load on the communication network. Theorem 6.3 only bounds the resulting cost difference from performing one iteration using the communication events (6.25) instead of full communication, but does not bound the suboptimality arising from performing multiple iterations. Therefore, the next theorem investigates the influence on the convergence of the overall algorithm.

Theorem 6.4. *If the assumptions of Theorem 6.3 hold, the cost difference to the optimum converges over the iterations* p *according to*

$$V_d(x_k, \mathbf{v}_{k,p}) \leq \bar{\beta}^p V_d(x_k, \mathbf{v}_{k,0}) + \frac{1-\bar{\beta}^p}{1-\bar{\beta}}\bar{V}_e,$$

and in the limit the solution converges to $\lim_{p\to\infty} V_d(x_k, \mathbf{v}_{k,p}) \leq (1-\bar{\beta})^{-1}\bar{V}_e$.

Proof. It follows from Proposition 4.3 that

$$V(x_k, \bar{\mathbf{u}}^i_{k,p}) \leq V(x_k, \mathbf{u}_{k,p}) - \frac{(\mu^i_{k,p})^2}{4c^i_H} V_d(x_k, \mathbf{u}_{k,p}). \tag{6.27}$$

Next, summation over all $i \in \mathcal{N}$, subtracting $V^*(x_k)$ on both sides, and substituting $w^i_{k,p}$ and $\mu^i_{k,p}$ from Theorem 4.2 results in

$$\textstyle\sum_{i\in\mathcal{N}} w^i V(x_k, \bar{\mathbf{u}}^i_{k,p}) - V^*(x_k) \leq \bar{\beta} V_d(x_k, \mathbf{v}_{k,p}). \tag{6.28}$$

Furthermore, it follows from Theorem 6.3 and the convexity of the cost $V(x_k, \mathbf{v}_{k,p})$ that

$$V(x_k, \mathbf{v}_{k,p}) \leq \textstyle\sum_{i\in\mathcal{N}} w^i V(x_k, \bar{\mathbf{v}}^i_{k,p}) \leq \sum_{i\in\mathcal{N}} w^i V(x_k, \bar{\mathbf{u}}^i_{k,p}) + \bar{V}_e. \tag{6.29}$$

Considering (6.28), this results in

$$V_d(x_k, \mathbf{v}_{k,p+1}) \leq \textstyle\sum_{i\in\mathcal{N}} w^i V(x_k, \bar{\mathbf{v}}^i_{k,p}) - V^*(x_k) \leq \bar{\beta} V_d(x_k, \mathbf{v}_{k,p}) + \bar{V}_e. \tag{6.30}$$

Applying this inequality iteratively from $(k, 0)$ to (k, p) results in

$$V_d(x_k, \mathbf{v}_{k,p}) \leq \bar{\beta}^p V_d(x_k, \mathbf{v}_{k,0}) + \textstyle\sum_{l=0}^{p-1} \bar{\beta}^l \bar{V}_e = \bar{\beta}^p V_d(x_k, \mathbf{v}_{k,0}) + \dfrac{1-\bar{\beta}^p}{1-\bar{\beta}} \bar{V}_e,$$

for all $p > 1$ and with $0 \leq \bar{\beta} < 1$ the theorem directly follows. □

In conclusion, using the communication events (6.25), the edges of the time-varying communication graph $\mathscr{C}_{k,p}$ are given by

$$\mathcal{E}_{k,p} = \left\{ (i,j) \in \mathcal{N} \times (\mathcal{N} \setminus i) \,\middle|\, \|\mathscr{M}^{j,i}(\xi^{i,j,-}_{k,p} - \varphi^i_{k,p})\| > \gamma^j \right\}, \tag{6.31}$$

and the impact of the communication events on the cooperative distributed MPC can be made arbitrarily small by choosing a suitable threshold $\gamma \in \mathbb{R}_{\geq 0}$. Another question is when to terminate the iterative algorithm in order to guarantee bounded suboptimality.

Stopping Condition

Terminating the algorithm early (i.e. before convergence of $V_d(x_k, \mathbf{v}_{k,p})$ below a bound) may result in strongly suboptimal solutions. Thus, a stopping criterion which bounds the suboptimality of the input sequence obtained from the distributed MPC may be required. Let $\mathcal{T}_{k,p} \subseteq \mathcal{N}$ denote the set of controllers which communicate at time (k, p):

$$\mathcal{T}_{k,p} := \{ i \in \mathcal{N} \,|\, \exists j \in \mathcal{N} : (i,j) \in \mathcal{E}_{k,p} \}. \tag{6.32}$$

A simple choice would be to assume $p \geq 1$ and terminate the scheme if $\mathcal{T}_{k,p} = \emptyset$ for any $p \geq 1$. However, this stopping condition cannot be verified by every controller $\mathcal{C}^i$ because the communication is no longer global when a communication event is triggered. Furthermore, $\mathcal{T}_{k,p} = \emptyset$ in combination with the communication event (6.25) is, in general, not suitable to guarantee a certain degree of suboptimality. To begin with, a lack of communication by a controller $\mathcal{C}^i$ may not have any implications with respect to the local costs optimized by $\mathcal{C}^i$, because $(\mathscr{M}^{i,j})^T\mathscr{M}^{i,j} \succeq 0$ may not be strictly positive definite. Furthermore, $\mathcal{T}_{k,\bar{p}_k} = \emptyset$ does not guarantee that $\mathcal{T}_{k,p} \neq \emptyset$ holds for all $p > \bar{p}_k$, because (6.3) results in an updated input sequence.

In other words, the triggering condition ensures that the information required to compute the inputs is available to each controller, but is not suitable to bound the suboptimality of the algorithm. The following preliminary results are required to derive a stopping criterion which implies bounded suboptimality.

Proposition 6.3. *The suboptimality $V_d(x_k, \mathbf{v}_{k,p})$ is bounded by*

$$V_d(x_k, \mathbf{v}_{k,p_k}) \leq (1-\bar{\beta})^{-1} \left(\sum\nolimits_{i\in\mathcal{N}} w^i \left(V(x_k, \mathbf{v}_{k,p_k}) - V(x_k, \bar{\mathbf{v}}^i_{k,p_k}) \right) + \bar{V}_e \right),$$

Proof. Based on (6.30) it holds that

$$\sum\nolimits_{i\in\mathcal{N}} w^i V(x_k, \bar{\mathbf{v}}^i_{k,p}) - V^*(x_k) \leq V_d(x_k, \mathbf{v}_{k,p}) + (\bar{\beta}-1)V_d(x_k, \mathbf{v}_{k,p}) + \bar{V}_e, \tag{6.33}$$

$$\sum\nolimits_{i\in\mathcal{N}} w^i V(x_k, \bar{\mathbf{v}}^i_{k,p}) \leq V(x_k, \mathbf{v}_{k,p}) + (\bar{\beta}-1)V_d(x_k, \mathbf{v}_{k,p}) + \bar{V}_e, \tag{6.34}$$

$$(1-\bar{\beta})V_d(x_k, \mathbf{v}_{k,p}) \leq V(x_k, \mathbf{v}_{k,p}) - \sum\nolimits_{i\in\mathcal{N}} w^i V(x_k, \bar{\mathbf{v}}^i_{k,p}) + \bar{V}_e, \tag{6.35}$$

$$(1-\bar{\beta})V_d(x_k, \mathbf{v}_{k,p}) \leq \sum\nolimits_{i\in\mathcal{N}} w^i (V(x_k, \mathbf{v}_{k,p}) - V(x_k, \bar{\mathbf{v}}^i_{k,p})) + \bar{V}_e. \tag{6.36}$$

The proposition follows by solving (6.36) for $V_d(x_k, \mathbf{v}_{k,p})$. □

This result shows that a bound on the suboptimality $V_d(x_k, \mathbf{v}_{k,p})$ can be obtained by checking the weighted sum of the differences $V(x_k, \mathbf{v}_{k,p}) - V(x_k, \bar{\mathbf{v}}^i_{k,p})$. However, this requires global communication.

In the next theorem a stopping condition which is based on the local costs $V^i(x^i_k, \mathbf{v}^i_{k,\tilde{p}_k}, \hat{\varphi}^{\setminus i}_{k,\tilde{p}_k})$ is given. This condition can be checked by using local computations and limited global communication.

Theorem 6.5. *If the condition*

$$V^i(x^i_k, \mathbf{v}^i_{k,p}, \hat{\varphi}^{\setminus i}_{k,p}) - V^i(x^i_k, \boldsymbol{\phi}^i_{k,p}, \hat{\varphi}^{\setminus i}_{k,p}) \leq \zeta - \gamma \|(\mathbf{v}^i_{k,p} - \boldsymbol{\phi}^i_{k,p})^T 2(H^{i,i})^{\frac{1}{2}}\|, \tag{6.37}$$

holds for $\zeta \in \mathbb{R}_{>0}$ and all $i \in \mathcal{N}$, the suboptimality is bounded by

$$V_d(x_k, \mathbf{v}_{k,p}) \leq (1-\bar{\beta})^{-1} \left(\zeta + \bar{V}_e \right) =: \bar{V}_d.$$

Proof. By construction of $V^i(x^i_k, \mathbf{v}^i_{k,p}, \varphi^{\setminus i}_{k,p})$ it holds that

$$V(x_k, \mathbf{v}_{k,p}) - V(x_k, \bar{\mathbf{v}}^i_{k,p}) = V^i(x^i_k, \mathbf{v}^i_{k,p}, \varphi^{\setminus i}_{k,p}) - V^i(x^i_k, \boldsymbol{\phi}^i_{k,p}, \varphi^{\setminus i}_{k,p}).$$

However, locally only $\hat{\varphi}_{k,p}^{\backslash i}$ is available to compute the costs. This results in

$$
\begin{aligned}
&V^i(x_k^i, \mathbf{v}_{k,p}^i, \hat{\varphi}_{k,p}^{\backslash i}) - V^i(x_k^i, \boldsymbol{\phi}_{k,p}^i, \hat{\varphi}_{k,p}^{\backslash i}) && (6.38)\\
&= V^i(x_k^i, \mathbf{v}_{k,p}^i, \varphi_{k,p}^{\backslash i}) - V^i(x_k^i, \boldsymbol{\phi}_{k,p}^i, \varphi_{k,p}^{\backslash i}) + (\delta\hat{\varphi}_{k,p}^{\backslash i})^T F_c^{\backslash i}(\mathbf{v}_{k,p}^i - \boldsymbol{\phi}_{k,p}^i), && (6.39)\\
&= V(x_k, \mathbf{v}_{k,p}) - V(x_k, \bar{\mathbf{v}}_{k,p}^i) + (\delta\hat{\varphi}_{k,p}^{\backslash i})^T F_c^{\backslash i}(\mathbf{v}_{k,p}^i - \boldsymbol{\phi}_{k,p}^i). && (6.40)
\end{aligned}
$$

Considering the triggering condition (6.25), it holds that

$$
\begin{aligned}
\|(\delta\hat{\varphi}_{k,p}^{i,j})^T F_c^{j,i}(\mathbf{v}_{k,p}^i - \boldsymbol{\phi}_{k,p}^i)\| &= \|(\mathbf{v}_{k,p}^i - \boldsymbol{\phi}_{k,p}^i)^T 2(H^{i,i})^{\frac{1}{2}} 0.5(H^{i,i})^{-\frac{1}{2}}(F_c^{j,i})^T \delta\hat{\varphi}_{k,p}^{i,j}\|,\\
&\leq \|(\mathbf{v}_{k,p}^i - \boldsymbol{\phi}_{k,p}^i)^T 2(H^{i,i})^{\frac{1}{2}}\|\|\mathscr{M}^{i,j}\delta\hat{\varphi}_{k,p}^{i,j}\|,\\
&\leq \gamma^i\|(\mathbf{v}_{k,p}^i - \boldsymbol{\phi}_{k,p}^i)^T 2(H^{i,i})^{\frac{1}{2}}\|.
\end{aligned}
$$

Next, by definition of γ^i it holds for all $i \in \mathcal{N}$ that

$$
\begin{aligned}
\|(\delta\hat{\varphi}_{k,p}^{\backslash i})^T F_c^{\backslash i}(\mathbf{v}_{k,p}^i - \boldsymbol{\phi}_{k,p}^i)\| &\leq \textstyle\sum_{j\in\mathcal{N}^i} \gamma^i\|(\delta\hat{\varphi}_{k,p}^{i,j})^T F_c^{j,i}(\mathbf{v}_{k,p}^i - \boldsymbol{\phi}_{k,p}^i)\|, && (6.41)\\
&\leq \gamma\|(\mathbf{v}_{k,p}^i - \boldsymbol{\phi}_{k,p}^i)^T 2(H^{i,i})^{\frac{1}{2}}\|. && (6.42)
\end{aligned}
$$

Based on this it follows from (6.40) that

$$
\begin{aligned}
V(x_k, \mathbf{v}_{k,p}) - V(x_k, \bar{\mathbf{v}}_{k,p}^i) \leq &V^i(x_k^i, \mathbf{v}_{k,p}^i, \hat{\varphi}_{k,p}^{\backslash i}) - V^i(x_k^i, \boldsymbol{\phi}_{k,p}^i, \hat{\varphi}_{k,p}^{\backslash i}) + && (6.43)\\
&\gamma\|(\mathbf{v}_{k,p}^i - \boldsymbol{\phi}_{k,p}^i)^T 2(H^{i,i})^{\frac{1}{2}}\|.
\end{aligned}
$$

If (6.37) holds, this results in

$$
V(x_k, \mathbf{v}_{k,p}) - V(x_k, \bar{\mathbf{v}}_{k,p}^i) \leq \zeta, \tag{6.44}
$$

and the theorem follows by substituting this bound into the bound given in Proposition 6.3 and noting that $\sum_{i\in\mathcal{N}} w^i\zeta = \zeta$. □

In the following, the condition (6.37) for all $i \in \mathcal{N}$ is used as a stopping criterion if a bound on the suboptimality is required. It can be seen that this stopping criterion cannot be checked without some form of centralized coordination or global communication. This could have been expected because the controllers cooperate, perform the iterations synchronously, and the suboptimality is a property of the cost of the overall system. The stopping criterion may be checked by communicating with one controller which acts as a centralized coordinator. Alternatively, network wide arbitration [20] can be used used to check whether the stopping condition (6.37) holds for all controllers.

Furthermore, if the stopping condition holds for $\tilde{p}_k \geq 0$ it follows that $V_d(x_k, \mathbf{v}_{k,\tilde{p}_k}) \leq (1-\bar{\beta})^{-1}(\zeta + \bar{V}_e)$. Substituting this bound into $V_d(x_k, \mathbf{v}_{k,p+1}) \leq \bar{\beta}V_d(x_k, \mathbf{v}_{k,p}) + \bar{V}_e$ results in $V_d(x_k, \mathbf{v}_{k,\tilde{p}_k+1}) \leq (1-\bar{\beta})^{-1}(\bar{\beta}\zeta + \bar{V}_e)$. It follows that the algorithm can be terminated at any iteratation $p \geq \tilde{p}_k$ and the bound $V_d(x_k, \mathbf{v}_{k,p}) \leq \bar{V}_d$ still holds.

Another aspect is whether or not this stopping condition is guaranteed to hold for a finite number of iterations.

Proposition 6.4. *If* $\zeta > 2\gamma\|(\mathbf{v}^i - \boldsymbol{\phi}^i)^T 2(H^{i,i})^{\frac{1}{2}}\|$ *holds for all* $\mathbf{v}^i \in \mathbf{U}^i := \mathbb{U}^i \times \ldots \times \mathbb{U}^i$*, and all* $\boldsymbol{\phi}^i$*, the stopping condition* (6.37) *holds after a finite number of iterations.*

Proof. Because of $V^*(x_k) \leq \sum_{i\in\mathcal{N}} w^i V(x_k, \bar{\mathbf{v}}^i_{k,p})$ and the convergence result given in Theorem 6.4 the cost converges if $V_d(x_k, \mathbf{v}_{k,p}) > (1-\bar{\beta})^{-1}\bar{V}_e$ but cannot decrease below $V^*(x_k)$. This implies that

$$\sum\nolimits_{i\in\mathcal{N}} w^i(V(x_k, \mathbf{v}_{k,p}) - V(x_k, \bar{\mathbf{v}}^i_{k,p})) \leq c_\zeta \tag{6.45}$$

holds after a finite number of iterations for $c_\zeta \in \mathbb{R}_{>0}$. In other words, either the convergence slows below this bound or the cost may increase because $V_d(x_k, \mathbf{v}_{k,p}) < (1-\bar{\beta})^{-1}\bar{V}_e$ holds.

By the same arguments employed in the proof of Theorem 6.5, (6.45) implies that

$$\sum\nolimits_{i\in\mathcal{N}} w^i(V^i(x^i_k, \mathbf{v}^i_{k,p}, \hat{\varphi}^{\backslash i}_{k,p}) - V^i(x^i_k, \boldsymbol{\phi}^i_{k,p}, \hat{\varphi}^{\backslash i}_{k,p})) \leq c_\zeta + \gamma\|(\mathbf{v}^i_{k,p} - \boldsymbol{\phi}^i_{k,p})^T 2(H^{i,i})^{\frac{1}{2}}\|.$$

Furthermore, optimality of $\boldsymbol{\phi}^i_{k,p}$ with respect to the local optimization problem (6.1) implies that $0 \leq V^i(x^i_k, \mathbf{v}^i_{k,p}, \hat{\varphi}^{\backslash i}_{k,p}) - V^i(x^i_k, \boldsymbol{\phi}^i_{k,p}, \hat{\varphi}^{\backslash i}_{k,p})$ holds. Therefore, after a finite number of iterations it holds that

$$V^i(x^i_k, \mathbf{v}^i_{k,p}, \hat{\varphi}^{\backslash i}_{k,p}) - V^i(x^i_k, \boldsymbol{\phi}^i_{k,p}, \hat{\varphi}^{\backslash i}_{k,p}) \leq c_\zeta + \gamma\|(\mathbf{v}^i_{k,p} - \boldsymbol{\phi}^i_{k,p})^T 2(H^{i,i})^{\frac{1}{2}}\|. \tag{6.46}$$

In turn, this implies that $V^i(x^i_k, \mathbf{v}^i_{k,p}, \hat{\varphi}^{\backslash i}_{k,p}) - V^i(x^i_k, \boldsymbol{\phi}^i_{k,p}, \hat{\varphi}^{\backslash i}_{k,p}) \leq \zeta - \gamma\|(\mathbf{v}^i_{k,p} - \boldsymbol{\phi}^i_{k,p})^T 2(H^{i,i})^{\frac{1}{2}}\|$ holds for $\zeta > 2\gamma\|(\mathbf{v}^i_{k,p} - \boldsymbol{\phi}^i_{k,p})^T 2(H^{i,i})^{\frac{1}{2}}\|$ and the proposition directly follows. □

It can be seen that this result may be very conservative because, even though $(\mathbf{v}^i_{k,p} - \boldsymbol{\phi}^i_{k,p})$ is usually small at convergence, the bound $\zeta > 2\gamma\|(\mathbf{v}^i_{k,p} - \boldsymbol{\phi}^i_{k,p})^T 2(H^{i,i})^{\frac{1}{2}}\|$ has to hold for all $\mathbf{v}^i$ and $\boldsymbol{\phi}^i$. Because the corresponding sets are compact such a ζ exists, but it may be very large, resulting in a very conservative bound on the suboptimality. Another reason for the conservatism of this bound, as well as the bounds given in Theorem 6.4 and Theorem 6.5 is that in the worst-case all triggering conditions may just be below their respective thresholds. However, in practice this scenario is very unlikely, especially if the number of subsystems is large.

6.4. Stability Analysis

Stability Using Terminal Costs and Terminal Constraints

For Case 2.3 (coupled costs, decoupled dynamics and constraints) it is assumed that Assumption 5.2 holds and the algorithm may be terminated at any iteration $0 \leq \bar{p}_k \leq p_{\max}$, i.e. no stopping criterion is used. This is motivated by the fact that checking any suitable stopping criterion requires additional communication. At the same time, iterations in which no new information is received only require a low computational effort because (6.8) does not need to be solved again. As in the previous chapter, the initialization for $(k+1,0)$ is obtained based on the terminal control law $K_{\mathbb{T}}$. The overall cooperative distributed MPC scheme is given in Algorithm 6.1.

The following initialization is performed locally by each controller $\mathcal{C}^i$ for its local input sequence:

$$\mathbf{v}^i_{k+1,0} := (v^i_{k+1|k,\bar{p}_k}; \ldots; v^i_{k+N-1|k,\bar{p}_k}; K^i_{\mathbb{T}} x^i_{k+N|k,\bar{p}_k}), \tag{6.47}$$

where $x^i_{k+N|k,\bar{p}_k} = (A^i)^N x^i_k + \sum_{l=0}^{N-1} (A^i)^{N-1-l} B^i v^i_{k+l|k,\bar{p}_k}$. Similarly, the values assumed by $\mathcal{C}^i$ for all $j \in \mathcal{N} \setminus i$ are initialized as follows:

$$\hat{\varphi}^{i,j,-}_{k+1,0} := \mathscr{P}^j \hat{\varphi}^{i,j,-}_{k,\bar{p}_k}, \quad \xi^{j,i,-}_{k+1,0} := \mathscr{P}^j \xi^{j,i,-}_{k,\bar{p}_k}, \tag{6.48}$$

with the linear predictor $\mathscr{P}^j$ given by

$$\mathscr{P}^j := \begin{bmatrix} A^j & B^j & 0 & \cdots & 0 \\ 0 & 0 & I_{m^j} & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & I_{m^j} \\ K^j_{\mathbb{T}}(A^j)^N & K^j_{\mathbb{T}}(A^j)^{N-1}B^j & \cdots & \cdots & K^j_{\mathbb{T}}B^j \end{bmatrix}. \tag{6.49}$$

These computations can be carried out by all controllers without additional communication, and by keeping track of which values were communicated, each controller can check the triggering condition used in in (6.25).

Algorithm 6.1: distributed MPC with terminal constraint for each subsystem $\mathcal{P}^i$

Given γ, $k=0$, $p=0$, $\hat{\varphi}^{i,j}_{0,0} = \xi^{j,i}_{0,0}$, and $\mathbf{v}^{i,j}_{0,0}$ for all $j \in \mathcal{N} \setminus i$:
for $k \geq 0$ **do**
  while $p \leq p_{\max}$ **do**
    If $\|\mathscr{M}^{j,i}(\xi^{i,j,-}_{k,p} - \varphi^i_{k,p})\| > \gamma^j$ send $\varphi^i_{k,p}$ to $\mathcal{C}^j$
    Based on the communicated information update $\hat{\varphi}^{i,j}_{k,p}$ and $\xi^{j,i}_{k,p}$ according to (6.25)
    Solve (6.1) and compute $\mathbf{v}^i_{k,p+1}$ according to (6.3)
    Set $\hat{\varphi}^{i,j,-}_{k,p+1} = \hat{\varphi}^{i,j}_{k,p}$, $\xi^{j,i,-}_{k,p+1} = \xi^{j,i}_{k,p}$, for all $j \in \mathcal{N} \setminus i$, and set $p = p+1$
  end while
  Apply $u^i_k = v^i_{k|k,p}$ to the system (2.2)
  Compute the initialization for $k+1$ according to (6.47), (6.48) and set $(k,p) = (k+1,0)$.
end for

The following result states that practical stability of the closed-loop can be established for the extended state vector $\mathbf{z}_{k,p}$ with respect to the corresponding set $\mathbf{\Sigma} := \Sigma \times \ldots \times \Sigma$ if Assumption 5.2 holds (cf. Section 5.3).

Theorem 6.6. *Given a feasible initialization $\mathbf{z}_{0,0} \in \boldsymbol{\mathcal{Z}}$, the following holds for the distributed MPC given in Algorithm 6.1 and Case 2.3 at any iteration $0 \leq p \leq p_{\max}$ if Assumption 5.2 holds:*

(I) the distributed MPC problem is feasible for all $k \geq 0$,

(II) the closed-loop is practically stable with respect to $\mathbf{\Sigma}$.

Proof. Recursive feasibility can be established by showing that the initialization (6.47) is feasible for $(k+1, 0)$ and that feasibility is preserved at every iteration (see the proof of Theorem 5.6). These arguments hold for Algorithm 6.1 because the constraints are purely local and only the non-local information is uncertain. Therefore, it holds that $x_{k+l} \in \mathcal{X} = \mathbb{X}$ for all $l > 0$, if $x_k \in \mathbb{X}$.

Furthermore, it is established in the proof of Theorem 5.6 that the cost is decreasing if no iterations take place at $k + l$ (i.e. $\bar{p}_{k+l} = 0$) for all $l \in \mathbb{N}_0$:

$$\mathbf{V}_z(\mathbf{z}_{k+1,0}) - \mathbf{V}_z(\mathbf{z}_{k,\bar{p}_k}) \leq -\|x_{k|k,\bar{p}_k}\|_Q^2 - \|u_{k|k,\bar{p}_k}\|_R^2. \tag{6.50}$$

Applying this inequality and the initialization (6.47) for $N-1$ time steps results in

$$\mathbf{V}_z(\mathbf{z}_{k+N,0}) - \mathbf{V}_z(\mathbf{z}_{k,\bar{p}_k}) \leq -\sum_{l=0}^{N-1}(\|x_{k+l|k,\bar{p}_k}\|_Q^2 + \|u_{k+l|k,\bar{p}_k}\|_R^2) \tag{6.51}$$

because applying (6.47) implies that $u_{k+l|k+l,0} = u_{k+l|k,0}$, $x_{k+l|k+l,0} = x_{k+l|k,0}$ for all $l \in \{0, \ldots, N-1\}$. Next, Theorem 6.4 and $\bar{p}_k \leq p_{\max}$ for all $k \geq 0$ imply that $\mathbf{V}_z(\mathbf{z}_{k+N,\bar{p}_{k+N}}) \leq \mathbf{V}(\mathbf{z}_{k+N,0}) + \frac{1-\bar{\beta}^{p_{\max}}}{1-\beta}\bar{V}_e$ and it holds that

$$\mathbf{V}_z(\mathbf{z}_{k+N,\bar{p}_{k+N}}) - \mathbf{V}_z(\mathbf{z}_{k,\bar{p}_k}) \leq -\alpha_3(\|\mathbf{z}_{k,\bar{p}_k}\|_{\mathbf{\Sigma}}) + N\frac{1-\bar{\beta}^{p_{\max}}}{1-\bar{\beta}}\bar{V}_e.$$

Finally, as shown in Theorem 5.6 there exist $\alpha_1(\|\mathbf{z}\|_{\mathbf{\Sigma}})$ and $\alpha_2(\|\mathbf{z}\|_{\mathbf{\Sigma}})$ such that $\alpha_1(\|\mathbf{z}\|_{\mathbf{\Sigma}}) \leq \mathbf{V}_z(\mathbf{z}) \leq \alpha_2(\|\mathbf{z}\|_{\mathbf{\Sigma}})$ holds. In other words $\mathbf{V}_z(\mathbf{z}_{k,p})$ can be used as ISpS-Lyapunov function and the conditions of Theorem 2.1 hold for $L = N$, $d_1 = 0$, $d_2 = \frac{1-\bar{\beta}^{p_{\max}}}{1-\bar{\beta}}\bar{V}_e$, and $\omega_k = 0$ for all $k \in \mathbb{N}_0$ which implies that the closed-loop is practically stable. □

Stability Without Terminal Costs or Terminal Constraints

For Case 2.4 (coupled costs and dynamics, no state or terminal constraints), the initialization for $(k+1, 0)$ is obtained as follows for all $j \in \mathcal{N} \setminus i$:

$$\hat{\varphi}_{k+1,0}^{i,j,-} := \mathscr{P}^j \hat{\varphi}_{k,\bar{p}_k}^{i,j,-},\ \xi_{k+1,0}^{j,i,-} := \mathscr{P}^j \xi_{k,\bar{p}_k}^{j,i,-},\ \mathbf{v}_{k+1,0}^i := (v_{k+1|k,\bar{p}_k}^i; \ldots; v_{k+N-1|k,\bar{p}_k}^i; 0) \tag{6.52}$$

and the following linear predictor $\mathscr{P}^j$ is chosen:

$$\mathscr{P}^j := \begin{bmatrix} I_{n^j} & 0 & 0 & \cdots & 0 \\ 0 & 0 & I_{m^j} & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & I_{m^j} \\ 0 & 0 & 0 & \cdots & 0 \end{bmatrix}. \tag{6.53}$$

The overall distributed MPC algorithm without terminal constraint includes a distributed stopping criterion and is given in Algorithm 6.2.

The linear predictor $\mathscr{P}^j$ used by each subsystem may have a strong influence on the number of messages and robustness of the algorithm with respect to communication uncertainties. For instance, the difference $\delta\hat{\varphi}^{i,j}_{k+1,0}$ and the resulting value of the triggering function is given by

$$\|\mathscr{M}^{i,j}\delta\hat{\varphi}^{i,j}_{k+1,0}\| = \|\mathscr{M}^{i,j}(\mathscr{P}^j\hat{\varphi}^{i,j}_{k,\bar{p}_k} - \mathscr{P}^j\varphi^j_{k,\bar{p}_k} + \mathscr{P}^j\varphi^j_{k,\bar{p}_k} - \varphi^j_{k+1,0})\|, \tag{6.54}$$

$$\leq \|\mathscr{M}^{i,j}\mathscr{P}^j\delta\hat{\varphi}^{i,j}_{k,\bar{p}_k}\| + \|\mathscr{M}^{i,j}(\mathscr{P}^j\varphi^j_{k,\bar{p}_k} - \varphi^j_{k+1,0})\|. \tag{6.55}$$

Obviously, $\mathscr{P}^j$ should be chosen such that both the first and second term remain as small as possible, such that the increase in the triggering function remains small from one time step to the next. However, there is a trade-off involved because a predictor which minimizes the increase in the second term may result in large increase in the first term. In particular if the predictor $\mathscr{P}^j$ is unstable, the first term depending on $\delta\hat{\varphi}^{i,j}_{k,\bar{p}_k}$ may grow very fast. The predictor given in (6.49) for the case of decoupled dynamics in conjunction with using a terminal constraint is both stable,

---

Algorithm 6.2: distributed MPC without terminal constraint for each subsystem $\mathcal{P}^i$

Given γ, ζ, $\text{stop}^i = 0$, $k = 0$, $p = 0$, $\hat{\varphi}^{i,j}_{0,0} = \xi^{j,i}_{0,0}$, and $\mathbf{v}^{i,j}_{0,0}$ for all $j \in \mathcal{N} \setminus i$:
for $k \geq 0$ **do**
  while $\text{stop}^i = 0$ for all $i \in \mathcal{N}$ **do**
    If $\|\mathscr{M}^{j,i}(\xi^{i,j,-}_{k,p} - \varphi^i_{k,p})\| > \gamma^j$ send $\varphi^i_{k,p}$ to $\mathcal{C}^j$
    Based on the communicated information update $\hat{\varphi}^{i,j}_{k,p}$ and $\xi^{j,i}_{k,p}$ according to (6.25)
    Solve (6.1) and compute $\mathbf{v}^i_{k,p+1}$ according to (6.3)
    If (6.37) holds set $\text{stop}^i = 1$
    Set $\hat{\varphi}^{i,j,-}_{k,p+1} = \hat{\varphi}^{i,j}_{k,p}$, $\xi^{j,i,-}_{k,p+1} = \xi^{j,i}_{k,p}$, for all $j \in \mathcal{N} \setminus i$, and set $p = p + 1$
  end while
  Apply $u^i_k = v^i_{k|k,p}$ to the system (2.2)
  Compute the initialization for $k + 1$ according to (6.52) and set $(k, p) = (k + 1, 0)$
end for

---

and using this predictor, the second term in (6.55) is equal to zero. In contrast, when not using a terminal constraint and considering fully coupled dynamics it is not obvious how to choose $\mathscr{P}^j$. Using the local part of the dynamics A^j and B^j in the predictor may result in an unstable $\mathscr{P}^j$, and it is not clear if this would result in slower increase of the triggering function. Furthermore, a more complicated predictor which considers the interconnected dynamics may either require larger messages (e.g. communication of predicted state trajectories), or have to consider the interconnections between the different subsystems by involving $\hat{\varphi}^{i,j}$ for all $j \in \mathcal{N}^i$ in the prediction. Both of these approaches complicate the analysis. At the same time, it is not obvious if they provide any improved bounds, because the additional information provided by these approaches is also not accurate and $\mathscr{P}^j$ may again be unstable.

In the following, it will be shown that the closed-loop system is practically stable by means of an ISpS-Lyapunov function.

Theorem 6.7. *Given $x_k \in \mathcal{X}$, $\mathcal{X}$ according to Assumption 5.3, and a feasible initial solution at time $k = 0$, the distributed MPC applied to Case 2.4 with the stopping criterion given by Theorem 6.5:*

(I) is recursively feasible,

(II) renders the system practically stable in $\mathcal{X}$ with respect to the origin.

Proof. According to Theorem 6.5 it holds that $\bar{V}_d = (1-\bar{\beta})^{-1}(\zeta+\bar{V}_e)$ for all $x_k \in \mathcal{X}$ if the stopping criterion $V^i(x^i_k, \mathbf{v}^i_{k,\tilde{p}_k}, \hat{\varphi}^{\setminus i}_{k,\tilde{p}_k}) - V^i(x^i_k, \boldsymbol{\phi}^i_{k,\tilde{p}_k}, \hat{\varphi}^{\setminus i}_{k,\tilde{p}_k}) \leq \zeta$ holds for all $i \in \mathcal{N}$. Based on this bound, a proof can be obtained by the same arguments made in Theorem 5.7. ☐

This result does not depend on $\mathscr{P}^i$. However, the robustness results given in the next section with respect to uncertain communication may strongly depend on $\mathscr{P}^i$.

6.5. Uncertain Communication

Due to effects such as communication delays and packet loss the difference $\delta\hat{\varphi}^{i,j}_{k,p}$ may become larger than allowed by the triggering conditions. Therefore, the impact of packet loss and communication delays can be modeled as an enlargement of the threshold γ, which will be denoted by $\gamma_d \geq \gamma$.

To this end, the following assumption is made with respect to the uncertainties induced by the communication network.

Assumption 6.2. *It is assumed that the communication between controllers is subject to a bounded time-varying communication delay. The delay, in number of iterations, of a message sent from $\mathcal{C}^i$ to $\mathcal{C}^j$ at time (k,p) is denoted by $\tau^{i,j}_{k,p} \in \mathbb{N}_0$. Messages which do not arrive after $\tau_{\max} \in \mathbb{N}_0$, $\tau_{\max} \leq p_{\max}$ iterations or before the* *next time step $(k+1,0)$ are considered lost, and the maximal number of consecutive packet losses between any pair of controllers $(\mathcal{C}^i, \mathcal{C}^j)$, $(i,j) \in \mathcal{N} \times (\mathcal{N} \setminus i)$ is bounded by $\tau_l \in \mathbb{N}_0$.*

This network model allows considering both packet loss and bounded delays smaller than the sampling time Δt. The restriction to the case of small delays is made because the algorithm considered in this chapter is iterative and, in contrast to the algorithm proposed in Chapter 3, does not explicitly consider communication delays in the local optimization. For instance, Algorithm 6.2 often requires multiple iterations and communication events per time step and is therefore ill suited to deal with communication delays. In this case, a network only affected by packet loss may be described by $\tau_{\max} = 0$ and $\tau_l \in \mathbb{R}_{>0}$. To analyze the effect of delays and packet loss let $(k_e^{i,j}, p_e^{i,j}) < (k_{e+1}^{i,j}, p_{e+1}^{i,j}) < \ldots < (k_{e+\tau_l+1}^{i,j}, p_{e+\tau_l+1}^{i,j})$ denote a sequence of times of communication events between $\mathcal{C}^j$ and $\mathcal{C}^i$, and $(k_1, p_1) < (k, p)$ denotes that (k,p) is a later point in time than (k_1, p_1), i.e. $k_1 \Delta t + p_1 \Delta t_p < k \Delta t + p \Delta t_p$.

In the case of delayed communication, an event results in a delayed update of the assumed values, e.g. in the worst case: $\hat{\varphi}_{k,p+\tau_{\max}}^{i,j} := \varphi_{k,p}^{j}$ and $\xi_{k,p}^{j,i} := \varphi_{k,p}^{j}$ if $\|\mathscr{M}^{i,j}(\xi_{k,p}^{j,i,-} - \varphi_{k,p}^{j})\| > \gamma^i$. This situation is shown in Figure 6.1, which gives an example for the evolution of the actual values $\varphi_{k,p}^{j}$ of subsystem $\mathcal{P}^j$, the value $\hat{\varphi}_{k,p}^{i,j}$ assumed by $\mathcal{C}^i$ for $\varphi_{k,p}^{j}$, the value $\xi_{k,p}^{j,i}$ used by $\mathcal{C}^i$ to evaluate the triggering condition, and the corresponding bounds on the difference between $\hat{\varphi}^{i,j}$ and $\varphi_{k,p}^{j}$. In the figure, multiple iterations for each time step k are shown and the message sent from $\mathcal{C}^j$ to $\mathcal{C}^i$ at $(4,2)$ arrives with a delay of three iterations. Similarly, a communication event is triggered at $(5,0)$ and the message arrives at $(5,3)$. It can be seen that the difference is outside of the bounds due to the delayed communication.

In the case of a packet loss (6.25) becomes:

$$(\hat{\varphi}_{k,p}^{i,j}; \xi_{k,p}^{j,i}) := \begin{cases} (\hat{\varphi}_{k,p}^{i,j,-}; \varphi_{k,p}^{j}) & \text{if } \|\mathscr{M}^{i,j}(\xi_{k,p}^{j,i,-} - \varphi_{k,p}^{j})\| > \gamma^i \\ (\hat{\varphi}_{k,p}^{i,j,-}; \xi_{k,p}^{j,i,-}) & \text{otherwise} \end{cases}, \tag{6.56}$$

i.e. it no longer holds that $\hat{\varphi}_{k,p}^{i,j} = \xi_{k,p}^{j,i}$. This situation is depicted in Figure 6.2, which shows the evolution of the assumed and true values for an unstable predictor $\mathscr{P}^j$. Specifically, a communication event is triggered at time $(6,0)$ and the message sent from $\mathcal{C}^j$ to $\mathcal{C}^i$ is lost and $\hat{\varphi}_{6,0}^{i,j}$ is not reset. In contrast, the value $\xi_{6,0}^{j,i}$ used by $\mathcal{C}^j$ to evaluate the triggering condition is reset because $\mathcal{C}^j$ is not aware that the message was lost. The same situation occurs at $(6,3)$, and at this point $\varphi_{k,p}^{j}$ and $\xi_{k,p}^{j,i}$ reach an equilibrium. However, because $\hat{\varphi}_{k,p}^{i,j}$ was not reset and the predictor is unstable, the difference between $\hat{\varphi}_{k,p}^{i,j}$ and $\xi_{k,p}^{j,i}$ grows with each time step and results in an unbounded error $\delta\hat{\varphi}_{k,p}^{i,j}$.

Thus, in order to bound the difference $\hat{\varphi}_{k,p}^{i,j}$ it is assumed that the dynamical system $\hat{\varphi}_{k+1,0}^{i,j} = \mathscr{P}^j \hat{\varphi}_{k,0}^{i,j}$ is marginally stable. It directly follows that there exist

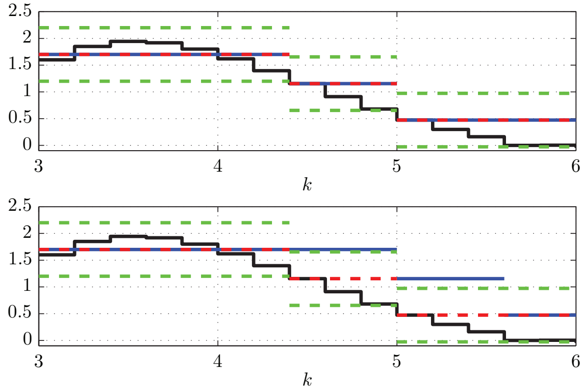

Figure 6.1.: Evolution of $\varphi^{j}_{k,p}$ (black), $\hat{\varphi}^{i,j}_{k,p}$ (blue), $\xi^{j,i}_{k,p}$ (red, dashed) and corresponding thresholds (magenta, dashed) for $\tau_{\max} = 0$ and $\tau_l = 0$ (upper subplot) and $\tau_{\max} = 3$ and τ_l (lower subplot).

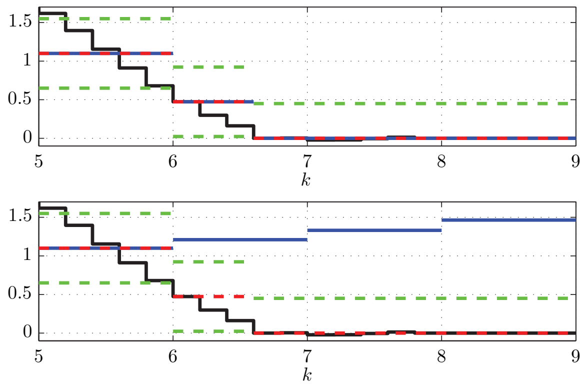

Figure 6.2.: Evolution of $\varphi^{j}_{k,p}$ (black), $\hat{\varphi}^{i,j}_{k,p}$ (blue), $\xi^{j,i}_{k,p}$ (red, dashed) and corresponding thresholds (magenta, dashed) for $\tau_{\max} = 0$ and $\tau_l = 0$ (upper subplot) and $\tau_{\max} = 0$ and $\tau_l = 2$ (lower subplot).

scalars $c^{i}_{\mathscr{P}} \geq 1$ such that the following holds for all $l \in \mathbb{N}_0$, all $i \in \mathcal{N}$, and all $\delta\hat{\varphi}^{i,j}_{k,p}$:

$$c^{i}_{\mathscr{P}} \|\mathscr{M}^{i,j} \delta\hat{\varphi}^{i,j}_{k,p}\| \geq \|\mathscr{M}^{i,j} (\mathscr{P}^{j})^{l} \delta\hat{\varphi}^{i,j}_{k,p}\|, \quad \forall j \in \mathcal{N}^{i}. \tag{6.57}$$

Furthermore, because $\mathbb{U}$ and $\mathcal{X}$ are assumed to be compact, there exist constants

$c^i_\varphi \in \mathbb{R}_{\geq 0}$ such that for all $p > 0$, all $i \in \mathcal{N}$, and all $\varphi^j_{k,p}$ it holds that

$$c^i_\varphi \geq \|\mathscr{M}^{i,j}(\varphi^j_{k,p} - \varphi^j_{k,p+1})\|, \quad \forall j \in \mathcal{N}^i, \tag{6.58}$$
$$c^i_\varphi \geq \|\mathscr{M}^{i,j}(\varphi^j_{k,\bar{p}_k-1} - \varphi^j_{k+1,0})\|, \quad \forall j \in \mathcal{N}^i, \tag{6.59}$$

where the first line bounds the increase in the triggering function from one iteration to the next, and the second inequality bounds the increase of the triggering function due to one iteration and the reinitialization for the next time step. To obtain tight bounds on γ_d the smallest $c^i_{\mathscr{P}}$ and c^i_φ should be used for which the conditions (6.57) to (6.59) hold.

The following result is concerned with the case of delayed communication $\tau_{\max} \in \mathbb{N}_0$ without packet loss (i.e. $\tau_l = 0$). It follows from Assumption 6.2 that $\tau_l = 0$ implies that $\tau_{\max}\Delta t_p \leq \Delta t - p_{\max}\Delta t_p$.

Theorem 6.8. *If $\tau_{\max} \in \mathbb{N}_0$, $\tau_l = 0$ (i.e. $\tau_{\max}\Delta t_p \leq \Delta t - p_{\max}\Delta t_p$), $\|\mathscr{M}^{i,j}\delta\hat{\varphi}^{i,j}_{0,0}\| \leq \gamma^i$, and $\xi^{j,i}_{0,0} = \hat{\varphi}^{i,j}_{0,0}$ for all $(i,j) \in \mathcal{N} \times (\mathcal{N} \backslash i)$ it holds for all (k,p) that $\|\mathscr{M}^{i,j}\delta\hat{\varphi}^{i,j}_{k,p}\| \leq \gamma^i + \tau_{\max}c^i_\varphi$.*

Proof. The conditions $\|\mathscr{M}^{i,j}\delta\hat{\varphi}^{i,j}_{0,0}\| \leq \gamma^i$ and $\xi^{j,i}_{0,0} = \hat{\varphi}^{i,j}_{0,0}$ ensure that $\|\mathscr{M}^{i,j}\delta\hat{\varphi}^{i,j}_{k,p}\| \leq \gamma^i$ and $\xi^{j,i}_{k,p} = \hat{\varphi}^{i,j}_{k,p}$ for $(k,p) < (k^{i,j}_1, p^{i,j}_1)$. Next, using (6.58), and the fact that $\hat{\varphi}^{i,j}_{k,p}$ and $\xi^{j,i}_{k,p}$ are updated at most $\tau_{\max}$ iterations apart it follows that

$$\|\mathscr{M}^{i,j}(\hat{\varphi}^{i,j}_{k,p} - \xi^{j,i}_{k,p})\| \leq \tau_{\max}c^i_\varphi, \tag{6.60}$$

for all (k,p). Furthermore, because $\xi^{j,i}_{k,p}$ is updated immediately when a communication event is triggered it holds that

$$\|\mathscr{M}^{i,j}(\xi^{j,i}_{k,p} - \varphi^j_{k,p})\| \leq \gamma^i, \tag{6.61}$$

for all (k,p). This results in

$$\|\mathscr{M}^{i,j}\delta\hat{\varphi}^{i,j}_{k,p}\| = \|\mathscr{M}^{i,j}(\hat{\varphi}^{i,j}_{k,p} - \xi^{j,i}_{k,p} + \xi^{j,i}_{k,p} - \varphi^j_{k,p})\|, \tag{6.62}$$
$$\leq \|\mathscr{M}^{i,j}(\hat{\varphi}^{i,j}_{k,p} - \xi^{j,i}_{k,p})\| + \|\mathscr{M}^{i,j}(\xi^{j,i}_{k,p} - \varphi^j_{k,p})\|, \tag{6.63}$$
$$\leq \gamma^i + \tau_{\max}c^i_\varphi, \tag{6.64}$$

for all (k,p) and the theorem follows. □

This result shows that this algorithm is rather ill-suited to deal with communication delays if multiple iterations are performed. In particular, the bound $\bar{V}_e$ grows rapidly with $\tau_{\max}$ if it is computed based on $\|\mathscr{M}^{i,j}\delta\hat{\varphi}^{i,j}_{k,p}\| \leq \gamma^i + \tau_{\max}c^i_\varphi$. This could be expected, because in contrast to Algorithm 3.1 communication delays are not explicitly considered in the local optimization. The next result focuses on the case $\tau_{\max} = 0$ and $\tau_l \in \mathbb{R}_{>0}$, i.e. only considers packet loss.

Theorem 6.9. *If* $\tau_{\max} = 0$, $\tau_l \in \mathbb{R}_{>0}$, $\|\mathscr{M}^{i,j}\delta\hat{\varphi}^{i,j}_{0,0}\| \leq \gamma^i$, *and* $\xi^{j,i}_{0,0} = \hat{\varphi}^{i,j}_{0,0}$ *for all* $(i,j) \in \mathcal{N} \times (\mathcal{N} \setminus i)$ *it holds that*

$$\|\mathscr{M}^{i,j}\delta\hat{\varphi}^{i,j}_{k,p}\| \leq \textstyle\sum_{l=1}^{\tau_l}(c^i_{\mathscr{P}})^l(\gamma^i + c^i_{\varphi}) + \gamma^i.$$

Proof. Let $(k^{i,j}_e, p^{i,j}_e)$ denote the time of the last update which was not lost and let $(k^{i,j}_{e+\tau_l+1},\ p^{i,j}_{e+\tau_l+1})$ denote the time of the next update guaranteed to arrive. For $(k^{i,j}_e, p^{i,j}_e) \leq (k,p) < (k^{i,j}_{e+1}, p^{i,j}_{e+1})$ it holds that $\|\mathscr{M}^{i,j}\delta\hat{\varphi}^{i,j}_{k,p}\| \leq \gamma^i$ because no event was triggered in this time interval. For $(k^{i,j}_{e+1}, p^{i,j}_{e+1}) \leq (k,p) < (k^{i,j}_{e+2}, p^{i,j}_{e+2})$ it holds that

$$\begin{aligned}
\|\mathscr{M}^{i,j}\delta\hat{\varphi}^{i,j}_{k,p}\| &= \|\mathscr{M}^{i,j}(\hat{\varphi}^{i,j}_{k,p} - \xi^{j,i}_{k,p} + \xi^{j,i}_{k,p} - \varphi^{j}_{k,p})\|, && (6.65)\\
&\leq \|\mathscr{M}^{i,j}(\hat{\varphi}^{i,j}_{k,p} - \xi^{j,i}_{k,p})\| + \|\mathscr{M}^{i,j}(\xi^{j,i}_{k,p} - \varphi^{j}_{k,p})\|, && (6.66)\\
&\leq \|\mathscr{M}^{i,j}(\hat{\varphi}^{i,j}_{k,p} - \xi^{j,i}_{k,p})\| + \gamma^i. && (6.67)
\end{aligned}$$

Next, let $s^{i,j}_e = k^{i,j}_{e+1} - k^{i,j}_e$ denote the number of time steps between consecutive events. For $(k^{i,j}_{e+1}, p^{i,j}_{e+1}) < (k,p) < (k^{i,j}_{e+2}, p^{i,j}_{e+2})$ considering the forward prediction $\hat{\varphi}^{i,j}_{k+1,0} = \mathscr{P}^j\hat{\varphi}^{i,j}_{k,\bar{p}_k}$ results in

$$\begin{aligned}
\|\mathscr{M}^{i,j}\delta\hat{\varphi}^{i,j}_{k,p}\| &\leq \|\mathscr{M}^{i,j}(\mathscr{P}^j)^{s^{i,j}_{e+1}}(\hat{\varphi}^{i,j}_{k^{i,j}_{e+1},p^{i,j}_{e+1}} - \xi^{j,i}_{k^{i,j}_{e+1},p^{i,j}_{e+1}})\| + \gamma^i, && (6.68)\\
&\leq c^i_{\mathscr{P}}\|\mathscr{M}^{i,j}(\hat{\varphi}^{i,j}_{k^{i,j}_{e+1},p^{i,j}_{e+1}} - \xi^{j,i}_{k^{i,j}_{e+1},p^{i,j}_{e+1}})\| + \gamma^i, && (6.69)\\
&\leq c^i_{\mathscr{P}}(\gamma^i + c^i_{\varphi}) + \gamma^i. && (6.70)
\end{aligned}$$

The last line follows from $\|\mathscr{M}^{i,j}(\hat{\varphi}^{i,j}_{k,p} - \varphi^{j}_{k,p})\| \leq \gamma^i$ for $(k,p) < (k^{i,j}_{e+1}, p^{i,j}_{e+1})$, inequality (6.58), which implies that $\|\mathscr{M}^{i,j}(\hat{\varphi}^{i,j}_{k^{i,j}_{e+1},p^{i,j}_{e+1}} - \varphi^{j}_{k^{i,j}_{e+1},p^{i,j}_{e+1}})\| \leq \gamma^i + c^i_{\varphi}$, and $\xi^{j,i}_{k^{i,j}_{e+1},p^{i,j}_{e+1}} = \varphi^{j}_{k^{i,j}_{e+1},p^{i,j}_{e+1}}$. By the same arguments it holds for $(k^{i,j}_{e+2}, p^{i,j}_{e+2}) \leq (k,p) < (k^{i,j}_{e+3}, p^{i,j}_{e+3})$ that

$$\begin{aligned}
\|\mathscr{M}^{i,j}\delta\hat{\varphi}^{i,j}_{k,p}\| &\leq \|\mathscr{M}^{i,j}(\mathscr{P}^j)^{s^{i,j}_{e+2}}(\hat{\varphi}^{i,j}_{k^{i,j}_{e+2},p^{i,j}_{e+2}} - \xi^{j,i}_{k^{i,j}_{e+2},p^{i,j}_{e+2}})\| + \gamma^i, && (6.71)\\
&\leq c^i_{\mathscr{P}}\|\mathscr{M}^{i,j}(\hat{\varphi}^{i,j}_{k^{i,j}_{e+2},p^{i,j}_{e+2}} - \xi^{j,i}_{k^{i,j}_{e+2},p^{i,j}_{e+2}})\| + \gamma^i, && (6.72)\\
&\leq c^i_{\mathscr{P}}(c^i_{\mathscr{P}}(\gamma^i + c^i_{\varphi}) + \gamma^i + c^i_{\varphi}) + \gamma^i, && (6.73)\\
&\leq (c^i_{\mathscr{P}})^2(\gamma^i + c^i_{\varphi}) + c^i_{\mathscr{P}}(\gamma^i + c^i_{\varphi}) + \gamma^i. && (6.74)
\end{aligned}$$

The theorem follows by induction over the event times up to $(k^{i,j}_{e+\tau_l+1}, p^{i,j}_{e+\tau_l+1})$ and noting that $\|\mathscr{M}^{i,j}\delta\hat{\varphi}^{i,j}_{k^{i,j}_{e+\tau_l+1},p^{i,j}_{e+\tau_l+1}}\| = 0$. Furthermore, if no further event is triggered after $(k^{i,j}_{e+\tau_l}, p^{i,j}_{e+\tau_l})$ the difference remains within the bounds because

$$\|\mathscr{M}^{i,j}(\mathscr{P}^j)^{s_{e+\tau_l}}(\hat{\varphi}^{i,j}_{k^{i,j}_{e+\tau_l},p^{i,j}_{e+\tau_l}} - \xi^{j,i}_{k^{i,j}_{e+\tau_l},p^{i,j}_{e+\tau_l}})\| \leq c^i_{\mathscr{P}}\|\mathscr{M}^{i,j}(\hat{\varphi}^{i,j}_{k^{i,j}_{e+\tau_l},p^{i,j}_{e+\tau_l}} - \xi^{j,i}_{k^{i,j}_{e+\tau_l},p^{i,j}_{e+\tau_l}})\|$$

holds for all $s_{e+\tau_l} \in \mathbb{N}_0$ (cf. (6.57)). □

This result highlights that the choice of $\mathscr{P}^j$ has a strong influence on the robustness of the algorithm with respect to packet loss, even if only stable $\mathscr{P}^j$ are considered. For instance, it can be seen that $\mathscr{P}^j = I$ may result in significantly smaller bounds than $\mathscr{P}^j$ given by (6.49) or (6.53). This can be interpreted as follows. According to (6.55) it holds that

$$\|\mathscr{M}^{i,j}\delta\hat{\varphi}^{i,j}_{k+1,0}\| \leq \|\mathscr{M}^{i,j}\mathscr{P}^j\delta\hat{\varphi}^{i,j}_{k,\bar{p}_k}\| + \|\mathscr{M}^{i,j}(\mathscr{P}^j\varphi^j_{k,\bar{p}_k} - \varphi^j_{k+1,0})\|. \tag{6.75}$$

The forward predictions used in (6.48) and (6.52) were chosen to reduce the two terms on the right hand side. Choosing the predictors such that $c^i_{\mathscr{P}}$ is close to 1, the second term grows significantly faster and more communication events are triggered. At the same time, the impact of applying forward predictions to inaccurate data is diminished. Overall, the fact that the forward prediction error grows slower and communication events are triggered more frequently reduces the impact of lost packets. Thus, if $\tau_{\max} = 0$ and $\tau_l = 1$ reasonable performance may be obtained by choosing $\mathscr{P}^j$ such that $c^i_{\mathscr{P}} = 1$ and choosing a suitably small threshold γ. In the next theorem a combination of delays and packet loss is considered.

Theorem 6.10. *If $\tau_{\max} \in \mathbb{R}_{>0}$, $\tau_l \in \mathbb{R}_{>0}$, $\|\mathscr{M}^{i,j}\delta\hat{\varphi}^{i,j}_{0,0}\| \leq \gamma^i$, and $\xi^{j,i}_{0,0} = \hat{\varphi}^{i,j}_{0,0}$ for all $(i,j) \in \mathcal{N} \times (\mathcal{N} \setminus i)$ it holds that $\|\mathscr{M}^{i,j}\delta\hat{\varphi}^{i,j}_{k,p}\| \leq \gamma^i_d$, with*

$$\gamma^i_d := \textstyle\sum_{l=1}^{\tau_l}(c^i_{\mathscr{P}})^l(\gamma^i + c^i_{\varphi}) + \gamma^i + \tau_{\max}c^i_{\varphi}.$$

Proof. Let $\bar{\varphi}^{i,j}_{k,p}$ denote the values of $\hat{\varphi}^{i,j}_{k,p}$ which would be obtained in the case $\tau_{\max} = 0$, $\tau_l \in \mathbb{R}_{>0}$. Considering Theorem 6.9 it follows that

$$\|\mathscr{M}^{i,j}(\bar{\varphi}^{i,j}_{k,p} - \varphi^j_{k,p})\| \leq \textstyle\sum_{l=1}^{\tau_l}(c^i_{\mathscr{P}})^l(\gamma^i + c^i_{\varphi}) + \gamma^i, \tag{6.76}$$

for all (k,p). Furthermore, using the same arguments made in the proof of Theorem 6.8 $\|\mathscr{M}^{i,j}(\hat{\varphi}^{i,j}_{k,p} - \bar{\varphi}^{i,j}_{k,p})\| \leq \tau_{\max}c^i_{\varphi}$ holds for all (k,p). This results in

$$\begin{aligned}
\|\mathscr{M}^{i,j}\delta\hat{\varphi}^{i,j}_{k,p}\| &= \|\mathscr{M}^{i,j}(\hat{\varphi}^{i,j}_{k,p} - \bar{\varphi}^{i,j}_{k,p} + \bar{\varphi}^{i,j}_{k,p} - \varphi^j_{k,p})\|, & (6.77)\\
&\leq \|\mathscr{M}^{i,j}(\hat{\varphi}^{i,j}_{k,p} - \bar{\varphi}^{i,j}_{k,p})\| + \|\mathscr{M}^{i,j}(\bar{\varphi}^{i,j}_{k,p} - \varphi^j_{k,p})\|, & (6.78)\\
&\leq \textstyle\sum_{l=1}^{\tau_l}(c^i_{\mathscr{P}})^l(\gamma^i + c^i_{\varphi}) + \gamma^i + \tau_{\max}c^i_{\varphi}, & (6.79)
\end{aligned}$$

for all (k,p) and the theorem follows. □

The bounds $\bar{V}_e = c_e\gamma_d + \gamma_d^2$ and $\bar{V}_d = (1-\bar{\beta})^{-1}(\zeta + \bar{V}_e)$ in the case of delays or packet loss can now be computed based on an enlarged threshold $\gamma_d := \max_{i\in\mathcal{N}}\gamma^i_d \operatorname{card}(\mathcal{N}^i)$. In the case of delays it may be impossible to pick a $\gamma \in \mathbb{R}_{\geq 0}$ such that Assumption 5.3 holds, which is crucial to establish practical stability of the algorithm without terminal constraint (cf. Theorem 6.7). On the other hand, if $\tau_{\max} = 0$, $\tau_l = 1$ and suitable $\mathscr{P}^j$ and $\gamma \in \mathbb{R}_{>0}$ is chosen it may be possible to establish practical stability of the closed-loop. In other words, the iterative algorithm without terminal constraint in general requires a communication network which ensures transmission of the data without significant delays and small number of lost packets. Because of the decoupled dynamics, practical stability of the algorithm with terminal constraint can be established for a communication network with significant delays and packet loss, but the resulting suboptimality may be large.

6.6. Numerical Example

To illustrate the proposed scheme in closed loop, Algorithm 6.2 is applied to the numerical example motivated by a system of coupled water tanks introduced in the previous chapter (see Figure 5.1). Using the bounds computed in Theorem 6.3, the thresholds γ and ζ would have to be chosen in the range of 10^{-3} to guarantee the same suboptimality bounds as in Section 5.4. However, as discussed below Proposition 6.4 these bounds may be very conservative. In order to allow for a better comparison of the communication requirements of Algorithm 5.1 and Algorithm 6.2, the thresholds $\gamma = 1.5$ and $\zeta = 0.5$ were chosen based on simulations. These thresholds provide comparable closed-loop performance to the results shown for Algorithm 5.1 in Section 5.4. All other parameters are identical to those used in Section 5.4, e.g. $p_{\max} = 20$, $N = 20$.

Simulation results for the water levels and inputs are shown in Figure 6.3. Again, only the inputs of subsystem 2 and 4 are used due to the high cost for the control inputs of tank 1 and 3. It can be seen that in contrast to Algorithm 5.1 (cf. Figure 5.2) the inputs are not oscillating close to the equilibrium. This can be explained by the fact that in Algorithm 5.1 small updates to the control inputs are discarded, while they are applied but not communicated in Algorithm 6.2.

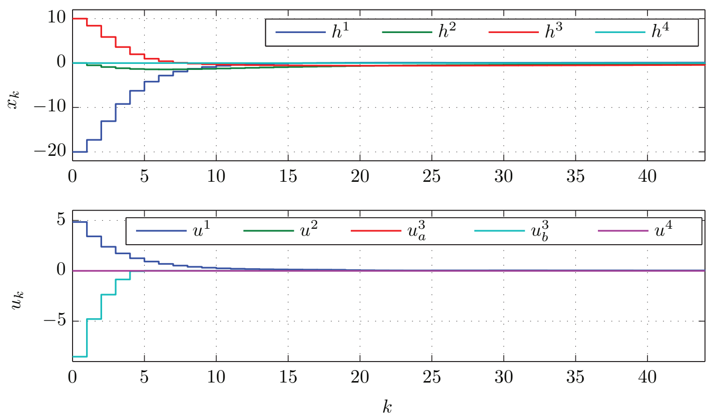

Figure 6.3.: Simulation results for the coupled water tanks using Algorithm 6.2.

The number of communication events triggered between each pair of controllers $\mathcal{C}^i$ and $\mathcal{C}^j$ is shown in Figure 6.4. In particular, $e_k^{i,j} = \sum_{p=0}^{\bar{p}_k} e_{k,p}^{i,j}$, where $\bar{p}_k$ again denotes the last iteration performed at time k, and $e_{k,p}^{i,j} = 1$ if $(i,j) \in \mathcal{T}_{k,p}$ and $e_{k,p}^{i,j} = 0$ otherwise. While a communication event in Algorithm 5.1 is global (e.g. may require $N_s - 1$ messages), a communication event in Algorithm 6.2 results in one message being sent from one controller to one other controller. Figure 6.4 does not include messages used to verify the stopping criterion. The total number of messages for information exchange between controllers is 58, compared to 345 messages (115 events) used by Algorithm 5.1. In the implementation chosen for this example, the stopping criterion (6.37) is verified as follows. Each controller checks whether (6.37) holds, and only if this condition does not hold, a message is sent to a

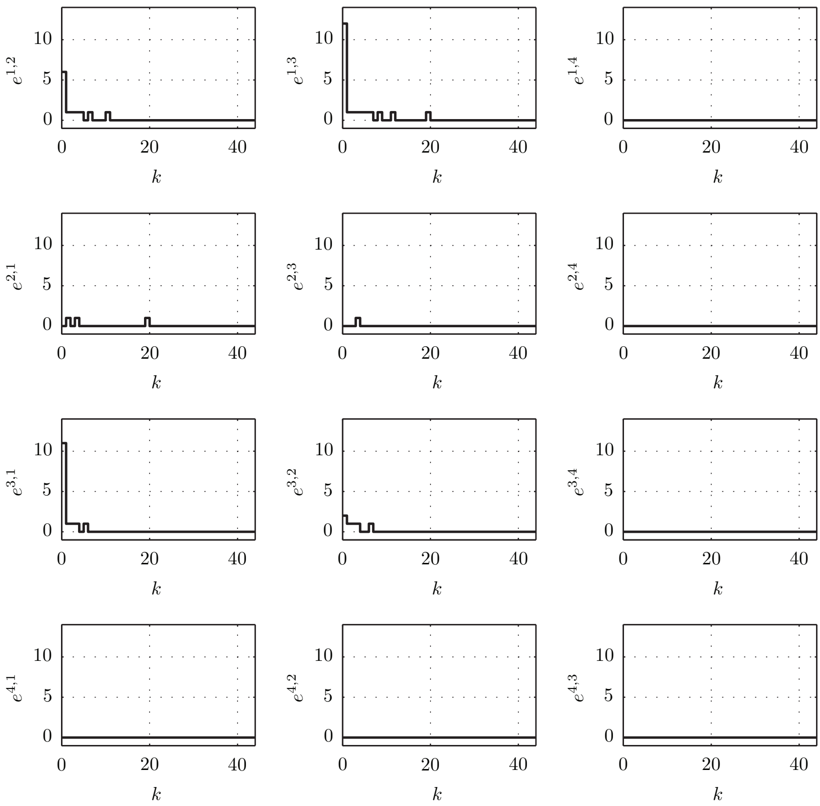

Figure 6.4.: Number of communication events $e_k^{i,j}$ per time step k.

controller which acts as coordinator. This controller subsequently sends a message to all other controllers. Note that these messages are very small compared to the 58 messages used to communicate the states and input sequences. In the chosen implementation 93 messages were required to verify the stopping criterion.

The resulting suboptimality $V_d(x_k, \mathbf{v}_{k,p})$ is shown in Figure 6.5. In comparison to Figure 5.3 it can be seen that, for the chosen thresholds, the suboptimality of Algorithm 6.2 is about three times larger than that of Algorithm 5.1 for $k = 0$, but significantly smaller for $k > 2$. Similarly, Algorithm 6.2 performs slightly more iterations at $k = 0$, but significantly less for $k > 0$. For comparison, the absolute costs are $V(x_0, \mathbf{v}_{0,0}) \approx 1100$ and $V(x_{45}, \mathbf{v}_{45,0}) \approx 1.5$).

Figure 6.6 shows the costs $V(x_k, \mathbf{v}_{k,\bar{p}_k})$ of the closed-loop resulting from Algorithm 6.2 and Algorithm 5.1 for identical initial conditions. Overall it can be seen that Algorithm 5.1 performs slightly worse than Algorithm 6.2 in this example for $k \geq 5$, while requiring both more iterations and communication than Algorithm 6.2.

With respect to the messages required to verify the stopping criterion, it should be

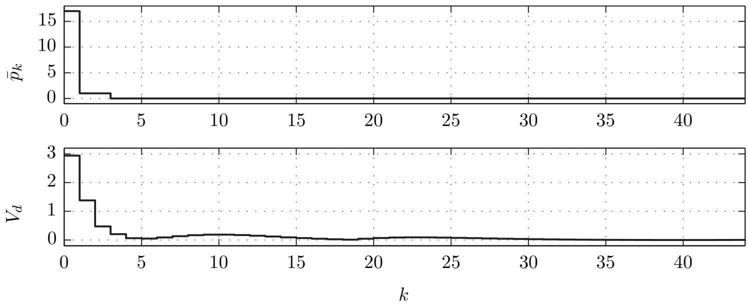

Figure 6.5.: Number of iterations and suboptimality $V_d(x_k, \mathbf{v}_{k,p})$ for Algorithm 6.2.

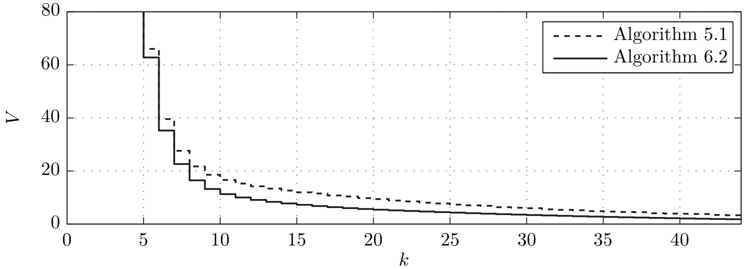

Figure 6.6.: Comparison of the costs $V(x_k, \mathbf{v}_{k,\bar{p}_k})$ resulting from Algorithm 6.2 and Algorithm 5.1 (cf. Section 5.4) for $k \geq 5$.

noted that from the point of view of computational complexity the stopping criterion may not be required. Specifically, the computational cost of iterations is very low if no new information is received in an iteration, because the optimization problem (6.8) does not need to be solved in this case. Based on this observation, Algorithm 6.2 was applied to the example at hand without stopping condition for $p_{\max} = 20$ iterations. This resulted in 60 messages and comparable performance to the results obtained by using a stopping condition, while the number of optimization problems solved was only increased from 207 to 211. Therefore, if a reasonably small $p_{\max}$ can be chosen a-priori, such that the suboptimality after $p_{\max}$ iterations is sufficiently small, using a stopping criterion may not result in less computational complexity.

6.7. Discussion

In this chapter, a distributed MPC algorithm with event-based communication based on sensitivity analysis is developed. The main idea in this chapter is to analyze how the communicated information affects the local optimization by means of sensitivity analysis. In contrast to the approach used in Chapter 5, this answers the question of when to communicate as well as the question between which controllers information needs to be exchanged. Specifically, the analysis in Chapter 5 is concerned with how strongly the local optimization influences the global cost. In contrast, the analysis in this chapter goes one step further and also considers how strongly the result of the local optimization depends on the information exchanged between interconnected controllers.

In each iteration the controllers $\mathcal{C}^i$ compute candidate input sequences by solving the same problem used in Chapter 4, but the optimization does not require exact knowledge of the states and input sequences of interconnected subsystems and controllers. To this end, each controller $\mathcal{C}^i$ assumes values for the states and inputs of interconnected subsystems. These values are synchronized by event-based communication if the difference between the assumed and the true values is above a certain threshold. Based on these triggering conditions, a distributed MPC algorithm is obtained, in which the triggering conditions not only decide when to communicate but also between which controllers. A downside of this approach is that the algorithm cannot be applied to Case 2.2, i.e. it is either applicable to subsystems with coupling only by costs, or to coupled dynamics and cost without terminal constraint. In the latter case, the algorithm cannot be terminated at any iteration, but only if a stopping condition ensuring bounded suboptimality holds.

Developing a rigorous stopping condition for the algorithm discussed in this chapter is a challenging problem because the triggering conditions for communication are no longer directly related to the suboptimality of the current iterate. Furthermore, ensuring bounded suboptimality of the algorithm requires that a stopping condition holds for all the controllers at the same time. To this end, a stopping condition is derived which can be checked in a distributed fashion, i.e. all computations are performed locally and communication is only required to ensure that the condition holds for all controllers. If a terminal constraint is used, this check can be omitted and the algorithm can be terminated at any iteration. Because the states and input sequences of other subsystems and controllers are only known approximately, only practical stability can be guaranteed even when using a terminal constraint. In contrast, Algorithm 5.1 with event-based communication guarantees asymptotic stability of the closed-loop if a terminal constraint is used.

The threshold $\gamma \in \mathbb{R}_{>0}$ in the sensitivity based algorithm allows for a trade-off between how closely, in terms of the costs, the sensitivity based algorithm tracks the input sequences generated by the algorithm with full communication discussed in Chapter 4. In other words, a small threshold $\gamma \in \mathbb{R}_{>0}$ ensures that the resulting input sequence closely resembles that of the algorithm from Chapter 4, while a larger threshold will result in less communication and larger suboptimality. Therefore, the threshold γ again allows for a trade-off between the load on the communication network and closed-loop performance.

Finally, the framework presented in this chapter also allows quantifying the impact of packet loss and communication delays. In particular, it is shown that both delays and packet loss can be modeled as enlargement of the threshold γ. These results indicate that the cooperative distributed MPC algorithm is not well suited to deal with communication delays and, in many cases, should be modified to better cope with packet loss. This could have been expected because, in contrast to the robust Algorithm presented in Chapter 3, delays are not explicitly considered in the local optimization. With respect to packet loss the lack of robustness is also not surprising because the main aim of the sensitivity based approach to distributed MPC with event-based communication is to reduce the load on the communication network by only communicating information which is absolutely required. Therefore, in comparison to algorithms which employ communication more frequently, the loss of one or multiple messages will have a larger impact. The theoretical results given in this chapter for the case of packet loss support this interpretation, i.e. they show that by modifying parameters of the algorithm such that communication events are triggered more frequently, the impact of packet loss on the closed-loop performance can be diminished.

Similarly to Algorithm 5.1, the sensitivity based distributed MPC is well suited for closed-loop application if the communication network does not induce uncertainties. While it cannot be applied to Case 2.2, the sensitivity based algorithm allows for a further reduction of the load on the communication network. In particular, in Algorithm 5.1 a communication event for a controller $\mathcal{C}^i$ always requires this controller to send information to all other controllers $\mathcal{C}^j$, $j \in \mathcal{N} \setminus i$, even if the interconnection graph $\mathscr{G}$ is sparse. In contrast, the triggering conditions based on sensitivity analysis typically only result in communication if the interconnections between two controllers are significant and therefore result in a sparse communication graph $\mathscr{C}_{k,p}$ if the interconnections between subsystems and controllers are sparse.

7. Event-Based Communication for DMPC of Piecewise Affine Systems

In this chapter, cooperative distributed MPC for decoupled discrete-time hybrid systems with piecewise affine dynamics, which are interconnected by costs and possibly non-convex constraints, is investigated. Centralized MPC of piecewise affine systems of medium to large scale is often not feasible due to the computational complexity of the resulting optimization problem. Because of this, distributed control of interconnected piecewise affine systems is of great interest, but few results are available in the literature which are applicable to piecewise affine systems. In [14] an iterative distributed MPC algorithm for hybrid systems is considered, but only within the scope of linear systems with discrete inputs. In principle, the results presented in [83] and [50] for non-linear dynamics may be applied to piecewise affine systems, but the algorithms are sequential and the assumptions made in [50] are very hard to verify. In any case, because of the relatively long computation times of local problems involving piecewise affine dynamics a sequential approach may be problematic.

The algorithm presented in Chapter 3 employs parallel optimization, but because the local controllers do not truly cooperate, interactions are partially modeled as disturbance and a robust local controller is required. However, the literature on robust MPC of piecewise affine systems is rather sparse and the available techniques appear to be either only applicable to systems of very low dimension [60] or with only very small disturbances, i.e. weak interactions [67].

Finally, the cooperative distributed MPC algorithm presented in the previous chapters is not directly applicable to distributed MPC of piecewise affine systems because it requires convexity of the cost function $V(x_k, \mathbf{u}_k)$ (i.e. with the dynamics substituted into the costs). In [102] an extension of the algorithm discussed in Chapter 4 to nonlinear systems is proposed, which recomputes the weights $w^i_{k,p}$ online in order to ensure that the cost is decreasing in each iteration even though the costs $V(x_k, \mathbf{u}_k)$ are not convex. However, it is assumed that $V(x_k, \mathbf{u}_k)$ is twice continuously differentiable and full communication between all controllers is required. Because the costs $V(x_k, \mathbf{u}_k)$ are, in general, neither convex nor twice continuously differentiable for piecewise affine dynamics, these results are not applicable.

The results presented in this chapter have been previously published in part in [43] and use parallel local optimization, event-based sequential communication, and compensation of small communication delays. The event-based approach also resolves the issue of access to the communication network. Specifically, only subsystems with a cost decrease above a certain threshold participate in a round of communication, and only the subsystem which offers the largest cost decrease (i.e. largest increase in global control performance) is granted access to the network.

7.1. Distributed System Model

In this chapter, distributed MPC of N_s dynamically decoupled hybrid subsystems $\mathcal{P}^i$ with discrete-time affine dynamics defined on a polyhedral partition of the continuous state space is considered. The dynamics are given by

$$x^i_{k+1} = A^i_{\mathfrak{p}^i} x^i_k + B^i_{\mathfrak{p}^i} u^i_k + g^i_{\mathfrak{p}^i}, \quad \text{if } x^i_k \in \mathbb{X}^i_{\mathfrak{p}^i} \tag{7.1}$$

As discussed in Section 2.1, the affine dynamics parametrized by $A^i_{\mathfrak{p}^i}$, $B^i_{\mathfrak{p}^i}$, $g^i_{\mathfrak{p}^i}$ is valid in the region with index $\mathfrak{p}^i \in \{1, \dots, N^i_{\mathfrak{p}}\}$ resulting in decoupled dynamics. In order to ensure that the solution of (7.1) is unique it is assumed that the polyhedral regions do not overlap. In order to define adjacent regions without non-overlapping boundaries and no gap between them, the regions $\mathbb{X}^i_{\mathfrak{p}^i}$ are often defined as the intersection of a finite number of open and closed half-spaces in the literature on piecewise affine systems.

However, when formulating optimal control problems based on the dynamics (7.1) this approach results in both theoretical and numerical issues. From a theoretical point of view using strict inequalities (i.e. open half-spaces) results in an optimal control problem which may not have a minimum but only an infimum (cf. Section 2.4). Furthermore, numerical solvers are subject to a numerical tolerance $\epsilon_n \in \mathbb{R}_{>0}$ and the underlying algorithms (e.g. interior point methods) often produce solutions which are not strictly feasible. For these reasons, most numerical solvers either do not support strict inequalities, or internally use a conversion to non-strict inequalities similar to the approach used in the subsequent parts of this chapter. Overall it can be seen that these issues are partially inherent to piecewise affine systems and partially caused by the fact that optimal control problems for piecewise affine systems can typically only be solved numerically.

Due to these effects the regions $\mathbb{X}^i_{\mathfrak{p}^i}$ are defined as closed sets reduced in size by a tolerance $\epsilon > \epsilon_n$:

$$\mathbb{X}^i_{\mathfrak{p}^i} = \{x^i_k \in \mathbb{R}^{n^i} | C^i_{\mathbb{X}_{\mathfrak{p}^i}} x^i_k \leq b^i_{\mathbb{X}_{\mathfrak{p}^i}} - \epsilon\}, \tag{7.2}$$

where $C^i_{\mathbb{X}_{\mathfrak{p}^i}} \in \mathbb{R}^{h^i_{\mathbb{X}_{\mathfrak{p}^i}} \times n}$, and $b^i_{\mathbb{X}_{\mathfrak{p}^i}} \in \mathbb{R}^{h^i_{\mathbb{X}_{\mathfrak{p}^i}}}$.

The subsystem may be interconnected by costs and common state constraints, and the input constraints are assumed to be decoupled. In order to define the state constraints, let $\mathbb{X}^c = \{x_k \in \mathbb{R}^n | C_{\mathbb{X}^c} x \leq b_{\mathbb{X}^c}\}$ denote coupled polyhedral state constraints with $C_{\mathbb{X}^c} \in \mathbb{R}^{h_{\mathbb{X}^c} \times n}$, $b_{\mathbb{X}^c} \in \mathbb{R}^{h_{\mathbb{X}^c}}$. Furthermore, let $\mathbb{X}_r^e := \{x_k \in \mathbb{R}^n | C_{\mathbb{X}_r^e} x_k \leq b_{\mathbb{X}_r^e}\}$, with $C_{\mathbb{X}_r^e} \in \mathbb{R}^{h_{\mathbb{X}_r^e} \times n}$, $b_{\mathbb{X}_r^e} \in \mathbb{R}^{h_{\mathbb{X}_r^e}}$ denote N_e regions which are excluded from the feasible set. Then, the overall state constraint is given by

$$\mathbb{X} := \left\{ x_k \in \mathbb{R}^n \;\middle|\; \begin{array}{l} x_k^i \in \cup_{\mathfrak{p}^i=1}^{N_{\mathfrak{p}}^i} \mathbb{X}_{\mathfrak{p}^i}^i, \forall i \in \mathcal{N}, \\ x_k \in \mathbb{X}^c, \\ x_k \notin \mathbb{X}_r^e, \forall r \in \{1, \ldots, N_e\} \end{array} \right\}. \tag{7.3}$$

This formulation allows to approximate arbitrary non-convex constraints, and the constraint $x_k \notin \mathbb{X}_r^e$ for all $r \in \{1, \ldots, N_e\}$ can be formulated by mixed-integer linear constraints (see below).

The following assumptions are made in order to ensure that the solutions of the difference equations (7.1) are unique, and that the dynamics are continuous and stabilizable in a neighborhood of the control goal.

Assumption 7.1. *It is assumed that for all* $i \in \mathcal{N}$ *:*

1. $(\mathbb{X}_{j_1}^i \oplus \mathcal{B}_\epsilon^{n^i}(0)) \cap (\mathbb{X}_{j_2}^i \oplus \mathcal{B}_\epsilon^{n^i}(0)) = \emptyset$ *for all* $j_1 \neq j_2$, *with* $j_1, j_2 \in \{1, \ldots, N_{\mathfrak{p}^i}^i\}$,
2. $\mathcal{B}_\epsilon^{m^i}(0) \subseteq \mathbb{U}^i$, $\mathcal{B}_\epsilon^n(0) \subseteq \mathbb{X}$, *and* $0 \in int(\mathbb{X}_1^i)$,
3. *the pair* (A_1^i, B_1^i) *is stabilizable and* $g_1^i = 0$.

Assumption 7.1.1 ensures that the solutions of the difference equations (7.1) are unique. Furthermore, Assumption 7.1.2 and Assumption 7.1.3 ensure that the dynamics are continuous and stabilizable in a neighborhood of the control goal. These properties will be used to establish stability of the distributed MPC algorithm.

The following assumptions are made in order to establish recursive feasibility and stability of the distributed MPC algorithm:

Assumption 7.2. *It is assumed that there exists a terminal control law* $u_k = K_1 x_k$, *with* $K_1 = blkdiag(K_1^1, \ldots, K_1^{N_s})$, *with* $K_1^i \in \mathbb{R}^{m^i \times n^i}$, *and a decoupled, compact, terminal set* $\mathbb{T} = \mathbb{T}^1 \times \ldots \times \mathbb{T}^{N_s}$*:*

$$\mathbb{T} \subseteq (\mathbb{X}_1^1 \times \ldots \times \mathbb{X}_1^{N_s}) \cap (\mathbb{X}^c \setminus \cup_{r=1}^{N_e} \mathbb{X}_r^e), \tag{7.4}$$

with $0 \in int(\mathbb{T})$, *and* $P = blkdiag(P^1, \ldots, P^{N_s})$, *such that the following holds for all* $x_k \in \mathbb{T}$*:*

(i) $\|x_k\|_P^2 - \|(A_1 + B_1 K_1) x_k\|_P^2 \geq \|x_k\|_Q^2 + \|K_1 x_k\|_R^2$,

(ii) $(A_1 + B_1 K_1) x_k \in \mathbb{T}$, $K_1 x_k \in \mathbb{U}$.

It can be seen that the generalization of these assumptions to the case $\Sigma \neq \{0\}$ may become quite complicated and hard to verify. Therefore, only the case $\Sigma = \{0\}$ is considered here. A terminal control law and constraint satisfying Assumption 7.2

can always be constructed (cf. Appendix A.2 with $(A^i, B^i) = (A^i_1, B^i_1)$ and $\mathbb{X}_1$ instead of $\mathbb{X}$). The centralized MPC Problem is given by

$$
\begin{aligned}
&(\mathbf{x}^*_k, \mathbf{u}^*_k) = \arg\min\nolimits_{\mathbf{x}_k,\mathbf{u}_k} \mathbf{V}(\mathbf{x}_k, \mathbf{u}_k) && (7.5)\\
&\text{s.t.}\\
&x^i_{k+1+l|k} = A^i_{\mathfrak{p}^i} x^i_{k+l|k} + B^i_{\mathfrak{p}^i} u^i_{k+l|k} + g^i_{\mathfrak{p}^i},\ \text{if}\, x^i_{k+l|k} \in \mathbb{X}^i_{\mathfrak{p}^i},\ \forall i \in \mathcal{N}, \forall l \in \{0,\dots,N-1\},\\
&\quad u_{k+l|k} \in \mathbb{U}, && \forall l \in \{0,\dots,N-1\},\\
&\quad x_{k+l|k} \in \mathbb{X}^c, && \forall l \in \{0,\dots,N-1\},\\
&\quad x_{k+l|k} \notin \mathbb{X}^e_r, && \forall r \in \{1,\dots,N_e\}, \forall l \in \{0,\dots,N-1\},\\
&\quad x_{k+N|k} \in \mathbb{T},\ x_{k|k} = x_k
\end{aligned}
$$

The dependency of the dynamics on the condition $x^i_k \in \mathbb{X}^i_{\mathfrak{p}^i}$ can be formulated as linear mixed-integer constraint by introducing $N \sum_{i\in\mathcal{N}}(N^i_{\mathfrak{p}} - 1)$ binary variables (cf. [5]). The non-convex constraint $x_k \notin \mathbb{X}^e_r$ holds if, and only if, one component of the componentwise inequality $C_{\mathbb{X}^e_r} x_k > b_{\mathbb{X}^e_r}$ holds for each $r \in \{1,\dots,N_e\}$. This condition can again be formulated using $\sum_{r=1}^{N_e} h_{\mathbb{X}^e_r}$ integer variables (see e.g. [114], [96]). These reformulations may also be (partially) automated by using high level modeling tools such as YALMIP [78]. While the resulting MIQP can be solved to optimality (cf. Section 2.4), the computational complexity is high due to a large number of binary variables arising from the combinatorial nature of the dynamics (7.1). Thus, to allow for fast local computations in the distributed algorithm it is important to reduce the number of binary variables by decomposing (7.5).

In the following, $\tau_{\max} \ll \Delta t$ denotes an upper bound on the communication delays which are compensated by the distributed MPC algorithm. This assumption is based on the observation that even for piecewise affine systems of low dimension the computation times for MPC are often much larger than typical communication delays. Obviously, MPC can only be applied to piecewise affine systems if the desired closed-loop performance can be achieved by sampling times Δt larger than the computation time. In this case, it appears reasonable to employ parallel computation and assume that the local computation time and communication delay $\tau_{\max}$ is smaller than one sampling period.

7.2. Distributed Algorithm

Within the distributed MPC algorithm, the state and input vector of the local and neighboring subsystems is optimized by each controller $\mathcal{C}^i$. To this end, let $\mathcal{N}^i_0 := \mathcal{N}^i \cup \{i\}$ denote an index set containing the indices of the subsystem $\mathcal{P}^i$ and of all subsystems with which $\mathcal{P}^i$ is directly interconnected. In the distributed algorithm, each controller $\mathcal{C}^i$ optimizes over its own inputs and those of directly interconnected subsystems. Therefore, the resulting local optimization problem also depends on the state and input sequences of the subsystems $\mathcal{P}^s$, $s \in \mathcal{N}^i_1 \setminus \mathcal{N}^i_0$ for $\mathcal{N}^i_1 := \cup_{j\in\mathcal{N}^i_0} \mathcal{N}^j_0$.

Each controller optimizes the input and state trajectory by solving the following MIQP in each time-step k and iteration $p = 0$:

$$
\begin{aligned}
&(\hat{\mathbf{x}}_k^{i*}; \hat{\mathbf{u}}_k^{i*}) = \arg\min_{\hat{\mathbf{x}}_k, \hat{\mathbf{u}}_k} \mathbf{V}(\hat{\mathbf{x}}_k, \hat{\mathbf{u}}_k) && (7.6)\\
&\text{s.t.}\\
&\hat{x}^{i,j}_{k+1+l|k} = A^j_{\mathfrak{p}^j}\hat{x}^{i,j}_{k+l|k} + B^j_{\mathfrak{p}^j}\hat{u}^{i,j}_{k+l|k} + g^j_{\mathfrak{p}^j}, \text{ if } \hat{x}^{i,j}_{k+l|k} \in \mathbb{X}^j_{\mathfrak{p}^j}, && \forall j \in \mathcal{N}^i_0, \forall l \in \{0,\dots,N-1\},\\
&\hat{u}_{k+l|k} \in \mathbb{U},\ \hat{x}_{k+l|k} \in \mathbb{X}^c, && \forall l \in \{0,\dots,N-1\},\\
&\hat{x}_{k+l|k} \notin \mathbb{X}^e_g, && \forall r \in \{1,\dots,N_e\}, \forall l \in \{0,\dots,N-1\},\\
&\hat{x}^{i,s}_{k+l|k} = x^s_{k+l|k,0},\ \hat{u}^{i,s}_{k+l|k} = u^s_{k+l|k,0}, && \forall s \in \mathcal{N}^i_1 \setminus \mathcal{N}^i_0, \forall l \in \{0,\dots,N-1\},\\
&\hat{x}^{i,q}_{k+l|k} = 0, \hat{u}^{i,q}_{k+l|k} = 0, && \forall q \notin \mathcal{N}^i_1, \forall l \in \{0,\dots,N-1\},\\
&\hat{x}_{k+N|k} \in \mathbb{T},\ \hat{u}_{k|k} = u_{k|k,0},
\end{aligned}
$$

where $\hat{x}^{i,j}_{k+l|k}$ and $\hat{u}^{i,j}_{k+l|k}$ denotes a state and input sequence planned by $\mathcal{C}^i$ for a subsystem $\mathcal{P}^j$ and $\hat{\mathbf{x}}_k^{i*}$, and $\hat{\mathbf{u}}_k^{i*}$ contain the planned states and inputs for all subsystems and over the whole prediction horizon. In other words, each controller optimizes a subset of the overall state and input sequences and the state and input sequences not optimized by $\mathcal{C}^i$ are fixed to constant values from the initialization $\mathbf{x}_{k,0}$ and $\mathbf{u}_{k,0}$. Because only the piecewise affine dynamics of subsystem $\mathcal{P}^i \in \mathcal{N}^j_0$ have to be considered, the number of binary variables used to encode the dependency of the dynamics on the polyhedral partition of the state space is reduced to $N\sum_{j\in\mathcal{N}^j_0}(N^j_{\mathfrak{p}} - 1)$. Furthermore, the number of integer variables required to model the constraint $\hat{x}_{k+l|k} \notin \mathbb{X}^e_g$, for all $r \in \{1,\dots,N_e\}$ is reduced because only the constraints involving the subsystems $\mathcal{P}^i$, $i \in \mathcal{N}^i_1$ need to be considered. This results in a large reduction of computational complexity if the interconnections are sparse. Furthermore, the optimizer in iteration p given by $(\bar{\mathbf{x}}^i_{k,p}, \bar{\mathbf{u}}^i_{k,p})$ does not depend on states and inputs of $\mathcal{P}^q$, $q \notin \mathcal{N}^i_1$. Thus, these states and inputs are set to 0 (which is feasible by Assumption 7.1.1).

The main idea is to solve the local problems in parallel once per time step and to compute a new global solution by iteratively combining the local solutions. To this end, the local candidate solution $(\bar{\mathbf{x}}^i_k, \bar{\mathbf{u}}^i_k)$ in each iteration is given by

$$
\bar{x}^{i,j}_{k+l|k,p} := \begin{cases} \hat{x}^{i*,j}_{k+l|k} & \text{if } j \in \mathcal{N}^i_0, \\ x^j_{k+l|k,p} & \text{if } j \in \mathcal{N}^i_1 \setminus \mathcal{N}^i_0, \\ 0 & \text{otherwise,} \end{cases} \qquad \forall l \in \{0,\dots,N\},
$$

$$
\bar{u}^{i,j}_{k+l|k,p} := \begin{cases} \hat{u}^{i*,j}_{k+l|k} & \text{if } j \in \mathcal{N}^i_0, \\ u^j_{k+l|k,p} & \text{if } j \in \mathcal{N}^i_1 \setminus \mathcal{N}^i_0, \\ 0 & \text{otherwise,} \end{cases} \qquad \forall l \in \{0,\dots,N-1\}.
$$

In other words, the locally optimized values $\hat{x}^{i*,j}_{k+l|k}$, $\hat{u}^{i*,j}_{k+l|k}$ are combined with the current iterates for sequences which have not been optimized locally. This candidate solution is compared to the corresponding sequence $(\check{\mathbf{x}}_{k,p}, \check{\mathbf{u}}_{k,p})$ without the locally optimized values, given by

$$\check{x}^j_{k+l|k,p} := \begin{cases} x^j_{k+l|k,p} & \text{if } j \in \mathcal{N}^i_1, \\ 0 & \text{otherwise,} \end{cases} \qquad \forall l \in \{0, \dots, N\},$$

$$\check{u}^j_{k+l|k,p} := \begin{cases} u^j_{k+l|k,p} & \text{if } j \in \mathcal{N}^i_1, \\ 0 & \text{otherwise,} \end{cases} \qquad \forall l \in \{0, \dots, N-1\}.$$

Next, these two solutions are compared as follows:

$$\Delta\mathbf{V}^i_{k,p} := \begin{cases} \mathbf{V}(\check{\mathbf{x}}^i_{k,p}, \check{\mathbf{u}}^i_{k,p}) - \mathbf{V}(\bar{\mathbf{x}}^i_{k,p}, \bar{\mathbf{u}}^i_{k,p}) & \text{if } (\bar{\mathbf{x}}^i_{k,p}; \bar{\mathbf{u}}^i_{k,p}) \in \Upsilon, \\ -\infty & \text{otherwise,} \end{cases} \tag{7.7}$$

where Υ denotes the set of state and input sequences $(\mathbf{x}_k; \mathbf{u}_k)$ for which all constraints of (7.5) hold. A communication event is then triggered for the controller $\mathcal{C}^i$ with the largest cost decrease $\Delta\mathbf{V}^i_{k,p}$ above a threshold $\gamma \in \mathbb{R}_{>0}$. In other words, the set of communicating controllers is given by

$$\mathcal{T}_{k,p} := \arg\max_{i \in \mathcal{N}} \Delta\mathbf{V}^i_{k,p}, \quad \text{s.t. } \Delta\mathbf{V}^i_{k,p} \geq \gamma. \tag{7.8}$$

If $\Delta\mathbf{V}^i_{k,p} < \gamma$ for all $i \in \mathcal{N}$, no communication takes place, and the algorithm terminates. The computation of the maximum over all controllers $\mathcal{C}^i$ requires some form of global coordination, but can be efficiently implemented in a wide range of shared and distributed communication networks by means of a suitable arbitration scheme (see e.g. [20]). If $\mathcal{T}_{k,p}$ is not a singleton, a priority assigned to each controller may be used for arbitration. This approach also effectively avoids packet collisions (and therefore packet loss) by ensuring that only one controller can access the network at a time. The global state and input sequences are updated as follows:

$$x^j_{k+l|k,p+1} := \begin{cases} \hat{x}^{i*,j}_{k+l|k} & \text{if } j \in \mathcal{N}^i_0, \\ x^j_{k+l|k,p} & \text{otherwise,} \end{cases} \qquad \forall l \in \{0, \dots, N\}, \tag{7.9}$$

$$u^j_{k+l|k,p+1} := \begin{cases} \hat{u}^{i*,j}_{k+l|k} & \text{if } j \in \mathcal{N}^i_0, \\ u^j_{k+l|k,p} & \text{otherwise,} \end{cases} \qquad \forall l \in \{0, \dots, N-1\}, \tag{7.10}$$

with $i \in \mathcal{T}_{k,p}$, i.e. only the planned inputs and states which were optimized by the controller $\mathcal{C}^i$, $i \in \mathcal{T}_{k,p}$ are updated. By definition of $\check{\mathbf{x}}_{k,p}$ and $\check{\mathbf{u}}_{k,p}$ it holds that $\mathbf{V}(\mathbf{x}_{k,p}, \mathbf{u}_{k,p}) = \mathbf{V}(\check{\mathbf{x}}^i_{k,p}, \check{\mathbf{u}}^i_{k,p}) + \mathscr{R}^i_{k,p}$, where $\mathscr{R}^i_{k,p}$ is a remainder that only depends on $\mathbf{u}^j_{k,p}$ and $\mathbf{x}^j_{k,p}$ with $j \notin \mathcal{N}^i_1$. Considering (7.9), it holds for $i \in \mathcal{T}_{k,p}$ that $\mathbf{V}(\mathbf{x}_{k,p+1}, \mathbf{u}_{k,p+1}) = \mathbf{V}(\bar{\mathbf{x}}^i_{k,p}, \bar{\mathbf{u}}^i_{k,p}) + \mathscr{R}^i_{k,p}$. Therefore, by construction of (7.7), the triggering rule (7.8), and (7.9), it holds for $i \in \mathcal{T}_{k,p}$ that $\Delta\mathbf{V}^i_{k,p} = \mathbf{V}(\mathbf{x}_{k,p}, \mathbf{u}_{k,p}) - \mathbf{V}(\mathbf{x}_{k,p+1}, \mathbf{u}_{k,p+1})$.

The communication scheme (e.g. (7.7) to (7.9)) may be repeated if enough time is left before the next sampling instant. Otherwise $\bar{p}_k$ denotes the last iteration performed in time k and the initialization for $k+1$ is computed by shifting the sequences obtained at $(k, \bar{p}_k)$ one time step forward and prolonging it with the terminal control law, i.e.:

$$u^i_{k+l|k+1,0} := u^i_{k+l|k,\bar{p}_k}, \quad \forall l \in \{1, \dots, N-1\} \tag{7.11}$$

$$x^i_{k+l|k+1,0} := x^i_{k+l|k,\bar{p}_k}, \quad \forall l \in \{1, \dots, N\} \tag{7.12}$$

$$u^i_{k+N|k+1,0} := K^i_1 x^i_{k+N|k,\bar{p}_k}, \quad x^i_{k+1+N|k+1,0} := (A^i_1 + B^i_1 K^i_1) x^i_{k+N|k,\bar{p}_k}. \tag{7.13}$$

Each controller $\mathcal{C}^i$ can locally compute these initializations for all $j \in \mathcal{N}^i_1$, because the required values are known through communication. The overall scheme is given by Algorithm 7.1, where t denotes the current absolute time, and k the current time step. The communication scheme is illustrated in Figure 7.1. Similarly to sequential algorithms, a feasible initialization for all $\mathcal{C}^j$, $j \in \mathcal{N}^i_1$ has to be assigned to each controller $\mathcal{C}^i$. Within the algorithm, feasibility of a combination of the current state and input sequences and local candidate solutions is checked, and combinations which are either not feasible or do not lead to a cost reduction are discarded (similar to [83]). In particular, this implies that an update by $\mathcal{C}^i$ at time (k, p) is feasible and decreases the cost as long as $\mathcal{N}^i_1 \cap \mathcal{N}^j_1 = \emptyset$ for all $j \in \cup_{l=0}^{l=p-1} \mathcal{T}_{k,l}$.

Algorithm 7.1: Distributed MPC Algorithm

Initialization: $\mathbf{u}_{0,0}$, $\mathbf{x}_{0,0}$, $(k,p) = (0,0)$
while $k \geq 0$ **do**
  Each controller $\mathcal{C}^i$ applies $u^i_{k|k}$
  The local optimization problem (7.6) is solved in parallel by all $\mathcal{C}^i$, $i \in \mathcal{N}$
  while $t < (k+1)\Delta t - \tau_{\max}$ **do**
    Each $\mathcal{C}^i$ computes $\Delta \mathbf{V}^i_{k,p}$ according to (7.7) and $\mathcal{T}_{k,p}$ according to (7.8)
    Arbitration results in $i^* \in \mathcal{T}_{k,p}$ or $\mathcal{T}_{k,p} = \emptyset$
    if $\mathcal{T}_{k,p} = \emptyset$ **then**
      break
    else
      $\mathcal{C}^{i*}$ sends $(\bar{\mathbf{x}}^{i*}_k; \bar{\mathbf{u}}^{i*}_k)$ to all $\mathcal{C}^j$, $j \in \mathcal{N}^1_{i^*}$.
      Each $\mathcal{C}^j$, $j \in \mathcal{N}^1_{i^*}$ computes the updated sequences for $p+1$ to (7.9) and sets $\bar{p}_k := p$, $p := p+1$
    end if
  end while
  Each $\mathcal{C}^i$ computes the initialization for $(k+1, 0)$ according to (7.11) to (7.13) and sets $(k,p) := (k+1, 0)$.
end while

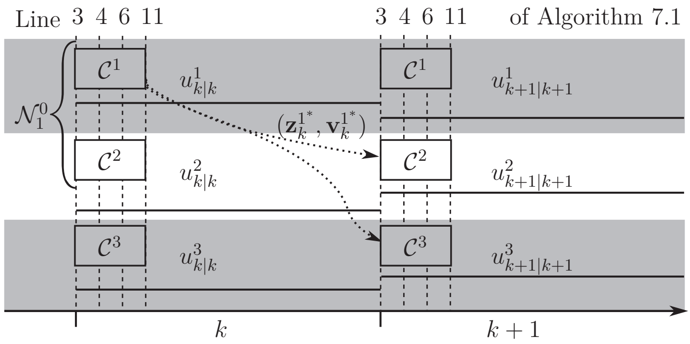

Figure 7.1.: Steps from Algorithm 7.1 (dashed) and communication (dotted, shown for $\mathcal{C}^1$).

With respect to the communication requirements it can be seen that the threshold $\gamma \in \mathbb{R}_{>0}$ can be used to establish a trade-off between communication and closed-loop performance, since updates which improve the cost by less than γ are not communicated and are discarded. The absolute time and communication delays $\tau_k \leq \tau_{\max} \ll \Delta t$ are explicitly considered in line 5 of Algorithm 7.1, i.e. the number of iterations is limited by the communication delay and computation time. These delays are compensated by the fact that the algorithm only optimizes over the input and state sequence starting at $k+1$, while already applying the input for time k which was computed at $k-1$.

7.3. Stability Analysis

A similar approach to the one used in the previous chapters can be used to establish asymptotic stability of the distributed MPC given in Algorithm 7.1. Let $\mathbf{z}_{k,p} = (z_{k|k,p}; \ldots; z_{k+N-1|k,p})$ again denote an extended state vector with $z_{k+l|k,p} = (x_{k+l|k,p}; u_{k+l|k,p})$ for all $l \in \{0, \ldots, N-1\}$, and let $\boldsymbol{\mathcal{Z}}$ denote the set of feasible $\mathbf{z}_{k,p}$ for problem (7.5). Furthermore, $\mathbf{V}_z(\mathbf{z}_{k,p})$ denotes the cost formulated with respect to $\mathbf{z}_{k,p}$, i.e. by substituting the dynamics for $x_{k+N|k}$ such that $\mathbf{V}_z(\mathbf{z}_{k,p}) = \mathbf{V}(\mathbf{x}_{k,p}, \mathbf{u}_{k|k,p})$.

Proposition 7.1. *It holds that $\mathbf{V}_z(\mathbf{z}_{k,p}) = 0$ if $\mathbf{z}_{k,p} = 0$ and there exists $c_d \in \mathbb{R}_{>0}$ such that $\mathbf{V}_z(\mathbf{z}_{k,p})$ is a continuous quadratic function on a ball $\mathcal{B}^{n_z}_{c_d}(0)$ of dimension $n_z = N(n+m)$.*

Proof. Assumption 7.1.1 and Assumption 7.1.2 imply for all $i \in \mathcal{N}$ that $g^i_1 = 0$, $x^i_k = 0 \in \operatorname{int}(\mathbb{X}^i_1)$, and $x^i_k = 0 \notin \mathbb{X}^i_j$ for all $j \in \{2, \ldots, N^i_{\mathfrak{p}^i}\}$. Furthermore, by Assumption 7.2 it holds that $x_k = 0 \in \mathbb{T}$. Therefore, $\mathbf{z}_{k,p} = 0$ implies $x_{k+N|k,p} = 0 \in \mathbb{T}$ and it directly follows that $\mathbf{V}_z(\mathbf{z}_{k,p}) = 0$ if $\mathbf{z}_{k,p} = 0$.

The fact that $\mathbb{T}$ is compact and $0 \in \mathbb{T}$ imply that there exists $c_d \in \mathbb{R}_{>0}$ such that $x_{k+N|k,p} \in \mathbb{T}$ for all $\mathbf{z}_{k,p} \in \mathcal{B}^{n_z}_{c_d}(0) \subseteq (\mathbb{T} \times \mathbb{U}) \times \ldots \times (\mathbb{T} \times \mathbb{U})$. It follows from the linearity of the dynamics on $\mathbb{T}$ that $\mathbf{V}_z(\mathbf{z}_{k,p})$ is a quadratic function for $\mathbf{z}_{k,p} \in \mathcal{B}^{n_z}_{c_d}(0)$. □

Theorem 7.1. *Suppose that Assumption 7.2 is satisfied and a feasible initialization exists. Then, Algorithm 7.1 ensures feasibility for all times, as well as asymptotic stability of the closed loop.*

Proof. By assumption a feasible initialization exists, is known to all subsystems, and it holds that $x_{k+N|k,0} \in \mathbb{T}$. Next, consider the local optimization problem (7.6), the hypothetical update of the state and input variables in (7.9), and note that any infeasible update is discarded because feasibility is checked in (7.8) and all variables required to check feasibility locally are included in the local candidate solution $\bar{\mathbf{x}}^i_{k,p}$, $\bar{\mathbf{u}}^i_{k,p}$. It follows that $x_{k+N|k,p} \in \mathbb{T}$ at any iteration p. The initialization for the time step $k+1$ is computed according to (7.11) to (7.13). Assumption 7.2 then directly implies that $u^i_{k+N|k+1,0} \in \mathbb{U}^i$ and $x^i_{k+N+1|k+1,0} \in \mathbb{T}^i$. In other words, the initialization for $(k+1,0)$ is feasible and remains feasible for any number of iterations p. Feasibility for all times follows by induction over k. Therefore, it is assumed that $\mathbf{z}_{k,p} \in \boldsymbol{\mathcal{Z}}$ in the remainder of the proof.

It directly follows from the definition of the cost (2.5) that there exists $\alpha_1(\|\mathbf{z}_{k,p}\|)$ such that $\alpha_1(\|\mathbf{z}_{k,p}\|) \leq \mathbf{V}_z(\mathbf{z}_{k,p})$ holds. Considering Proposition 7.1 there exists $\tilde{\alpha}_2(\|\mathbf{z}_{k,p}\|)$ such that $\mathbf{V}_z(\mathbf{z}_{k,p}) \leq \tilde{\alpha}_2(\|\mathbf{z}_{k,p}\|)$ holds for all $\mathbf{z}_{k,p} \in \mathcal{B}^{n_z}_{c_d}(0)$. Next, the fact that $\mathbb{T}$ is compact (cf. Assumption 7.2) implies that $\boldsymbol{\mathcal{Z}}$ is compact and it follows from Theorem 2.2 that the maximum $\mathbf{V}_{\max} := \max_{\mathbf{z}_{k,p} \in \boldsymbol{\mathcal{Z}}} \mathbf{V}_z(\mathbf{z}_{k,p})$ is finite. This implies that there exists a constant $c_m \in \mathbb{R}$ such that $\mathbf{V}_{\max} \leq c_m \tilde{\alpha}_2(\|\mathbf{z}_{k,p}\|) =: \alpha_2(\|\mathbf{z}_{k,p}\|)$ for all $\mathbf{z}_{k,p} \in \boldsymbol{\mathcal{Z}} \setminus \mathcal{B}^{n_z}_{c_d}(0)$ and it follows that $\mathbf{V}_z(\mathbf{z}_{k,p}) \leq \alpha_2(\|\mathbf{z}_{k,p}\|)$.

Considering the initialization for $k+1$ based on (7.11) and (7.13) it holds that

$$\begin{aligned}\mathbf{V}_z(\mathbf{z}_{k+1,0}) - \mathbf{V}_z(\mathbf{z}_{k,\bar{p}_k}) = \|x_{k+N|k+1,0}\|^2_Q + \|u_{k+N|k+1,0}\|^2_R + \|x_{k+N+1|k+1,0}\|^2_P \\ -\|x_{k|k,\bar{p}_k}\|^2_Q - \|u_{k|k,\bar{p}_k}\|^2_R - \|x_{k+N|k,\bar{p}_k}\|^2_P.\end{aligned}$$

Furthermore, by Assumption 7.1.3 it holds that

$$\mathbf{V}_z(\mathbf{z}_{k+1,0}) - \mathbf{V}_z(\mathbf{z}_{k,\bar{p}_k}) \leq -\|x_{k|k,\bar{p}_k}\|^2_Q - \|u_{k|k,\bar{p}_k}\|^2_R. \tag{7.14}$$

Applying this inequality and the initialization for $N-1$ time steps results in

$$\mathbf{V}_z(\mathbf{z}_{k+N,0}) - \mathbf{V}_z(\mathbf{z}_{k,\bar{p}_k}) \leq -\textstyle\sum_{l=0}^{N-1}(\|x_{k+l|k,\bar{p}_k}\|^2_Q + \|u_{k+l|k,\bar{p}_k}\|^2_R). \tag{7.15}$$

Finally, by construction of the triggering rules for communication events only updates which decrease the cost by at least $\gamma \in \mathbb{R}_{>0}$ are performed and for any number of iterations p this results in

$$\mathbf{V}_z(\mathbf{z}_{k+N,p}) - \mathbf{V}_z(\mathbf{z}_{k,\bar{p}_k}) \leq -\textstyle\sum_{l=0}^{N-1}(\|x_{k+l|k,\bar{p}_k}\|^2_Q + \|u_{k+l|k,\bar{p}_k}\|^2_R). \tag{7.16}$$

Therefore, the conditions of Theorem 2.1 hold with $L = N$, $d_1 = d_2 = 0$, and it follows that the distributed MPC renders the extended state vector $\mathbf{z}_{k,p}$ of the piecewise affine system asymptotically stable in $\boldsymbol{\mathcal{Z}}$. □

7.4. Numerical Example

The following numerical example concerned with controlling a platoon of $N_s = 7$ identical vehicles was previously published in [43]. The vehicles are modeled by switched second order dynamics with three regions $\mathbb{X}^i_{\mathfrak{p}^i}$, which model in abstraction changes in the dynamics due to shifting gears and nonlinear effects. The states are given by $x^i_k = ((x_p)^i - (i-1)c_s, (\dot{x}_p)^i)$, where $(x_p)^i$ is the position, $(\dot{x}_p)^i$ the velocity, and $c_s = 5\text{m}$ the desired spacing between vehicles. Thus, $(x_p)^{i+1}_k - (x_p)^i_k = c_s$ is equal to $x^{i+1}_k - x^i_k = 0$. The cost is formulated such that the distance of the first vehicle from the desired position and the spacing between each vehicle and its follower are penalized. Furthermore, the problems are interconnected by collision avoidance constraints between subsequent vehicles, i.e. $x^{i+1}_k - x^i_k > -5\text{m}$. The coupling structure is given by

$$\begin{aligned}
&\mathcal{N}^0_1 := \{1,2\}, && \mathcal{N}^0_2 := \{1,2,3\}, && \mathcal{N}^0_3 := \{2,3,4\}, && \mathcal{N}^0_4 := \{3,4,5\},\\
&\mathcal{N}^0_5 := \{4,5,6\}, && \mathcal{N}^0_6 := \{5,6,7\}, && \mathcal{N}^0_7 := \{6,7\}.
\end{aligned}$$

The piecewise affine dynamics are parametrized by

$$\begin{aligned}
A^i_1 &= \begin{bmatrix}1 & 1\\ 0 & 0.95\end{bmatrix}, & B^i_1 &= \begin{bmatrix}0.5\\ 1\end{bmatrix}, & g^i_1 &= \begin{bmatrix}0\\ 0\end{bmatrix},\\
A^i_2 &= \begin{bmatrix}1 & 1\\ 0 & 0.75\end{bmatrix}, & B^i_2 &= \begin{bmatrix}0.4\\ 0.8\end{bmatrix}, & g^i_2 &= \begin{bmatrix}0\\ 1.11\end{bmatrix},\\
A^i_3 &= \begin{bmatrix}1 & 1\\ 0 & 0.75\end{bmatrix}, & B^i_3 &= \begin{bmatrix}0.25\\ 0.5\end{bmatrix}, & g^i_3 &= \begin{bmatrix}0\\ 0\end{bmatrix}.
\end{aligned}$$

The corresponding regions $\mathbb{X}^i_{\mathfrak{p}^i}$ are specified by

$$\begin{aligned}
\mathbb{X}^i_1 &:= \left\{ x^i_k \in \mathbb{R}^{n^i} \,\middle|\, (-0.278+\epsilon)\frac{\text{m}}{\text{s}} \leq \begin{bmatrix}0 & 1\end{bmatrix} x^i_k \leq (5.55-\epsilon)\frac{\text{m}}{\text{s}} \right\},\\
\mathbb{X}^i_2 &:= \left\{ x^i_k \in \mathbb{R}^{n^i} \,\middle|\, (5.55+\epsilon)\frac{\text{m}}{\text{s}} \leq \begin{bmatrix}0 & 1\end{bmatrix} x^i_k \leq (27.78-\epsilon)\frac{\text{m}}{\text{s}} \right\},\\
\mathbb{X}^i_3 &:= \left\{ x^i_k \in \mathbb{R}^{n^i} \,\middle|\, (-5.55+\epsilon)\frac{\text{m}}{\text{s}} \leq \begin{bmatrix}0 & 1\end{bmatrix} x^i_k \leq (-0.278-\epsilon)\frac{\text{m}}{\text{s}} \right\},
\end{aligned}$$

for all $i \in \mathcal{N}$. Simulation results for $N = 30$ and $\gamma = 1$ are shown in Figure 7.2, where the subplots show the position, velocity, and input of each vehicle.

The threshold $\gamma = 1$ was chosen to closely emulate the behavior of time-triggered communication and the scheme is initialized with a suboptimal centralized solution. The last plot shows the number of communication events per time step, which was

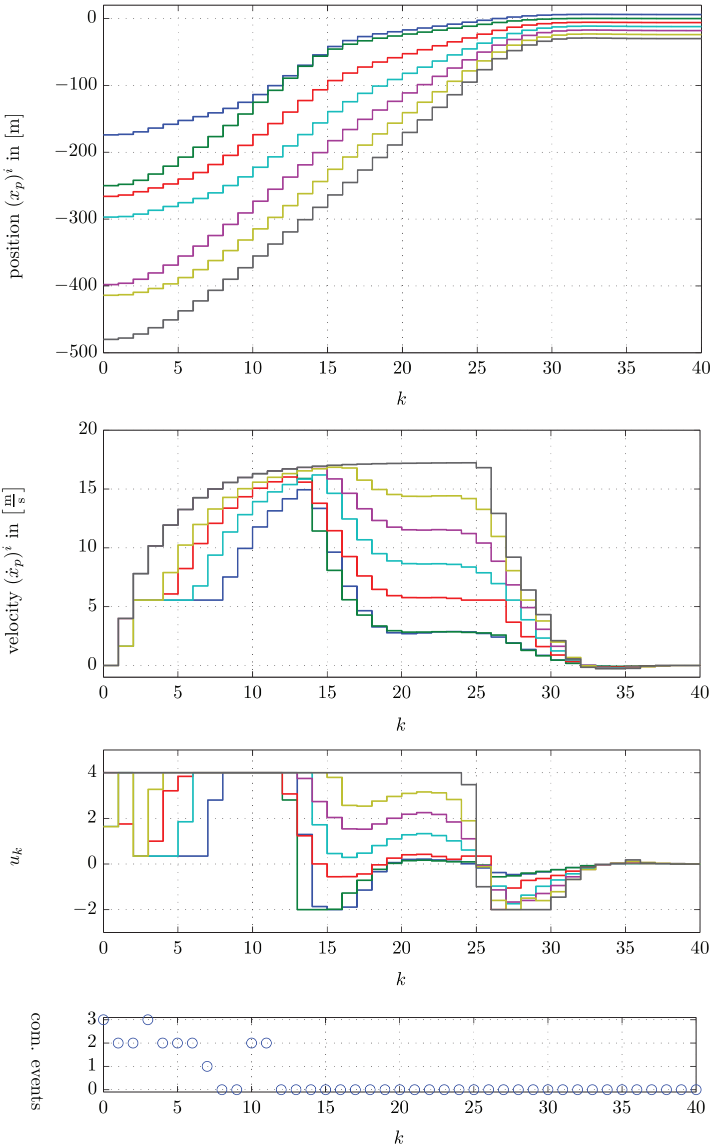

Figure 7.2.: Simulation results for a platoon of $N_s = 7$ vehicles.

limited to three. It can be seen that the vehicles cooperate to reach the desired spacing. For example, $\mathcal{P}^1$ (blue) does not accelerate strongly until ca. $k = 7$, thereby allowing $\mathcal{P}^2$ (green) to reach the desired spacing. For $k \leq 11$ the subsystems communicate frequently, but for $k > 11$ the local solutions do not sufficiently improve the overall cost and are no longer communicated. Nonetheless, the vehicles reach the desired position and no collisions between vehicles occur.

Using CPLEX 12 on an AMD Phenom II X4 920 with 4 GB RAM, the computation times for the local problems (7.6) range from 0.1s to 3s, the sum of local computation times is between 1s and 10s, and a comparable centralized problem typically required between 3s and 100s. In order to allow for real-time operation, shorter prediction horizons, or longer sampling intervals of the subproblems may be used.

7.5. Discussion

In this chapter, a distributed MPC algorithm for dynamically decoupled piecewise-affine systems interconnected by costs and constraints is developed. The main challenge in this setting is that no assumptions about the continuity of the dynamics or the costs can be made and the optimal costs are, in general, not convex. At the same time, even for distributed problems of low to medium scale the computational complexity is relatively high, therefore the centralized problem has to be decomposed to reduce the computational complexity.

The proposed algorithm can be seen as a modification of the algorithms discussed in the previous chapters, in which the weight $w^i_{k,p}$ of the controller $\mathcal{C}^i$ which communicates is equal to one and all other weights are zero, i.e. the controllers optimize the local candidate solutions in parallel but employ sequential communication. In contrast to [102], no assumptions about the continuity of the costs is made, global communication is only partially required in the arbitration phase, and the computation of adjusted weights (which are either one or zero) is carried out in a distributed fashion by the arbitration.

Compared to sequential distributed MPC algorithms (e.g. [83], [50]), the parallel local optimization ensures that the overall computation times scale well with the system size. Furthermore, this enables the compensation of the delays due to computation as well as small communication delays. The communication scheme uses arbitration based on the cost improvement to only grant one subsystem access to the communication network at a time to reduce the load on the network and avoid packet collisions. Furthermore, each subsystem only communicates with a set of neighboring subsystems with which it is interconnected directly or via one other subsystem. The method, including the arbitration scheme, is suitable for multi-hop networks with shared communication channels.

Similarly to sequential distributed MPC algorithms the algorithm developed in this chapter does not guarantee convergence to the optimal centralized solution in each time step. Due to the non-convexity of the optimal costs this problem remains open except for very limited classes of hybrid systems. For example, in [14] a parallel algorithm based on dual decomposition and Lagrangian relaxation is proposed which converges to a so called “differential maximum” of the dual function over the iterations p. However, the hybrid systems considered in [14] are restricted to linear dynamics with discrete inputs. In contrast, the algorithm developed in this chapter utilizes parallel optimization of the primal variables and can be applied to piecewise affine dynamics which allow to approximate a wide range of hybrid dynamical systems.

8. Discussion and Outlook

In this thesis different methods for distributed model predictive control with a focus on linear discrete-time systems and event-based communication have been presented. Specifically, in Chapter 3 concepts from robust optimization are used to deal with time-varying delays in distributed MPC of dynamically decoupled discrete-time linear subsystems interconnected by costs and constraints. Based on results for cooperative MPC with time-triggered communication given in Chapter 4, two approaches for event-based communication are derived in Chapter 5 and Chapter 6 for linear discrete-time systems and different classes of interconnections. Finally, in Chapter 7 some results for distributed MPC of piecewise affine systems are given. In this chapter, the different methods are compared, the results are summarized, and some directions for future research are discussed.

8.1. Summary and Comparison of the Proposed Methods

Throughout this thesis, distributed model predictive control algorithms with parallel computation and communication between the local controllers are considered. The different methods can be broadly categorized into methods with time-triggered periodic communication (i.e. in every time step or iteration) and methods with event-based communication in which communication only occurs if a triggering condition holds. A further classification can be made based on the class of dynamical systems and interconnections considered. In the following the results for the different algorithms are briefly summarized and compared. Because the algorithms discussed are inherently suboptimal, stability analysis can be a challenging problem. To resolve this issue, conditions for input-to-state practical stability less restrictive than those available in the literature are presented in Section 2.3.

The algorithm presented in Chapter 3 for dynamically decoupled linear-discrete time systems interconnected by costs and constraints utilizes ideas from robust optimization to ensure that constraints are robustly satisfied even in the presence of communication delays. In particular, the controllers communicate once per time step over a communication network with time-varying bounded delays, which may be larger than one time step. These delays are explicitly considered in the local optimization problems by means of optimizing over a delayed affine feedback. Local constraints then ensure that each subsystem does not deviate too much from its previously planned and communicated trajectory, such that robust constraint satisfaction is guaranteed. However, this rather strong result comes at the expense of a high computational complexity of the local optimization problems and is only applicable to a limited class of interconnection graphs $\mathscr{G}$. Specifically, Assumption 3.3 essentially states that the distributed, delayed, terminal control law renders the terminal constraint invariant even when other subsystems do not cooperate. This assumption is quite strong but similar in scope to the assumptions made in [50], and is implicitly made in many other works which rely on small gain type results. In numerical examples and using commercial solvers the computational complexity of the algorithm is about one to two orders of magnitude higher compared to a nominal MPC problem. Even when resorting to more efficient approaches (cf. [40]) the worst-case computational complexity of the local problems is still much higher than that of comparable state of the art methods in nominal MPC (cf. [56]). On the other hand, the algorithm provides good robustness and performance in the presence of large communication delays, utilizes parallel computation to avoid drawbacks of sequential algorithms, and can also be applied to control tasks such as synchronization. Therefore, this algorithm is well suited for problems such as formation control over unreliable communication networks.

In contrast, the cooperative distributed MPC algorithm with parallel computation discussed in Chapter 4 relies heavily on communication and cooperation of the local controllers. The results in Chapter 4 focus on three specific classes of interconnected discrete-time linear systems which result in decoupled input constraints (cf. Cases 2.2 to 2.4) and cover application scenarios such as formation control, water or energy distribution systems, and chemical processes. For these cases the original centralized problem can be solved to optimality by the iterative cooperative distributed MPC algorithm first proposed in [112]. However, the scalability of this method is limited by the fact that each controller has to communicate with every other controller in each iteration. Furthermore, the original works on this algorithm do not provide any convergence rate. Without a result on the convergence rate it is difficult to judge the scalability of the method with respect to the number of subsystems. Therefore, in Chapter 4 a bound on the convergence rate is given, which depends on the strength of coupling between subsystems in the centralized MPC problem as well as the number of subsystems. Furthermore, these results give some insight into the role of the information exchanged between the controllers and allows choosing parameters used in the algorithm to ensure fast convergence.

Based on these results, the cooperative distributed MPC algorithm is combined with a suitable triggering condition for communication events in Chapter 5. In the resulting algorithm with event-based communication, each controller solves the same local optimization problem used in the cooperative distributed MPC discussed in Chapter 4, but results of this optimization are only communicated to all other controllers if the triggering condition holds. Local optimization results which are not communicated are discarded. This results in an iterative algorithm with significantly reduced communication requirements and, in many cases, slightly faster convergence than the original algorithm. The improved convergence speed observed in the simulation examples can be formally explained by the fact that locally optimized candidate sequences which do not result in a large global cost decrease are discarded. Compared to the algorithm with time-triggered communication, this gives higher priority to the candidate solutions which provide a larger global cost decrease. Furthermore, based on the triggering conditions for communication a distributed stopping criterion is given which can be verified without additional computations or communication and ensures bounded suboptimality. Overall this improves the scalability of the cooperative distributed MPC algorithm discussed in Chapter 4 because significantly less communication is required, the convergence speed is slightly improved, and the computational complexity is not increased. On the other hand, this cooperative distributed MPC with event-based communication requires a controller $\mathcal{C}^i$ to send information to all other controllers $\mathcal{C}^j$, $j \in \mathcal{N} \setminus i$ if its triggering condition holds. In other words, this approach only answers the question of when to communicate. Furthermore, it is not clear which impact uncertainties induced by the communication network have on the algorithm. In comparison with the robust distributed MPC the computational complexity of this algorithm is much lower, but the algorithm is iterative and requires a communication network which does not induce any uncertainties. However, in some cases the algorithm can be terminated at any iteration and asymptotic stability is still guaranteed. In contrast, the robust distributed MPC only ensures practical input-to-state stability.

To further reduce the load on the communication network, the cooperative distributed MPC algorithm is combined with a triggering condition which triggers communication events between pairs of controllers in Chapter 6. In other words, these triggering conditions are used to decide when and between which controllers to communicate. These triggering conditions are obtained by computing how the optimizer of the local MPC problems changes if the communicated information changes. Based on this result it can be determined how accurately a controller needs to know the state and input sequence of each interconnected subsystem and controller. Compared to the algorithm with event-based communication presented in Chapter 5, this results in a further reduction of the load on the communication network. In particular, since the controllers no longer need to communicate with all other controllers, the scalability of the algorithm is improved. However, there are some drawbacks: the algorithm in general converges more slowly than the algorithm discussed in Chapter 4 and Chapter 5, it is not applicable to one of the three classes of interconnected linear discrete-time systems under consideration because the terminal equality constraint required in Case 2.2 can no longer be included in the local optimization, and verifying the stopping condition which ensures bounded suboptimality requires additional communication. Nonetheless, this algorithm can be used to further reduce the load on the communication network. Furthermore, the framework proposed in Chapter 6 allows to quantify the impact of time varying communication delays and packet loss on the closed-loop performance of the distributed MPC algorithm. The results suggest that the number of messages exchanged between the controllers has been reduced to a point where every message is important and packet loss and small delays (i.e. smaller than the sampling time) may strongly degrade closed-loop performance.

Finally, in Chapter 7 a modification of the algorithm developed in Chapter 5 is presented which is tailored to a class of dynamically decoupled hybrid systems described by discrete-time piecewise affine dynamics. Because the original centralized MPC problem is no longer convex, no results on the suboptimality can be given and only one controller may update the input sequence per iteration. The algorithm uses a variant of the triggering condition proposed in Chapter 5 and compensates both for computation times and small communication delays. Overall this approach significantly reduces the load on the communication network and also greatly reduces the computational complexity compared to a centralized formulation.

The main contributions of this thesis can be summarized as follows:

- Robust optimization is utilized to ensure constraint satisfaction and robust stability for distributed MPC of discrete-time linear subsystems interconnected by costs and constraints in the presence of bounded time-varying communication delays. The delays are partially compensated by communicating timestamped input sequences. As discussed in Section 1.2, only few results on distributed MPC over delayed communication networks are available in the literature. Compared to those results, the contribution of this thesis is that time-varying delays are explicitly considered in the local optimization.

- Rigorous convergence results are given for the cooperative distributed MPC algorithm first proposed in [112] which relate the convergence of the algorithm to a measure for the coupling strength of the subsystems in the original (centralized) MPC problem, the number of subsystems, and parameters used in the algorithm. Based on these results, the parameters of the algorithm are optimized for fast convergence. No results in this direction were known in the literature before.

- Event-based communication protocols are proposed which significantly reduce the load on the communication network by avoiding communication of information which does not improve closed-loop performance, is redundant, or not required by the receiving controller. The results focus on linear discrete-time systems and some results are extended to the case of hybrid systems given by dynamically decoupled piecewise affine discrete-time dynamics. Despite the vast literature on event-based control in the context of networked control systems, no results on event-based communication in distributed MPC are available at the moment.

- Conditions for input-to-state practical stability in terms of ISS-Lyapunov functions are proposed which are less restrictive than those available in the literature and allow to establish robust stability of suboptimal distributed MPC without additional measures commonly used in suboptimal MPC (e.g. stability constraints, reinitialization of the controllers in a neighborhood of the control goal). Furthermore, the results allow to consider ISpS with respect to a set of states (e.g. consensus) and are well suited to analyze stability of suboptimal distributed MPC with event-based and / or uncertain communication.

Furthermore, all algorithms developed in this thesis utilize parallel local computation to avoid the potentially large aggregated computation times of sequential approaches.

8.2. Outlook

The main ideas developed in this thesis can be extended in various directions and a number of open questions remain. In the following some possible directions for future research are discussed:

- The assumptions made in distributed MPC with respect to the local dynamics, properties of the communication network and types of interconnections largely differ (see the discussion in Section 1.2). While the results given in this thesis are applicable to a wide range of control problems, it would be desirable to extend the results to a wider class of interconnections and dynamics. In particular, the results in Chapter 3 could be extended to include coupled dynamics and more general coupling graphs $\mathscr{G}$. Similarly, the algorithms presented in Chapter 4 and Chapter 5 are in theory applicable to subsystems with interconnected constraints. However, as discussed in the corresponding chapters, extensions are required to ensure convergence to the centralized optimal solution. Finally, in theory the triggering conditions developed in Chapter 5 should be applicable to a wide class of distributed MPC algorithms. Therefore applying these ideas to different distributed MPC algorithms may provide some interesting results.

- In many cases the interconnections between subsystems are rather sparse and only a few neighboring subsystems are strongly interconnected. Therefore, using hierarchical approaches may prove useful in order to obtain faster convergence and reduce the amount of communication. Some results in this direction can be found in [103], where subsystems are grouped into clusters and communication within a cluster is used more frequently than between clusters. However, it is not clear how to perform the clustering and to the best of the author's knowledge, no results have been published on how to effectively re-partition the overall system if there exists some freedom in choosing the partition of the overall system. The results of Theorem 4.1, in particular (4.18) and the preceding analysis, link the convergence of the distributed

MPC algorithm to the structure of the Hessian. This gives some insight into the problem of partitioning the overall system.

- Recently, the field of plug-and-play model predictive control (cf. [59]) has gained considerable interest. The main idea behind plug-and-play control is that controllers (and their corresponding subsystems) may join or leave the overall system during runtime if the distributed controller can be reconfigured to ensure stability. This entails a reconfiguration of the distributed control law as well as reconfiguration of the communication topology. The paradigm of event-based communication as well as the sensitivity analysis performed in Chapter 6 may be useful in this regard to automatically adapt the communication topology if the interconnection structure changes (e.g. by updating the local triggering functions).

- While the cooperative algorithm with event-based communication developed in Chapter 6 significantly reduces the number of messages sent over the communication network, the simulation results show that a large number of small messages may be required to ensure a synchronous termination of the algorithm with guaranteed suboptimality. This is not desirable from both a practical and a theoretical point of view. Therefore, to achieve better scalability of distributed MPC algorithms the results have to be extended to asynchronous iterations and stopping criteria. Extensions in this direction may be based on only applying the algorithm synchronously within groups of strongly interconnected subsystems.

- Finally, the distributed MPC algorithms proposed in this thesis implicitly assume some degree of cooperation between the controllers and are based on the assumption that models describing the dynamics of the subsystems are either available to the controllers of neighboring subsystems (e.g. Algorithm 3.1) or to the controllers of all subsystems (e.g. Algorithm 4.1 and Algorithm 5.1). This raises two questions. First of all, in truly large scale systems, such as power grids, it may not be practical or desirable to share local models with all controllers. For example, two competing utility companies may not want to share the dynamic models or local control goals of their power plants. While many distributed MPC algorithms are available that do not require sharing the local models (e.g. [50], [91], [83]) these algorithms require global model knowledge in the design phase to verify the underlying assumptions. Even if this step can be solved in distributed fashion, these algorithms rely on communicating both the predicted state and input trajectories. This information can be directly used to reconstruct the local closed-loop dynamics and therefore does not provide much (if any) additional privacy while resulting in additional communication. Secondly, a single malicious controller could easily exploit the assumption that the controllers cooperate as well as the knowledge of the dynamics of other subsystems to destabilize the overall system by communicating false information. Therefore, it is of paramount importance to develop methods which detect malicious controllers and do not require exact knowledge of all dynamic models or communication of full input and state trajectories. While this problem is quite complicated in general, a starting point may be to analyze how strongly the local optimization problems depend on the model data and local costs of interconnected subsystems.

Appendix A. Formulation of Costs and Constraints

A.1. Condensed Problem Formulation

All future states can be expressed based on the current state and predicted input sequence $\mathbf{u}_k$ as follows:

$$\underbrace{\begin{bmatrix} x_{k|k} \\ x_{k+1|k} \\ \vdots \\ x_{k+N|k} \end{bmatrix}}_{\mathbf{x}_k} = \underbrace{\begin{bmatrix} I_n \\ A \\ \vdots \\ A^N \end{bmatrix}}_{=:\mathbf{A}} x_k + \underbrace{\begin{bmatrix} 0 & 0 & \cdots & 0 \\ B & 0 & \cdots & 0 \\ A & B & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ A^{N-1}B & A^{N-2}B & \cdots & B \end{bmatrix}}_{=:\mathbf{B}} T_{up}\ \mathbf{u}_{k|k,p}, \tag{A.1}$$

where $T_{up} \in \mathbb{R}^{Nm \times Nm}$ is a permutation matrix, such that:

$$T_{up}\mathbf{u}_k = (u_{k|k}; \ldots ; u_{k+N-1|k}).$$

With the Kronecker product $\otimes$, the cost over the prediction horizon N is given by $V(\mathbf{x}_k, \mathbf{u}_k) = \mathbf{x}_k^T \mathbf{Q} \mathbf{x}_k + \mathbf{u}_k^T \mathbf{R} \mathbf{u}_k$, with $\mathbf{Q} := \text{blkdiag}(I_N \otimes Q, P)$, $\mathbf{R} := (T_{up})^T (I_N \otimes R) T_{up}$, where I_N denotes the identity matrix of dimension N. Substituting (A.1) one obtains $H := \mathbf{B}^T \mathbf{Q} \mathbf{B} + \mathbf{R}$, $F := 2\mathbf{A}^T \mathbf{Q} \mathbf{B}$, $H_x := \mathbf{A}^T \mathbf{Q} \mathbf{A}$. For Case 2.3 and Case 2.4 local input constraints

$$\mathbf{U}^i(x_k^i) := \{\mathbf{u}_k^i \in \mathbb{R}^{m^i N} | \mathbf{C}_{\mathbf{U}}^i \mathbf{u}_k^i \le \mathbf{b}_{\mathbf{U}}^i(x_k^i)\}$$

over the horizon $N-1$ are obtained by considering the local input constraints, as well as substituting the dynamics into the state and terminal constraints, such that

$$\mathbf{C}_{\mathbf{U}}^i := \begin{bmatrix} I_N \otimes C_{\mathbb{U}}^i \\ \begin{bmatrix} I_{N-1} \otimes C_{\mathbb{X}}^i & 0 \\ 0 & C_{\mathbb{T}}^i \end{bmatrix} \mathbf{B}^i \end{bmatrix},$$

$$\mathbf{b}_{\mathbf{U}}^i(x_k^i) := \begin{bmatrix} 1_N \otimes b_{\mathbb{U}}^i \\ \begin{bmatrix} 1_{N-1} \otimes b_{\mathbb{X}}^i \\ b_{\mathbb{T}}^i \end{bmatrix} - \begin{bmatrix} I_{N-1} \otimes C_{\mathbb{X}}^i & 0 \\ 0 & C_{\mathbb{T}}^i \end{bmatrix} \mathbf{A}^i x_k^i \end{bmatrix},$$

where $\mathbf{A}^i$ and $\mathbf{B}^i$ are the local dynamics (A^i, B^i) over the prediction horizon. In Case 2.4 no state constraints are present, i.e. $C_{\mathbb{X}}^i = 0$, $b_{\mathbb{X}}^i = 0$, $C_{\mathbb{T}}^i = 0$, and $b_{\mathbb{T}}^i = 0$. For the derivation of A and B and the constraints $\mathbf{U}^i(x_k)$ for Case 2.2, see [104] and the references given therein.

A.2. Terminal Costs and Constraints for Dynamically Decoupled Systems

Because the pairs (A^i, B^i) are stabilizable, there exists a control law K and matrix $P_d = P_d^T \succ 0$ with

$$K = \text{blkdiag}(K^1, \dots, K^{N_s}), \quad P_d = \text{blkdiag}(P_d^1, \dots, P_d^{N_l}),$$

such that

$$P_d - (A + BK)^T P_d (A + BK) \succ c_Q I_n$$

holds for some $c_Q \in \mathbb{R}_{>0}$. It directly follows that there exists

$$P = \text{blkdiag}(P^1, \dots, P^{N_l})$$

with $\quad P = P^T \succ 0$ such that

$$P - (A + BK)^T P (A + BK) \succeq Q - K^T R K^T.$$

A polyhedral terminal set $\mathbb{T}$ can be obtained by computing an invariant set of $(A + BK)$ which lies in $\mathbb{X}$ (see [12]). In particular, the terminal set $\mathbb{T}$ is fully decoupled, if a hypercube $\mathcal{H}_c^n := \{x_k \in \mathbb{R}^n | \|x_k\|_\infty \leq c\} \subseteq \mathbb{X}$ centered on 0 is used to initialize such a procedure. Finally, the assumptions made with respect to the constraints in Section 2.1 ensure that the terminal set is non-empty.

List of Symbols

Equalities, Inequalities and Definitions

$\mathscr{A} := \mathscr{B}$	$\mathscr{A}$ is defined to be $\mathscr{B}$
$\mathscr{A} = \mathscr{B}$, $\mathscr{A} \neq \mathscr{B}$	$\mathscr{A}$ is equal to $\mathscr{B}$, $\mathscr{A}$ is not equal to $\mathscr{B}$
$\mathscr{A} \in \mathscr{B}$, $\mathscr{A} \notin \mathscr{B}$	$\mathscr{A}$ is an element of a set $\mathscr{B}$, $\mathscr{A}$ is not an element of a set $\mathscr{B}$
$\mathscr{A} \subset \mathscr{B}$, $\mathscr{A} \subseteq \mathscr{B}$	a set $\mathscr{A}$ is a subset of a set $\mathscr{B}$, $\mathscr{A}$ is a subset of or equal to the set $\mathscr{B}$
$\mathscr{A} < \mathscr{B}$, $\mathscr{A} \leq \mathscr{B}$	component wise inequality of vectors $\mathscr{A} = (a_1; \ldots; a_n) \in \mathbb{R}^n$ and $\mathscr{B} = (b_1; \ldots; b_n) \in \mathbb{R}^n$: $a_i < b_i$ for all $i \in \{1, \ldots, n\}$, $a_i \leq b_i$ for all $i \in \{1, \ldots, n\}$
$\mathscr{A} \prec \mathscr{B}$, $\mathscr{A} \preceq \mathscr{B}$	inequality of symmetric positive definite matrices $\mathscr{A} \in \mathbb{R}^{n \times n}$ and $\mathscr{B} \in \mathbb{R}^{n \times n}$: $y^T \mathscr{A} y < y^T \mathscr{B} y$ for all $y \in \mathbb{R}^n$, $y^T \mathscr{A} y \leq y^T \mathscr{B} y$ for all $y \in \mathbb{R}^n$

Functions

$\alpha_c : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$	comparison function
$\beta_c : \mathbb{R}_{\geq 0} \times \mathbb{N}_0 \to \mathbb{R}_{\geq 0}$	ISpS comparison function
$\gamma_c : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$	ISpS comparison function
$\lfloor \cdot \rfloor : \mathbb{R} \to \mathbb{Z}$	floor function, $\lfloor y \rfloor := \max_{i \in \mathbb{Z}} i \leq y$
$f : \mathbb{X} \times \mathbb{U} \to \mathbb{X}$	uncontrolled dynamics of the overall system
$f_d : \mathbb{X} \times \mathbb{W} \to \mathcal{X}$	perturbed dynamical system
$g_d : \mathbb{R}^h \to \mathbb{R}$	Lagrange dual function with h dual multipliers
$\mathcal{L} : \mathbb{R}^y \times \mathbb{R}^h \to \mathbb{R}$	Lagrangian with y primal variables and h dual multipliers
$\mathbf{V} : \mathbf{X} \times \mathbf{U} \to \mathbb{R}_{\geq 0}$	objective function
$\mathcal{V} : \mathcal{X} \to \mathbb{R}_{\geq 0}$	ISpS-Lyapunov function

$V : \mathbb{X} \times \mathbf{U} \to \mathbb{R}_{\geq 0}$	objective function, future states $x_{k+l\|k}$ for $l \in \{1, \ldots, N\}$ are eliminated by substituting $f(x_k, u_k)$ into $\mathbf{V}(\mathbf{x}_k, \mathbf{u}_k)$.
$V^* : \mathbb{X} \to \mathbb{R}_{\geq 0}$	objective function $V(x_k, \mathbf{u}_k^*)$ for optimal input sequence $\mathbf{u}_k^*$
$\mathbf{V}_z : \boldsymbol{\mathcal{Z}} \to \mathbb{R}_{>0}$	objective function formulated using $\mathbf{z}_k$ obtained by substituting $x_{k+N\|k} := f(x_{k+N-1\|k}, x_{k+N-1\|k})$ into $\mathbf{V}(\mathbf{x}_k, \mathbf{u}_k)$.
$V_d : \mathbb{X} \times \mathbf{U} \to \mathbb{R}_{\geq 0}$	cost difference to the optimum: $V_d(x_k, \mathbf{u}_{k,p}) := V(x_k, \mathbf{u}_{k,p}) - V(x_k, \mathbf{u}_k^*)$

General

$\otimes$	Kronecker product
$(\cdot)_{[k_1:k_2]}$	time sequence from k_1 to k_2: $\omega_{[k_1:k_2]} = (\omega_{k_1}; \ldots; \omega_{k_2})$
$(\cdot)_{k,p}$	time-varying value at time (k, p)
$(\cdot)^i$	value associated with subsystem $\mathcal{P}^i$ or controller $\mathcal{C}^i$
$(\cdot)^{\setminus i}$	vector or matrix with the entries corresponding to $\mathcal{P}^i$ or $\mathcal{C}^i$ removed
$\hat{(\cdot)}^{i,j}$	value used by controller $\mathcal{C}^i$ for a value of subsystem $\mathcal{P}^j$
$(\cdot)_{\mathfrak{p}}$	matrix, vector, or set associated with a region $\mathfrak{p}$
$(\cdot)_{k\|k,p}$	vector or feedback matrix predicted for time k at time (k, p)
$(\cdot)^*$	optimal value, e.g. optimal input at time k: u_k^*
$\tilde{(\cdot)}^i$	augmented value, e.g. augmented state vector $\tilde{x}_k^i := (x_k^i; x_k^j)$
$\mathcal{C}^i$	local model predictive controller of subsystem $\mathcal{P}^i$
$\mathscr{C}_{k,p}$	time-varying communication graph: $\mathscr{C} = (\mathcal{N}, \mathcal{E}_k)$
$\delta(\cdot)$	uncertainty of a value, e.g.. $\delta\hat{u}_k^{i,j} = \hat{u}_k^{i,j} - u_k^j$
$\mathcal{I}$	edges of the interconnection graph $\mathscr{G}$: $\mathcal{I} \subseteq \mathcal{N} \times \mathcal{N}$
$\mathscr{G}$	interconnection graph: $\mathscr{G} = (\mathcal{N}, \mathcal{I}_G)$
(k, p)	discrete time k and iteration index p at time k
t	discrete time k
N_s, N_s^i	number of subsystems, of subsystems interconnected with $\mathcal{P}^i$

$\mathcal{P}^i$	dynamical subsystem i
p	number of iterations
$\mathcal{E}_{k,p}$	edges of the communication graph $\mathscr{C}_{k,p}$: $\mathcal{E}_{k,p} \subseteq \mathcal{N} \times \mathcal{N}$
Δt	sampling time, i.e. time increment for each time step k
Δt_p	time increment for each iteration p
$\tau_{k,p}^{i,j}$	time-varying communication delay between $\mathcal{C}^i$, $\mathcal{C}^j$
$\tau_{\max}$	maximal communication delay between any pair of controllers: $\tau_{\max} \geq \tau_k^{i,j}$ for all $(i,j) \in \mathcal{E}_{k,p}$, $(k,p) \geq (0,0)$
t	continuous time t

Scalars and Constants

$\beta_{k,p}$	convergence rate of distributed algorithm at time k and iteration p.
$\bar{\beta}$	upper bound on convergence rate $\beta_{k,p}$
c_H^i	coupling strength in Hessian matrix H
$c_{(\cdot)}$	constant scalars
d_c	constant in ISpS comparison function
ϵ	small positive constant
ϵ_n	numerical tolerance
γ	threshold of triggering conditions for a communication event
λ_i	multiplier of the i-th inequality constraints
$\mu_{k,p}^i$	multipliers used to analyze cooperative distributed MPC
N	finite prediction horizon: $N \in \mathbb{N}$, $N \geq 2$
ν_i	multiplier of i-th equality constraints
$p_{\max}$	maximal number of iterations per time step
θ	step size, e.g.: $y_\theta = (1-\theta)y_1 + \theta y_2$
$w_{k,p}^i$	weight used in cooperative distributed MPC

ζ	threshold used in the stopping condition of distributed MPC with sensitivity based communication

Sets

$\{\cdot\}$	discrete set
$\setminus$	relative complement: $\mathcal{S}_1 \setminus \mathcal{S}_2 := \{y \in \mathcal{S}_1 \,\vert\, y \notin \mathcal{S}_2\}$
$\times$	cartesian product of sets: $\mathcal{S}_1 \times \mathcal{S}_2 := \{(y_1; y_2) \,\vert\, y_1 \in \mathcal{S}_1, y_2 \in \mathcal{S}_2\}$
$\cap$	intersection of sets: $\mathcal{S}_1 \cap \mathcal{S}_2 := \{y \,\vert\, y \in \mathcal{S}_1$ and $y \in \mathcal{S}_2\}$
$\oplus$	Minkowski sum: $\mathcal{S}_1 \oplus \mathcal{S}_2 := \{y \,\vert\, \exists y_1 \in \mathcal{S}_1, y_2 \in \mathcal{S}_2 : y = y_1 + y_2\}$
$\cup$	union of sets: $\mathcal{S}_1 \cup \mathcal{S}_2 := \{y \,\vert\, y \in \mathcal{S}_1$ or $y \in \mathcal{S}_2\}$
∂	boundary of a set: $\partial\mathcal{S} := \mathcal{S} \setminus \mathrm{int}(S)$
$\mathbb{N}$, $\mathbb{N}_0$	set of natural numbers, set of natural numbers including 0
$\mathbb{R}_{>0}$, $\mathbb{R}_{\geq 0}$	set of positive real numbers, set of non-negative real numbers
$\mathbb{R}^n$	set of real vectors with n elements
$\bar{\mathbb{R}}$	set of extended real numbers: $\bar{\mathbb{R}} := \mathbb{R} \cup \{-\infty, \infty\}$
$\mathbb{Z}$	set of integers
$\mathcal{B}_r^n(y)$	closed ball of dimension $n \in \mathbb{N}$, radius $r \in \mathbb{R}_{>0}$, and with center $y \in \mathbb{R}^n$: $\mathcal{B}_r^n(y) := \{\bar{y} \in \mathbb{R}^n \,\vert\, \Vert\bar{y} - y\Vert \leq r\}$
$\mathrm{card}(\mathcal{S})$	cardinality of a set $\mathcal{S}$
$\boldsymbol{\mathcal{D}}$	uncertainty affecting robust distributed MPC
$\mathcal{H}_c^n$	hypercube of dimension n, diameter c, and with center 0: $\mathcal{H}_c^n := \{x_k \in \mathbb{R}^n \vert \Vert x_k\Vert_\infty \leq c\}$
$\mathrm{int}(\mathcal{S})$	interior of the set $\mathcal{S}$
$\mathcal{K}^i$	admissible affine feedback policies
$\mathcal{N}$	set of subsystems: $\mathcal{N} = \{1, \ldots, N_s\}$
$\mathcal{N}^i$	set of subsystems $\mathcal{P}^j$ interconnected with $\mathcal{P}^i$
Σ	set specifying control objectives
$\mathbb{T}$	terminal constraint: $\mathbb{U} \subseteq \mathbb{X}$

$\mathbb{U}$	input constraint: $\mathbb{U} \subseteq \mathbb{R}^m$
$\mathbf{U}$	input constraints over the prediction horizon N: $\mathbf{U} := \mathbb{U} \times \ldots \times \mathbb{U} \subseteq \mathbb{R}^{Nm}$
$\Delta\mathbb{U}$	constraint on input uncertainties, e.g.: $\delta\hat{u}_k^{i,j} \in \Delta\mathbb{U}_k^i$
$\mathbb{W}$	compact set of disturbances
$\mathbb{X}$	state constraint: $\mathbb{X} \subseteq \mathbb{R}^n$
$\mathbf{X}$	state constraints over the prediction horizon N: $\mathbf{X} := \mathbb{X} \times \ldots \times \mathbb{X} \subseteq \mathbb{R}^{(N+1)n}$
$\mathcal{X}$	subset of or equal to the state constraints: $\mathcal{X} \subseteq \mathbb{X}$
$\mathcal{Z}$	feasible set of extended state vectors $\mathbf{z}_k$

Vectors and Matrices

$0_{y\times h}$	zero matrix $0_{y\times h} \in \mathbb{R}^{y\times h}$ of dimension $y \times h$
$1_{y\times h}$	matrix or vector of ones of dimension $y \times h$
$(y_1; \ldots; y_s)$	stacked column vector or matrix $y_j \in \mathbb{R}^{n_{y_j}\times m_y}$: $(y_1; \ldots; y_s) := \left[(y_1)^T \quad \ldots \quad (y_s)^T\right]^T$
$(\cdot)^T$	transpose of a vector or matrix
$(\cdot)^{-1}$	inverse of a matrix
A, B, g	matrices and vector defining linear or piecewise affine dynamics
$\mathbf{A}$, $\mathbf{B}$, $\mathbf{G}$	matrices defining linear dynamics with uncertain inputs over the whole prediction horizon
$C_{(\cdot)}$, $b_{(\cdot)}$	matrix and vector defining a polytope, e.g: $\mathbb{X} := \{x \mid C_{\mathbb{X}}\, x \le b_{\mathbb{X}}\}$
H	Hessian matrix of a quadratic function
I_y	identity matrix $I_y \in \mathbb{R}^{y\times y}$ of dimension y:
K	feedback matrix
λ	vector of Lagrange multipliers of inequality constraints
$\lambda_{\min}(\cdot)$, $\lambda_{\max}(\cdot)$	smallest and largest eigenvalue of a matrix

$\mu_{k,p}$	vector of multipliers $\mu^i_{k,p}$
$\\|(\cdot)_{[k_1:k_2]}\\|$	maximal norm of a sequence: $\\|\omega_{[k_1:k_2]}\\| := \max_{l \in \{k_1,\ldots,k_2\}} \\|\omega_l\\|$
$\\|\cdot\\|$	Euclidean norm of a matrix or vector
$\\|\cdot\\|_\infty$	infinity norm of a matrix or vector
$\\|\cdot\\|_C$	norm with respect to a set C, e.g.: $\\|x_k\\|_\Sigma := \inf_{z\in\Sigma} \\|x_k - z\\|$
$\\|\cdot\\|_M$	weighted norm with weight M, e.g.: $\\|x\\|_M^2 := x^T M x$
ν	vector of Lagrange multipliers of equality constraints
ω_k	disturbance acting on the dynamics f_d
P	terminal weight for $x_{k+N\|k}$: $P = P^T \succeq 0$
$\boldsymbol{\phi}^i_{k,p}$	local input sequence optimized by distributed MPC with sensitivity based communication
Q	weighting matrix for the state x_k: $Q = Q^T \succeq 0$
$\boldsymbol{\rho}^i_{k,p}$	local input sequence optimized by $\mathcal{C}^i$
R	weighting matrix for the input u_k: $R = R^T \succeq 0$
$T_{(\cdot)}$	transformation / permutation matrix
$\Delta\mathbf{u}^i_{k,p}$	difference to the global optimizer for $\mathcal{P}^i$ at time (k,p)
$\Delta\mathbf{u}_{k,p}$	difference to the global optimizer at time (k,p)
u_k	input of the overall system at time k
u^i_k	input of subsystem $\mathcal{P}^i$ at time k
$\mathbf{u}_{k,p}$	input sequence over the horizon $N \geq 2$ (p is omitted if no iterations are performed): $\mathbf{u}_{k,p} := (u_{k\|k,p}; \ldots; u_{k+N-1\|k,p})$
$\mathbf{u}^i_{k,p}$	input sequence of $\mathcal{P}^i$ over the horizon $N \geq 2$ (p is omitted if no iterations are performed): $\mathbf{u}^i_{k,p} := (u^i_{k\|k,p}; \ldots; u^i_{k+N-1\|k,p})$
$\bar{\mathbf{u}}^i_{k,p}$	local candidate input sequence formed by $\mathcal{C}^i$
$\mathbf{v}_{k,p}$	input sequence over the prediction horizon computed by the distributed MPC with sensitivity based communication
$\mathbf{v}^i_{k,p}$	input sequence of subsystem $\mathcal{P}^i$ over the prediction horizon computed by distributed MPC with sensitivity based communication

$\bar{\mathbf{v}}^i_{k,p}$	local candidate input sequence formed by $\mathcal{C}^i$ used in distributed MPC with sensitivity based communication
x_k	state of the overall system at time k
x^i_k	state of subsystem $\mathcal{P}^i$ at time k
$\mathbf{x}_{k,p}$	state sequence over the horizon $N \geq 2$ (p is omitted if no iterations are performed): $\mathbf{x}_{k,p} := (x_{k\|k,p}; \ldots; x_{k+N\|k,p})$
$\mathbf{z}_k$	extended state vector over the horizon $N \geq 2$: $\mathbf{z}_k := (z_{k\|k}; \ldots; z_{k+N-1\|k})$
z_k	extended state vector: $z_k := (x_k; u_k)$

References

[1] A. Alessio and A. Bemporad, "Stability conditions for decentralized model predictive control under packet drop communication," *Proceedings of the American Control Conference*, pp. 3577 – 3582, 2008.

[2] I. Alvarado, D. Limon, D. Muñoz de la Peña, J. M. Maestre, M. A. Ridao, H. Scheu, W. Marquardt, R. R. Negenborn, B. D. Schutter, F. Valencia, and J. Espinosa, "A comparative analysis of distributed MPC techniques applied to the HD-MPC four-tank benchmark," *Journal of Process Control*, vol. 21, no. 5, pp. 800 – 815, 2011.

[3] R. Amrit, J. B. Rawlings, and L. T. Biegler, "Optimizing process economics online using model predictive control," *Computers & Chemical Engineering*, vol. 58, no. 0, pp. 334 – 343, 2013.

[4] N. W. Bauer, M. C. F. Donkers, N. van de Wouw, and W. P. M. H. Heemels, "Decentralized observer-based control via networked communication," *Automatica*, vol. 49, no. 7, pp. 2074 – 2086, 2013.

[5] A. Bemporad and M. Morari, "Control of systems integrating logic, dynamics, and constraints," *Automatica*, vol. 35, no. 3, pp. 407 – 427, 1999.

[6] A. Bemporad, M. Morari, V. Dua, and E. N. Pistikopoulos, "The explicit linear quadratic regulator for constrained systems," *Automatica*, vol. 38, no. 1, pp. 3 – 20, 2002.

[7] A. Ben-Tal, A. Goryashko, E. Guslitzer, and A. Nemirovski, "Adjustable robust solutions of uncertain linear programs," *Mathematical Programming*, vol. 99, pp. 351 – 376, 2004.

[8] D. Bernardini and A. Bemporad, "Energy-aware robust model predictive control based on noisy wireless sensors," *Automatica*, vol. 48, no. 1, pp. 36 – 44, 2012.

[9] D. Bernardini, M. C. F. Donkers, A. Bemporad, and W. P. M. H. Heemels, "A model predictive control approach for stochastic networked control systems," *Proceedings of the 2nd IFAC Workshop on Estimation and Control of Networked Systems*, pp. 7 – 12, 2010.

[10] D. P. Bertsekas, *Nonlinear Programming.* Athena Scientific, 1999.

[11] D. P. Bertsekas and J. N. Tsitsiklis, *Parallel and distributed computation: numerical methods.* Prentice-Hall, 1989.

[12] F. Blanchini and S. Miani, *Set Theoretic Methods in Control.* Birkhäuser, 2008.

[13] F. Borrelli, M. Baotic, A. Bemporad, and M. Morari, "Dynamic programming for constrained optimal control of discrete-time linear hybrid systems," *Automatica*, vol. 41, no. 10, pp. 1709 – 1721, 2005.

[14] R. Bourdais, H. Guéguen, and A. Belmiloudi, "Distributed model predictive control for a class of hybrid system based on lagrangian relaxation," *Proceedings of the 4th IFAC Conference on Analysis and Design of Hybrid Systems*, pp. 46 – 51, 2012.

[15] S. Boyd and L. Vandenberghe, *Convex Optimization.* Cambridge University Press, 2004.

[16] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends in Machine Learning*, vol. 3, no. 1, pp. 1 – 122, 2011.

[17] M. Bürger, G. Notarstefano, and F. Allgöwer, "A polyhedral approximation framework for convex and robust distributed optimization," *IEEE Transactions on Automatic Control*, vol. 59, no. 2, pp. 384 – 395, 2014.

[18] M. Bürger, G. Notarstefano, F. Bullo, and F. Allgöwer, "A distributed simplex algorithm for degenerate linear programs and multi-agent assignments," *Automatica*, vol. 48, no. 9, pp. 2298 – 2304, 2012.

[19] E. Camponogara, D. Jia, B. H. Krogh, and S. Talukdar, "Distributed model predictive control," *IEEE Control Systems Magazine*, vol. 22, no. 1, pp. 44 – 52, 2002.

[20] D. Christmann, R. Gotzhein, and S. Rohr, "The arbitrating value transfer protocol (AVTP) - deterministic binary countdown in wireless multi-hop networks," *Proceedings of the International Conference on Computer Communications and Networks*, pp. 1 – 9, 2012.

[21] D. Christmann, R. Gotzhein, S. Siegmund, and F. Wirth, "Realization of try-once-discard in wireless multihop networks," *IEEE Transactions on Industrial Informatics*, vol. 10, no. 1, pp. 17 – 26, 2014.

[22] M. B. G. Cloosterman, L. Hetel, N. van de Wouw, W. P. M. H. Heemels, J. Daafouz, and H. Nijmeijer, "Controller synthesis for networked control systems," *Automatica*, vol. 46, no. 10, pp. 1584 – 1594, 2010.

[23] M. D. Doan, T. Keviczky, and B. D. Schutter, “A distributed optimization-based approach for hierarchical MPC of large-scale systems with coupled dynamics and constraints,” *Proceedings of the 50th IEEE Conference on Decision and Control and European Control Conference*, pp. 5236 – 5241, 2011.

[24] J. Dold and O. Stursberg, “Distributed predictive control of communicating and platooning vehicles,” *Proceedings of the 48th IEEE Conference on Decision and Control and 28th Chinese Control Conference*, pp. 561 – 566, 2009.

[25] M. C. F. Donkers, W. P. M. H. Heemels, N. van de Wouw, and L. Hetel, “Stability analysis of networked control systems using a switched linear systems approach,” *IEEE Transactions on Automatic Control*, vol. 56, no. 9, pp. 2101 – 2115, 2011.

[26] W. B. Dunbar and R. M. Murray, “Distributed receding horizon control for multi-vehicle formation stabilization,” *Automatica*, vol. 42, pp. 549 – 558, 2006.

[27] A. Eqtami, D. V. Dimarogonas, and K. J. Kyriakopoulos, “Event-triggered strategies for decentralized model predictive controllers,” *Proceedings of the 18th IFAC World Congress*, pp. 10 068 – 10 073, 2011.

[28] A. Eqtami, D. V. Dimarogonas, and K. J. Kyriakopoulos, “Event-based model predictive control for the cooperation of distributed agents,” *Proceedings of the American Control Conference*, pp. 6473 – 6478, 2012.

[29] M. Farina, G. Betti, and R. Scattolini, “A solution to the tracking problem using distributed predictive control,” *Proceedings of the European Control Conference*, pp. 4347 – 4352, 2013.

[30] M. Farina and R. Scattolini, “Distributed predictive control: A non-cooperative algorithm with neighbor-to-neighbor communication for linear systems,” *Automatica*, vol. 48, no. 6, pp. 1088 – 1096, 2012.

[31] F. Farokhi, I. Shames, and K. H. Johansson, “Distributed MPC via dual decomposition and alternative direction method of multipliers,” ser. Intelligent Systems, Control and Automation: Science and Engineering, J. M. Maestre and R. R. Negenborn, Eds. Springer, 2014, vol. 69, pp. 115 – 131.

[32] A. Fiacco, *Introduction to Sensitivity and Stability Analysis in Nonlinear Programming.* Academic Press, 1983.

[33] E. Franco, L. Magni, T. Parisini, M. Polycarpou, and D. Raimondo, “Cooperative constrained control of distributed agents with nonlinear dynamics and delayed information exchange: A stabilizing receding-horizon approach,” *IEEE Transactions on Automatic Control*, vol. 53, no. 1, pp. 324 – 338, 2008.

[34] P. Giselsson and A. Rantzer, “On feasibility, stability and performance in distributed model predictive control,” *IEEE Transactions on Automatic Control*, vol. 59, no. 4, pp. 1031 – 1036, April 2014.

[35] P. Giselsson, “Gradient-based distributed model predictive control,” Ph.D. dissertation, Lund University, 2012.

[36] P. Giselsson, M. D. Doan, T. Keviczky, B. D. Schutter, and A. Rantzer, “Accelerated gradient methods and dual decomposition in distributed model predictive control,” *Automatica*, vol. 49, no. 3, pp. 829 – 833, 2013.

[37] P. Giselsson and A. Rantzer, “Distributed model predictive control with suboptimality and stability guarantees,” *Proceedings of the 49th IEEE Conference on Decision and Control*, pp. 7272–7277, 2010.

[38] R. Gotzhein and T. Kuhn, “Black burst synchronization (bbs) – a protocol for deterministic tick and time synchronization in wireless networks,” *Computer Networks*, vol. 55, no. 13, pp. 3015 – 3031, 2011.

[39] P. J. Goulart, E. C. Kerrigan, and J. M. Maciejowski, “Optimization over state feedback policies for robust control with constraints,” *Automatica*, vol. 42, no. 4, pp. 523 – 533, 2006.

[40] P. J. Goulart, E. C. Kerrigan, and D. Ralph, “Efficient robust optimization for robust control with constraints,” *Mathematical Programming*, vol. 114, no. 1, pp. 115 – 147, 2008.

[41] D. Groß, M. Jilg, and O. Stursberg, “Event-based communication in distributed model predictive control,” *at - Automatisierungstechnik*, vol. 61, no. 7, pp. 457 – 466, 2013.

[42] D. Groß and O. Stursberg, “Robust distributed predictive control of communicating and constrained systems,” *Proceedings of the 18th IFAC World Congress*, pp. 8926 – 8932, 2011.

[43] D. Groß and O. Stursberg, “Distributed predictive control for a class of hybrid systems with event-based communication,” *Proceedings of the 4th IFAC Workshop on Distributed Estimation and Control in Networked Systems*, pp. 383 – 388, 2013.

[44] D. Groß and O. Stursberg, “On the convergence rate of a Jacobi algorithm for cooperative distributed MPC,” *Proceedings of the 52nd IEEE Conference on Decision and Control*, pp. 1508 – 1513, 2013.

[45] D. Groß and O. Stursberg, “Optimized distributed control and network topology design for interconnected systems,” *Proceedings of the 50th IEEE Conference on Decision and Control and European Control Conference*, pp. 8112 – 8117, 2011.

[46] D. Groß and O. Stursberg, “A cooperative distributed MPC algorithm with event-based communication and parallel optimization,” *submitted*, 2014.

[47] D. Groß and O. Stursberg, “Distributed predictive control of communicating and constrained systems,” *ZAMM - Zeitschrift für Angewandte Mathematik und Mechanik*, vol. 94, no. 4, pp. 303 – 316, 2014.

[48] D. Groß and O. Stursberg, “Distributed predictive control of communicating decentralized systems,” J. Lunze, Ed. Springer, 2014, pp. 139 – 155.

[49] L. Grüne and J. Pannek, *Nonlinear Model Predictive Control: Theory and Algorithms*. Springer, 2011.

[50] L. Grüne and K. Worthmann, “A distributed NMPC scheme without stabilizing terminal constraints,” ser. Lecture Notes in Control and Information Sciences, R. Johansson and A. Rantzer, Eds. Springer, 2012, vol. 417, pp. 261 – 287.

[51] W. Hahn, *Stability of motion*, ser. Grundlehren der mathematischen Wissenschaften. Springer, 1967.

[52] W. P. M. H. Heemels, M. C. F. Donkers, and A. R. Teel, “Periodic event-triggered control for linear systems,” *IEEE Transactions on Automatic Control*, vol. 58, no. 4, pp. 847 – 861, 2013.

[53] W. P. M. H. Heemels, B. D. Schutter, and A. Bemporad, “Equivalence of hybrid dynamical models,” *Automatica*, vol. 37, no. 7, pp. 1085 – 1091, 2001.

[54] W. P. M. H. Heemels, A. R. Teel, N. V. de Wouw, and D. Nešić, “Networked control systems with communication constraints: Tradeoffs between transmission intervals, delays and performance,” *IEEE Transactions on Automatic Control*, vol. 55, no. 8, pp. 1781 – 1796, 2010.

[55] R. A. Horn and C. R. Johnson, *Matrix Analysis*, reprint ed. Cambridge University Press, 1990.

[56] J. L. Jerez, E. C. Kerrigan, and G. A. Constantinides, “A sparse and condensed qp formulation for predictive control of lti systems,” *Automatica*, vol. 48, no. 5, pp. 999 – 1002, 2012.

[57] D. Jia and B. H. Krogh, “Min-max feedback model predictive control for distributed control with communication,” *Proceedings of the American Control Conference*, pp. 4507 – 4512, 2002.

[58] Z.-P. Jiang and Y. Wang, “Input-to-state stability for discrete-time nonlinear systems,” *Automatica*, vol. 37, no. 6, pp. 857 – 869, 2001.

[59] C. N. Jones, Y. Pu, S. Riverso, G. Ferrari-Trecate, and M. N. Zeilinger, “Plug and play distributed model predictive control based on distributed invariance and optimization,” *Proceedings of the 52nd IEEE Conference on Decision and Control*, pp. 5770 – 5776, 2013.

[60] E. Kerrigan and D. Mayne, “Optimal control of constrained, piecewise affine systems with bounded disturbances,” *Proceedings of the 41st IEEE Conference on Decision and Control*, pp. 1552 – 1557, 2002.

[61] T. Keviczky, F. Borrelli, and G. J. Balas, “Decentralized receding horizon control for large scale dynamically decoupled systems,” *Automatica*, vol. 42, no. 12, pp. 2105 – 2115, 2006.

[62] C. Kirches, “Fast numerical methods for mixed–integer nonlinear model–predictive control,” Ph.D. dissertation, Heidelberg University, 2010.

[63] A. Kozma, J. Frasch, and M. Diehl, “A distributed method for convex quadratic programming problems arising in optimal control of distributed systems,” *Proceedings of the 52nd IEEE conference on Decision and Control*, pp. 1526 – 1513, 2013.

[64] A. Kozma, C. Savorgnan, and M. Diehl, “Distributed multiple shooting for large scale nonlinear systems.” Springer, 2014, pp. 327 – 340.

[65] Y. Kuwata and J. P. How, “Cooperative distributed robust trajectory optimization using receding horizon MILP,” *IEEE Transactions on Control Systems Technology*, vol. 19, no. 2, pp. 423 – 431, 2011.

[66] J. P. LaSalle, *The Stability of Dynamical Systems.* SIAM, 1976.

[67] M. Lazar and W. P. M. H. Heemels, “A new dual-mode hybrid MPC algorithm with a robust stability guarantee,” *Proceedings of the 2nd IFAC Conference on Analysis and Design of Hybrid Systems*, pp. 321 – 328, 2006.

[68] M. Lazar, D. Muñoz de la Peña, W. P. M. H. Heemels, and T. Alamo, “On input-to-state stability of min–max nonlinear model predictive control,” *Systems & Control Letters*, vol. 57, no. 1, pp. 39 – 48, 2008.

[69] T. Lehmann, “On efficient solution methods for mixed-integer nonlinear and mixed-integer quadratic optimization problems,” Ph.D. dissertation, University of Bayreuth, 2013.

[70] S. Leyffer, “Deterministic methods for mixed integer nonlinear programming,” Ph.D. dissertation, University of Dundee, 1993.

[71] F. Lin, M. Fardad, and M. R. Jovanović, “Augmented lagrangian approach to design of structured optimal state feedback gains,” *IEEE Transactions on Automatic Control*, vol. 56, no. 12, pp. 2923 – 2929, 2011.

[72] J. Liu, D. Muñoz de la Peña, and P. D. Christofides, "Distributed model predictive control of nonlinear systems subject to asynchronous and delayed measurements," *Automatica*, vol. 46, no. 1, pp. 52 – 61, 2010.

[73] J. Löfberg, "Minimax approaches to robust model predictive control," Ph.D. dissertation, Linköping University, 2003.

[74] J. Lunze, Ed., *Control Theory of Digitally Networked Dynamic Systems*. Springer, 2013.

[75] J. Lunze, *Feedback control of large scale systems*. Prentice-Hall, 1992.

[76] J. Lunze and F. Lamnabhi-Lagarrigue, *Handbook of Hybrid Systems Control: Theory, Tools, Applications*. Cambridge University Press, 2009.

[77] J. Lunze and D. Lehmann, "A state-feedback approach to event-based control," *Automatica*, vol. 46, no. 1, pp. 211 – 215, 2010.

[78] J. Löfberg, "Yalmip : A toolbox for modeling and optimization in MATLAB," *Proceedings of the CACSD Conference*, 2004.

[79] J. M. Maestre, D. Muñoz de la Peña, and E. F. Camacho, "Distributed model predictive control based on a cooperative game," *Optimal Control Applications and Methods*, vol. 32, no. 2, pp. 153 – 176, 2011.

[80] L. Magni and R. Scattolini, "Stabilizing decentralized model predictive control of nonlinear systems," *Automatica*, vol. 42, no. 7, pp. 1231 – 1236, 2006.

[81] D. Q. Mayne, J. B. Rawlings, C. V. Rao, and P. O. M. Scokaert, "Constrained model predictive control: Stability and optimality," *Automatica*, vol. 36, pp. 789 – 814, 2000.

[82] M. Meinel, M. Ulbrich, and S. Albrecht, "A class of distributed optimization methods with event-triggered communication," *Computational Optimization and Applications*, pp. 1 – 37, 2013.

[83] M. A. Müller, M. Reble, and F. Allgöwer, "Cooperative control of dynamically decoupled systems via distributed model predictive control," *International Journal of Robust and Nonlinear Control*, vol. 22, no. 12, pp. 1376 – 1397, 2012.

[84] J. R. Munkres, *Analysis on Manifolds*. Addison-Wesley, 1991.

[85] I. Necoara, D. N. Clipici, and S. Olaru, "Distributed model predictive control of leader-follower systems using an interior point method with efficient computations," *Proceedings of the American Control Conference*, pp. 1697 – 1702, 2013.

[86] J. Pannek, "Parallelizing a state exchange strategy for noncooperative distributed NMPC," *Systems & Control Letters*, vol. 62, no. 1, pp. 29 – 36, 2013.

[87] G. Pannocchia, J. B. Rawlings, and S. J. Wright, "Conditions under which suboptimal nonlinear MPC is inherently robust," *Systems & Control Letters*, vol. 60, no. 9, pp. 747 – 755, 2011.

[88] D. W. Peterson, "A review of constraint qualifications in finite-dimensional spaces," *SIAM Review*, vol. 15, no. 3, pp. 639 – 654, 1973.

[89] R. Postoyan, A. Anta, D. Nešić, and P. Tabuada, "A unifying lyapunov-based framework for the event-triggered control of nonlinear systems," *Proceedings of the 50th IEEE Conference on Decision and Control and European Control Conference*, pp. 2559 – 2564, 2011.

[90] J. B. Rawlings and D. Q. Mayne, *Model Predictive Control: Theory and Design.* Nob Hill Publishing, 2009.

[91] A. Richards and J. P. How, "Robust distributed model predictive control," *International Journal of Control*, vol. 80, no. 9, pp. 1517 – 1531, 2007.

[92] S. Riverso, M. Farina, and G. Ferrari-Trecate, "Plug-and-play decentralized model predictive control for linear systems," *IEEE Transactions on Automatic Control*, vol. 58, no. 10, pp. 2608 – 2614, 2013.

[93] R. T. Rockafellar and R. J. B. Wets, *Variational Analysis*, ser. Die Grundlehren der mathematischen Wissenschaften. Springer, 1998.

[94] R. Scattolini, "Architectures for distributed and hierarchical model predictive control – a review," *Journal of Process Control*, vol. 19, no. 5, pp. 723 – 731, 2009.

[95] H. Scheu and W. Marquardt, "Sensitivity-based coordination in distributed model predictive control," *Journal of Process Control*, vol. 21, no. 5, pp. 715 – 728, 2011.

[96] T. Schouwenaars, "Safe trajectory planning of autonomous vehicles," Ph.D. dissertation, Massachusetts Institute of Technology, 2006.

[97] A. Schrijver, *Theory of Linear and Integer Programming*, ser. Wiley Series in Discrete Mathematics & Optimization. John Wiley & Sons, 1998.

[98] S. Schuler, P. Li, J. Lam, and F. Allgöwer, "Design of structured dynamic output-feedback controllers for interconnected systems," *International Journal of Control*, vol. 84, no. 12, pp. 2081 – 2091, 2011.

[99] D. D. Šiljak, *Large-scale dynamic systems: stability and structure.* North-Holland, 1978.

[100] J. Skaf and S. Boyd, “Design of affine controllers via convex optimization,” *IEEE Transactions on Automatic Control*, vol. 55, no. 11, pp. 2476 – 2487, 2010.

[101] E. D. Sontag, “Smooth stabilization implies coprime factorization,” *IEEE Transactions on Automatic Control*, vol. 34, no. 4, pp. 435 – 443, 1989.

[102] B. T. Stewart, “Plantwide cooperative distributed model predictive control,” Ph.D. dissertation, University of Wisconsin-Madison, 2010.

[103] B. T. Stewart, J. B. Rawlings, and S. J. Wright, “Hierarchical cooperative distributed model predictive control,” *Proceedings of the American Control Conference*, pp. 3963 – 3968, 2010.

[104] B. T. Stewart, A. N. Venkat, J. B. Rawlings, S. J. Wright, and G. Pannocchia, “Cooperative distributed model predictive control,” *Systems & Control Letters*, vol. 59, no. 8, pp. 460 – 469, 2010.

[105] C. Stöcker and J. Lunze, “Distributed event-based control of physically interconnected systems,” *Proceedings of the 52nd IEEE Conference on Decision and Control*, pp. 7376 – 7383, 2013.

[106] C. Stöcker, D. Vey, and J. Lunze, “Decentralized event-based control: Stability analysis and experimental evaluation,” *Nonlinear Analysis: Hybrid Systems*, vol. 10, no. 0, pp. 141 – 155, 2013.

[107] P. Trnka, J. Pekar, and V. Havlena, “Application of distributed MPC to barcelona water distribution network,” *Proceedings of the 18th IFAC World Congress*, pp. 9139 – 9144, 2011.

[108] P. Trodden and A. Richards, “Distributed model predictive control of linear systems with persistent disturbances,” *International Journal of Control*, vol. 83, no. 8, pp. 1653 – 1663, 2010.

[109] P. Trodden and A. Richards, “Cooperative distributed MPC of linear systems with coupled constraints,” *Automatica*, vol. 49, no. 2, pp. 479 – 487, 2013.

[110] P. Varutti, B. Kern, and R. Findeisen, “Dissipativity-based distributed nonlinear predictive control for cascaded systems,” *Proceedings of the 8th IFAC International Symposium on Advanced Control of Chemical Processes*, pp. 439 – 444, 2012.

[111] A. N. Venkat, “Distributed model predictive control: Theory and applications,” Ph.D. dissertation, University of Wisconsin-Madison, 2006.

[112] A. N. Venkat, I. A. Hiskens, J. B. Rawlings, and S. J. Wright, "Distributed MPC strategies with application to power system automatic generation control," *IEEE Transactions on Control Systems Technology*, vol. 16, no. 6, pp. 1192 – 1206, 2008.

[113] G. Walsh, H. Ye, and L. Bushnell, "Stability analysis of networked control systems," *IEEE Transactions on Control Systems Technology*, vol. 10, no. 3, pp. 438 – 446, 2002.

[114] H. P. Williams, *Model building in mathematical programming.* John Wiley & Sons, 1999.

[115] Y. Y. Wakasa, M. Arakawa, K. Tanaka, and T. Akashi, "Decentralized model predictive control via dual decomposition," *Proceedings of the 47th IEEE Conference on Decision and Control*, pp. 381 – 386, 2008.

ISBN 978-3-86219-910-5