Parallelized Robust Distributed Model Predictive Control in the Presence of Coupled State Constraints *

Adrian Wiltz, Fei Chen, Dimos V. Dimarogonas

Division of Decision and Control Systems, KTH Royal Institute of Technology, Stockholm, Sweden

Abstract

In this paper, we present a robust distributed model predictive control (DMPC) scheme for dynamically decoupled nonlinear systems which are subject to state constraints, coupled state constraints and input constraints. In the proposed control scheme, all subsystems solve their local optimization problem in parallel and neighbor-to-neighbor communication suffices. The approach relies on consistency constraints which define a neighborhood around each subsystem's reference trajectory where the state of the subsystem is guaranteed to stay in. Contrary to related approaches, the reference trajectories are improved consecutively. In order to ensure the controller's robustness against bounded uncertainties, we employ tubes. The presented approach can be considered as a time-efficient alternative to the well-established sequential DMPC. In the end, we briefly comment on an iterative extension. The effectiveness of the proposed DMPC scheme is demonstrated with simulations, and its performance is compared to other DMPC schemes.

Key words: Networked control systems, Model predictive control, Control of constrained systems, Multi-agent systems, Robust model predictive control

1 INTRODUCTION

Model predictive control (MPC) algorithms are successfully employed in a wide range of applications and there exists a broad body of literature. The fact that hard constraints on states and inputs can be directly incorporated into the controller and a performance criterion can be taken into account by means of solving an optimization problem significantly contributes to their popularity. A basic approach for proving recursive feasibility and convergence of the optimization problems has been developed in [1] for continuous-time nonlinear systems, and is the basis for numerous MPC algorithms proposed since then. Similar results are derived for discrete-time systems in [2], and an overview can be found in [3–5].

1.1 Distributed MPC

Since the presentation of initial distributed model predictive control (DMPC) schemes [6], their development became a thriving branch in the research on MPC. The motivation behind the development of DMPC is that centralized MPC [4] becomes computationally intractable for large-scale systems, and a reliable communication with a central control-unit is difficult to realize in the case of spatially distributed systems [7, 8].

In [9], the methods by which distributed MPC algorithms compute control input trajectories are classified into four groups: iterative methods, sequential methods, methods employing consistency constraints, and approaches based on robustness considerations. In iterative methods, the local controllers exchange the solutions to their local optimization problems several times among each other until they converge. In sequential approaches, local optimization problems of neighboring subsystems are not evaluated in parallel but one after another. In algorithms based on consistency constraints, neighboring subsystems exchange reference trajectories and guarantee to stay in their neighborhood. Other DMPC algorithms consider the neighbors' control decisions as a disturbance. Examples can be found in [9]. As remarked in [10], the task of distributing MPC algorithms is too complex in order to solve it with one single approach. Instead, for various types of centralized MPC problems, distributed controllers have been taylored. A broad collection of notable DMPC algorithms can be found in [11].

Especially the distribution of MPC problems subject to

* This work was supported by the ERC Consolidator Grant LEAFHOUND, the Swedish Research Council (VR) and the Knut och Alice Wallenberg Foundation (KAW).

Email addresses: wiltz@kth.se (Adrian Wiltz), fchen@kth.se (Fei Chen), dimos@kth.se (Dimos V. Dimarogonas).

coupled state constraints turned out to be complicated [12], and most available DMPC schemes that are capable of handling them cannot avoid a sequential scheme [13–16]. However, sequential schemes have the drawback that the computation of the control input for all subsystems becomes very time-consuming for highly connected networks. A notable exception that does not rely on a sequential scheme can be found in [17], where a consistency constraint approach is used instead. This admits that even in the presence of coupled state constraints all subsystems can solve their local optimization problem in parallel and still retain recursive feasibility. In [18], this approach is transferred to a continuous-time setup, and in [19–21], it is extended to a so-called plug-and-play MPC algorithm. However, [17,18] employ fixed reference trajectories, i.e., reference trajectories that may not be modified once defined. This is limiting and restricts the possibility to optimize the system's performance significantly. In this paper, we overcome this limitation for dynamically decoupled systems subject to coupled state constraints.

1.2 Contributions

We develop a generic DMPC scheme based on consistency constraints that allows for the parallelized evaluation of the local optimization problems in the presence of coupled state constraints. In contrast to the consistency constraint based approaches [17,18] with fixed reference trajectories, we show that in the case of dynamically decoupled systems this limitation can be overcome. In particular, we show that recursive feasibility and asymptotic stability can be still obtained when invoking a less restrictive assumption on reference trajectories and consistency constraints. It allows the reference trajectories and consistency constraints to be updated at every timestep. As nominal MPC subject to constraints is sensitive to model uncertainties and disturbances, we formulate the proposed DMPC scheme via a constraint-tightening approach using robust tubes.

The proposed DMPC scheme can be considered as a time-efficient and scalable alternative to sequential DMPC [15, 22], and a more performant and less restrictive version of the consistency constraint based DMPC in [17, 18] for dynamically decoupled systems. The preliminary version of the proposed DMPC scheme as presented in [23] is restricted to nominal dynamics and the pairwise coupling of states through constraints which are limitations that we overcome in this paper.

The remainder is structured as follows. In Sec. 2, we present the partitioned system and the control objective. In Sec. 3, we define the local optimization problems (Sec. 3.1), present assumptions that allow for their parallelized evaluation and ensure robust asymptotic stability (Sec. 3.2), and derive guarantees of the closed-loop system (Sec. 3.3). A brief discussion concludes the section. In Sec. 4, we provide details on the initialization, the computation of reference trajectories and summarize the overall DMPC algorithm. In Sec. 5, the algorithm's effectiveness is demonstrated, and in Sec. 6 some conclusions are drawn.

1.3 Notation

A continuous function $\gamma : \mathbb{R}_+ \to \mathbb{R}_+$ is a class $\mathcal{K}$ function if it is strictly increasing and $\gamma(0) = 0$. If the domain of a trajectory $x[\kappa]$ with $\kappa = a,\dots,b$, $a, b \in \mathbb{N}$, is clear from the context, we also write $x[\cdot]$. By $x[\cdot|k]$, we denote a trajectory that is computed at time step k. The shorthand $x_{i\in\mathcal{V}}$ is equivalent to $\{x_i\}_{i\in\mathcal{V}}$ where $\mathcal{V}$ is a set of indices. Let $\mathcal{A}, \mathcal{B} \subset \mathbb{R}^n$. Their Minkowski sum is $\mathcal{A} \oplus \mathcal{B} := \{a + b \in \mathbb{R}^n \mid a \in \mathcal{A},\ b \in \mathcal{B}\}$, the Pontryagin difference $\mathcal{A} \ominus \mathcal{B} := \{a \in \mathbb{R}^n \mid a + b \in \mathcal{A},\ \forall b \in \mathcal{B}\}$, and $\mathcal{A} \times \mathcal{B}$ their Cartesian product. $\bigtimes_i$, $\bigoplus_i$ denote the repeated evaluation of the respective operations over sets of indices. The Hausdorff distance d_H is defined as the minimal distance of a point $x \in \mathbb{R}^n$ to a set $\mathcal{A}$, i.e., $d_H(x, \mathcal{A}) := \inf\{||x - x'|| \mid x' \in \mathcal{A}\}$; $||\cdot||$ denotes the Euclidean norm. Let $\boldsymbol{A} \in \mathbb{R}^{m\times n}$ be a matrix, and $\alpha \in \mathbb{R}$ a scalar. Then, $\alpha\mathcal{A} := \{y \mid y = \alpha x, x \in \mathcal{A}\}$ and $\boldsymbol{A}\mathcal{A} := \{y \mid y = \boldsymbol{A}x, x \in \mathcal{A}\}$. Finally, $\boldsymbol{0}, \boldsymbol{1}$ denote vectors of all zeros or ones, $\boldsymbol{I}$ an identity matrix, and for $x \in \mathbb{R}^n$ and $\boldsymbol{P} \in \mathbb{R}^{n\times n}$ positive-definite, we define the weighted norm $||x||_{\boldsymbol{P}} := \sqrt{x^T\boldsymbol{P}x}$.

2 Preliminaries

In this section, we introduce the system dynamics of a partitioned system and its constraints, define the network topology, review robustness related concepts, and finally state the control objective.

2.1 System Dynamics and Constraints

Consider a distributed system consisting of subsystems $i \in \mathcal{V} = \{1, \dots, |\mathcal{V}|\}$ which are dynamically decoupled and behave according to the discrete-time dynamics

$$x_i[k + 1] = f_i(x_i[k], u_i[k], w_i[k]), \qquad x_i[0] = x_{0,i} \quad (1)$$

where $x_i \in \mathbb{R}^{n_i}$, $u_i \in \mathbb{R}^{m_i}$ denote the actual state and input of subsystem i, respectively, and $w_i \in \mathcal{W}_i \subset \mathbb{R}^{q_i}$ a bounded uncertainty. In stack-vector form, the dynamics of the overall system are correspondingly given by

$$x[k + 1] = f(x[k], u[k], w[k]), \qquad x[0] = x_0 \quad (2)$$

where $f : \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^q \to \mathbb{R}^n$ with $n := \sum_i n_i$, $m := \sum_i m_i$, $q := \sum_i q_i$, stack vectors $x = [x_1^T, \dots, x_{|\mathcal{V}|}^T]^T$, $u = [u_1^T, \dots, u_{|\mathcal{V}|}^T]^T$, $w = [w_1^T, \dots, w_{|\mathcal{V}|}^T]^T$, and $x_0 = [x_{0,1}^T, \dots, x_{0,|\mathcal{V}|}^T]^T$.

All subsystems may be subject to coupled and non-coupled state constraints. If there exists a coupled state constraint of subsystem i that depends on state x_j of subsystem j, then we call subsystem j a *neighbor* of subsystem i and we write $j \in \mathcal{N}_i$ where $\mathcal{N}_i$ is the set of all neighboring subsystems of i. Let subsystem i be subject to R_i coupled state constraints. Then, we define the non-coupled and coupled state constraints of subsystem i, respectively, via inequalities as

$$h_i(x_i) \leqslant \mathbf{0} \tag{3a}$$
$$c_{i,r}(x_i, x_{j\in\mathcal{N}_{i,r}}) \leqslant \mathbf{0}, \qquad r = 1, \ldots, R_i \tag{3b}$$

where $h_i : \mathbb{R}^{n_i} \to \mathbb{R}^{s_i}$ and $c_{i,r} : \times_{j\in\{i\}\cup\mathcal{N}_{i,r}} \mathbb{R}^{n_j} \to \mathbb{R}^{s_{i,r}}$ are some continuous functions, and $\mathcal{N}_{i,r} \subseteq \mathcal{N}_i$ specifies those neighbors of subsystem i whose states are coupled with subsystem i via coupled state constraint r; $s_i, s_{i,r} \in \mathbb{N}_{>0}$ are some constants. The corresponding state constraint sets are defined as

$$\mathcal{X}_i := \{x_i \in \mathbb{R}^{n_i} \mid h_i(x_i) \leqslant \mathbf{0}\}$$
$$\mathcal{X}_{i,r}(x_{j\in\mathcal{N}_{i,r}}) := \{x_i \in \mathbb{R}^{n_i} | c_{i,r}(x_i, x_{j\in\mathcal{N}_{i,r}}) \leqslant \mathbf{0}\}$$

where $\mathcal{X}_{i,r}$ is a set valued function. Moreover, all subsystems $i \in \mathcal{V}$ are subject to input constraints

$$u_i \in \mathcal{U}_i \subseteq \mathbb{R}^{m_i}. \tag{4}$$

From the actual subsystem dynamics (1), we distinguish the *nominal dynamics* of the undisturbed system which are given by

$$\hat{x}_i[k+1] = \hat{f}_i(\hat{x}_i[k], \hat{u}_i[k]) := f_i(\hat{x}_i[k], \hat{u}_i[k], 0), \quad \hat{x}_i[0] = x_{0,i} \tag{5}$$

where $\hat{x}_i \in \mathbb{R}^{n_i}$, $\hat{u}_i \in \mathbb{R}^{m_i}$ denote the nominal state and nominal input of subsystem i, respectively. The nominal dynamics of the overall system are denoted by

$$\hat{x}[k+1] = \hat{f}(\hat{x}[k], \hat{u}[k]) := f(\hat{x}[k], \hat{u}[k], 0). \tag{6}$$

2.2 Network Topology

The coupled state constraints define a graph structure on the distributed system under consideration. The graph is given as $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ where $\mathcal{E} := \{(i,j) \,|\, j \in \mathcal{N}_i\}$ defines the communication links among the subsystems in $\mathcal{V}$. Hence neighboring subsystems can communicate with each other. Throughout the paper, we assume that graph $\mathcal{G}$ is undirected. [1] Moreover, to avoid that subsystems can behave in an adversarial way and "force" neighboring subsystems to infeasible states, we assume that neighboring subsystems have those constraints that couple their states in common. This is formally stated next.

Assumption 1 *Let $c_{i,r}(x_i, x_{j\in\mathcal{N}_{i,r}})$, $r \in \{1, \ldots, R_i\}$, be any coupled state constraint of a subsystem $i \in \mathcal{V}$. Then all neighbors $j \in \mathcal{N}_{i,r} \subseteq \mathcal{N}_i$ are subject to a constraint*

$$c_{j,r'}(x_j, x_{j'\in\mathcal{N}_{j,r'}}) \equiv c_{i,r}(x_i, x_{i'\in\mathcal{N}_{i,r}})$$

for some $r' \in \{1, \ldots, R_j\}$ where $\mathcal{N}_{j,r'} := (\mathcal{N}_{i,r} \backslash \{j\}) \cup \{i\}$ and where $\equiv$ denotes the equivalence of the functions.

[1] A graph $\mathcal{G}$ is undirected if $(i,j) \in \mathcal{E}$ implies $(j,i) \in \mathcal{E}$.

2.3 Robust Stability and Tubes

In order to handle uncertainties w_i in the dynamics (1), we resort to a tube-based approach [24, 25]. Here, an auxiliary controller is employed to bound the deviation of the system's actual state to the predicted state into an invariant set. To this end, we formally define the deviation of an actual state x_i and nominal state $\hat{x}_i$ as $p_i := x_i - \hat{x}_i$, and the corresponding dynamics of p_i are

$$p_i[k+1] = f_i(x_i[k], u_i[k], w_i[k]) - \hat{f}_i(\hat{x}_i[k], \hat{u}_i[k]) \tag{7}$$

with $p_i[0] = 0$, $x_i[0] = \hat{x}_i[0] = x_{0,i}$. Considering a control signal

$$u_i[k] = \hat{u}_i[k] + K_i(x_i[k], \hat{x}_i[k]) \tag{8}$$

consisting of a nominal input $\hat{u}_i$ and an auxiliary controller $K_i : \mathcal{X}_i \times \mathcal{X}_i \to \mathcal{U}_i$, we assume that for (7) controlled by (8) there exists a robust positively invariant (RPI) set $\mathcal{P}_i$.

Assumption 2 *Let the dynamics of p_i in (7) be controlled by (8). For all $i \in \mathcal{V}$, assume that there exists a neighborhood of the origin $\mathcal{P}_i \subseteq \mathbb{R}^{n_i}$ such that $p_i[k+1] \in \mathcal{P}_i$ for all $p_i[k] \in \mathcal{P}_i$, $w_i[k] \in \mathcal{W}_i$, and $\hat{u}_i[k] \in \mathcal{U}_i$. Such $\mathcal{P}_i$ is called an RPI set.*

Remark 3 *For the construction of RPI sets $\mathcal{P}_i$, we refer to the rich literature on robust tube-based MPC, see [25–32]. For a general nonlinear system, the dynamics of p_i in (7) cannot be written as a function $\mathfrak{f}(p_i[k], w_i[k])$, i.e., $p_i[k+1] = f_i(x_i[k], u_i[k], w_i[k]) - \hat{f}_i(\hat{x}_i[k], \hat{u}_i[k]) \neq \mathfrak{f}(p_i[k], w_i[k])$. Therefore, it is not always straightforward to find RPI sets $\mathcal{P}_i$. Instead, most works on the construction of RPI sets focus on particular classes of systems. In [25], linear systems of the form $x[k+1] = \boldsymbol{A}x[k] + \boldsymbol{B}u[k]$ are considered where $\boldsymbol{A}, \boldsymbol{B}$ are real matrices of respective sizes and the pair $(\boldsymbol{A}, \boldsymbol{B})$ is assumed to be controllable. In [27], this is extended to systems with matched nonlinearities of the form*

$$x[k+1] = \boldsymbol{A}x[k] + \boldsymbol{B}(g(x[k])u[k] + \varphi(x[k])) + w[k]$$

where g is assumed to be invertible for all $x \in \mathcal{X}$. For both dynamics, auxiliary controllers can be found such that the dynamics of p_i take the form $p[k+1] = \mathfrak{f}(p_i[k], w_i[k])$. In contrast, the construction in [29] does not require that the dynamics of p_i take this form. Other recent approaches improve uncertainty bounds online [31], or employ the high-gain idea from funnel control [32] (the latter however is confined to continuous-time systems). Our focus in this paper is on integrating $\mathcal{P}_i$ into the proposed consistency constraint based DMPC scheme irrespective of the particular robust MPC method.

As a consequence of Ass. 2, the actual state $x_i[k]$ stays in a neighborhood $\mathcal{P}_i$ of the nominal state $\hat{x}_i[k]$ for all times k since $p_i[k] = x_i[k] - \hat{x}_i[k] \in \mathcal{P}_i$, or equivalently

$$x_i[k] \in \hat{x}_i[k] \oplus \mathcal{P}_i, \quad \forall k \geqslant 0. \tag{9}$$

Then, we can determine the set of all inputs that K_i possibly takes and define it as

$$\Delta\mathcal{U}_i := \{K_i(x_i, \hat{x}_i) \in \mathcal{U}_i \mid x_i, \hat{x}_i \in \mathcal{X}_i,\ x_i - \hat{x}_i \in \mathcal{P}_i\}. \tag{10}$$

The resulting tightened constraint sets are given as $\widehat{\mathcal{X}}_i := \mathcal{X}_i \ominus \mathcal{P}_i$, $\widehat{\mathcal{X}}_{i,r}(\hat{x}_{j\in\mathcal{N}_{i,r}}) := \{\hat{x}_i | c_{i,r}(x_i, x_{j\in\mathcal{N}_{i,r}}) \leqslant 0,\ \forall x_i \in \hat{x}_i \oplus \mathcal{P}_i,\ \forall x_j \in \hat{x}_j \oplus \mathcal{P}_j,\ j \in \mathcal{N}_{i,r}\}$, and $\widehat{\mathcal{U}}_i := \mathcal{U}_i \ominus \Delta\mathcal{U}_i$.

By suitably choosing a nominal control input $\hat{u}_i$, we aim at guaranteeing the robust asymptotic stabilization of the overall uncertain dynamics (2).

Definition 4 (Robust Asymptotic Stability) *Let $\mathcal{R}$ be an RPI set for the autonomous discrete-time system $x[k+1] = g(x[k], w[k])$ with an equilibrium point $\xi = g(\xi, 0) \in \mathcal{R}$ and $x \in \mathbb{R}^n$, $w \in \mathcal{W} \subseteq \mathbb{R}^q$. The equilibrium ξ is said to be* robustly stable *if for each $\varepsilon > 0$, there exists a $\delta = \delta(\varepsilon) > 0$ such that $d_H(x[0], \mathcal{R}) < \delta \Rightarrow d_H(x[k], \mathcal{R}) < \varepsilon$ for all $k > 0$ and arbitrary disturbances $w \in \mathcal{W}$; d_H denotes the Hausdorff distance. The equilibrium ξ is* robustly asymptotically stable *if ξ is robustly stable and $d_H(x[0], \mathcal{R}) < \delta \Rightarrow \lim_{k\to\infty} d_H(x[k], \mathcal{R}) = 0$.*

At last, the following lemma provides useful relations for set-operations, especially in the context of uncertainties.

Lemma 5 ([33], Lem. 3.1.11) *Let $\mathcal{A}, \mathcal{B} \subset \mathbb{R}^n$. Then $(\mathcal{A} \oplus \mathcal{B}) \ominus \mathcal{B} \supseteq \mathcal{A}$, and $(\mathcal{A} \ominus \mathcal{B}) \oplus \mathcal{B} \subseteq \mathcal{A}$.*

2.4 Control Objective

Let $\xi_i = \hat{f}_i(\xi_i, u_{\xi_i})$ be a steady state of the nominal dynamics (5) of subsystem i for a constant nominal input u_{ξ_i}. Moreover, denote the stack vector of all steady states by $\xi = [\xi_1^T, \ldots, \xi_{|\mathcal{V}|}^T]^T$. For all subsystems $i \in \mathcal{V}$, the control objective is to robustly asymptotically stabilize desired states $\xi_{i\in\mathcal{V}}$ where

$$\begin{aligned}\xi \in \Xi := \{\xi \mid \xi_i &= \hat{f}_i(\xi_i, u_{\xi_i}),\\ &\xi_i \in \widehat{\mathcal{X}}_i,\ \xi_i \in \widehat{\mathcal{X}}_{i,r}(\xi_{j\in\mathcal{N}_{i,r}}),\ u_{\xi_i} \in \widehat{\mathcal{U}}_i,\\ &\forall r = 1, \ldots, R_i,\ i \in \mathcal{V}\}\end{aligned} \tag{11}$$

satisfies all nominal state constraints (coupled and uncoupled) and nominal input constraints. Note that the tightened constraint sets $\widehat{\mathcal{X}}_i, \widehat{\mathcal{X}}_{i,r}, \widehat{\mathcal{U}}_i$ as previously defined in Sec. 2.3 ensure that none of the actual constraints (3) and (4) are violated in the desired formation due to the uncertainties.

3 Distributed MPC Problems

For each subsystem, we now formulate local optimization problems that can be solved in parallel while guaranteeing the satisfaction of all constraints, most notably that of coupled state constraints. In this section, we provide the theoretic foundations of our approach. Implementation details and the overall DMPC algorithm are presented in the subsequent Sec. 4.

3.1 Local Optimization Problems

At every time-step k, a subsystem i predicts a nominal state trajectory $\hat{x}_i[\kappa|k]$ for $\kappa = k, \ldots, k+N$ and a corresponding nominal input trajectory $\hat{u}_i[\kappa|k]$ for $\kappa = k, \ldots, k+N-1$ in accordance with the nominal dynamics (5). We call N the prediction horizon. As the actual state x_i is uncertain, and it is only known that $x_i \in \hat{x}_i \oplus \mathcal{P}_i$ according to (9), the nominal state trajectories $\hat{x}_{i\in\mathcal{V}}[\cdot|k]$ are determined such that the initial state satisfies

$$\hat{x}_i[k|k] \in x_i[k] \oplus (-\mathcal{P}_i),\ i \in \mathcal{V} \tag{12}$$

where $-\mathcal{P}_i = -1\mathcal{P}_i$ (cf. notation in Sec. 1.3). In order to allow for parallelization, we introduce the notion of *consistency constraints.* A consistency constraint guarantees that the nominal state trajectory $\hat{x}_i$ stays in a $\widehat{\mathcal{C}}_i$-neighborhood of a given reference trajectory x_i^{ref}, namely

$$\hat{x}_i[\kappa|k] \in x_i^{\mathrm{ref}}[\kappa|k] \oplus \widehat{\mathcal{C}}_i, \quad \forall \kappa = k, \ldots, k+N \tag{13}$$

where $\widehat{\mathcal{C}}_i \subseteq \mathbb{R}^{n_i}$ is some compact neighborhood of the origin which we call *consistency constraint set*, and $x_i^{\mathrm{ref}}[\cdot|k]$ a given reference trajectory at time k. Both x_i^{ref} and $\widehat{\mathcal{C}}_i$ are assumed to be known to all neighbors $j \in \mathcal{N}_i$ of subsystem i, and are further specified later. While a consistency constraint allows to optimize the nominal state trajectory in a neighborhood of a reference trajectory, it makes the nominal state trajectory predictable to neighbors. Thereby, consistency constraints give rise to parallelization.

Remark 6 *Throughout the paper, we employ reference trajectories as an inherent ingredient to consistency constraint based DMPC which is aligned with the terminology used in [17, 18]. This should not be mixed up with reference trajectories in the tracking MPC literature.*

If Ass. 2 and thereby (9) hold, then consistency constraint (13) also implies

$$x_i[k] \in x_i^{\mathrm{ref}}[k|k] \oplus \mathcal{C}_i \quad \text{with} \quad \mathcal{C}_i := \widehat{\mathcal{C}}_i \oplus \mathcal{P}_i. \tag{14}$$

Figure 1 illustrates the concept of consistency constraints.

Given $x_i[k]$ and $x_i^{\mathrm{ref}}[\cdot|k]$, we define the local optimization problem of subsystem i at time-step k as

$$J_i^*(x_i[k]) = \min_{\hat{x}_i[k|k], \hat{u}_i[\cdot|k]} J_i(\hat{x}_i[\cdot|k], \hat{u}_i[\cdot|k]) \tag{15}$$

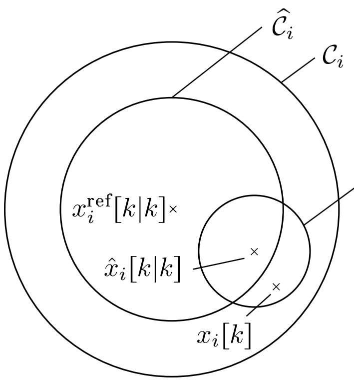


Fig. 1. Illustration of consistency constraints (13) (at $\kappa = k$) and (14).

where

$$J_i(\hat{x}_i[\cdot|k], \hat{u}_i[\cdot|k]) = \sum_{\kappa=k}^{k+N-1} l_i(\hat{x}_i[\kappa|k], \hat{u}_i[\kappa|k]) + J_i^f(\hat{x}_i[k+N|k])$$

$$l_i(\hat{x}_i[\kappa|k], \hat{u}_i[\kappa|k]) = ||\hat{x}_i[\kappa|k] - \xi_i||^2_{\boldsymbol{Q}_i} + ||\hat{u}_i[\kappa|k] - u_{\xi_i}||^2_{\boldsymbol{R}_i}$$

with stage-cost-function l_i, a positive-definite terminal cost-function J_i^f, and positive-definite matrices $\boldsymbol{Q}_i, \boldsymbol{R}_i$. The local optimality criterion (15) is subject to

$$\hat{x}_i[k|k] \in x_i[k] \oplus (-\mathcal{P}_i) \tag{16a}$$
$$\hat{x}_i[\kappa+1|k] = \hat{f}_i(\hat{x}_i[\kappa|k], \hat{u}_i[\kappa|k]) \tag{16b}$$
$$\hat{x}_i[\kappa|k] \in x_i^{\mathrm{ref}}[\kappa|k] \oplus \hat{\mathcal{C}}_i \tag{16c}$$
$$\hat{u}_i[\kappa|k] \in \hat{\mathcal{U}}_i \tag{16d}$$
$$\hat{x}_i[k+N|k] \in \hat{\mathcal{X}}_i^f \tag{16e}$$

for $\kappa = k, \ldots, k+N-1$ and some terminal set $\hat{\mathcal{X}}_i^f$ to be further specified below. We denote those trajectories $\hat{x}_i$, $\hat{u}_i$ that minimize (15) by $\hat{x}_i^*[\cdot|k]$ and $\hat{u}_i^*[\cdot|k]$. Then, the control law for the i-th subsystem (1), for all $k \geqslant 0$, is given by

$$u_i^{\mathrm{MPC}}(x_i[k]) := \hat{u}_i^{\mathrm{MPC}}(x_i[k]) + K_i(x_i[k], \hat{x}_i^*[k|k]) \tag{17}$$

where $\hat{u}_i^{\mathrm{MPC}}(x_i[k]) := \hat{u}_i^*[k|k]$ and K_i is the auxiliary controller rendering $\mathcal{P}_i$ an RPI set (cf. Ass. 2).

We impose the following slightly modified standard assumptions [4] on terminal sets $\hat{\mathcal{X}}_i^f$ and terminal cost J_i^f.

Assumption 7 (Terminal Constraints) *For the terminal sets $\hat{\mathcal{X}}_i^f$, the terminal cost function $J_i^f : \hat{\mathcal{X}}_i^f \to \mathbb{R}_{\geqslant 0}$, and some state-feedback controller $k_i^f(x_i)$, $i \in \mathcal{V}$, we assume:*

A7.1 for all $\hat{\mathcal{X}}_i^f$, $i \in \mathcal{V}$,
- *(i) $\hat{\mathcal{X}}_i^f \subseteq \mathcal{X}_i \ominus \mathcal{C}_i$, and*
- *(ii) for all $r = 1, \ldots, R_i$, it holds $c_{i,r}(x_i, x_{j \in \mathcal{N}_{i,r}}) \leqslant \mathbf{0}$ for all $x_i \in \hat{\mathcal{X}}_i^f \oplus \mathcal{C}_i$ and for all $x_j \in \hat{\mathcal{X}}_j^f \oplus \mathcal{C}_j$, $j \in \mathcal{N}_{i,r}$*

(terminal states satisfy uncoupled (i) and coupled state constraints (ii) in a $\mathcal{C}_i$-neighborhood);

A7.2 $k_i^f(x_i) \in \hat{\mathcal{U}}_i$ for all $x_i \in \hat{\mathcal{X}}_i^f$ (input constraint satisfaction);

A7.3 $\hat{f}_i(x_i, k_i^f(x_i)) \in \hat{\mathcal{X}}_i^f$ for all $x_i \in \hat{\mathcal{X}}_i^f$ (set invariance);

A7.4 $J_i^f(\hat{f}(x_i, k_i^f(x_i))) + l_i(x_i, k_i^f(x_i)) \leqslant J_i^f(x_i)$ for all $x_i \in \hat{\mathcal{X}}_i^f$ (terminal cost function is a local Lyapunov function).

3.2 Reference Trajectories and Consistency Constraints

The role of the consistency constraints is twofold: on the one hand, consistency constraints shall ensure the satisfaction of all state constraints (both coupled and uncoupled) by bounding the predicted nominal state trajectory $\hat{x}_i$ in the neighborhood of its reference trajectory x_i^{ref}. On the other hand, consistency constraints shall only restrict the evolution of nominal state trajectories in so far such that the feasibility of the local optimization problem (15)-(16) is preserved. To this end, we assume the following properties of reference trajectories x_i^{ref}, $i \in \mathcal{V}$, for given consistency constraint sets $\mathcal{C}_i$, $i \in \mathcal{V}$.

Assumption 8 (Reference Trajectories) *For $\kappa = k, \ldots, k+N-1$, $k \geqslant 0$, and reference states $x_i^{ref}[\kappa|k]$, $i \in \mathcal{V}$,*

A8.1 it holds that

$$h_i(x_i) \leqslant \mathbf{0} \quad \forall x_i \in x_i^{ref}[\kappa|k] \oplus \mathcal{C}_i \tag{18}$$

and for all $r = 1, \ldots, R_i$,

$$c_{i,r}(x_i, x_{j \in \mathcal{N}_{i,r}}) \leqslant \mathbf{0} \quad \forall x_i \in x_i^{ref}[\kappa|k] \oplus \mathcal{C}_i, \quad \forall x_j \in x_j^{ref}[\kappa|k] \oplus \mathcal{C}_j,\ j \in \mathcal{N}_{i,r}; \tag{19}$$

A8.2 and additionally, if $k > 0$, it holds that

$$\hat{x}_i^*[\kappa|k-1] \in x_i^{ref}[\kappa|k] \oplus \hat{\mathcal{C}}_i. \tag{20}$$

The first part of Ass. 8 states that consistency constraints imply the satisfaction of state constraints (3). The second part of Ass. 8 leads to recursive feasibility. In Sec. 4.2, we propose a method for the determination of reference trajectories that satisfy Ass. 8.

At time $k = 0$, we call a set of reference trajectories $x_i^{\mathrm{ref}}[\kappa|0]$, $\kappa = 0, \ldots, N$, $i \in \mathcal{V}$, and the corresponding input trajectories $u_i^{\mathrm{ref}}[\kappa|0]$, $\kappa = 0, \ldots, N-1$, *initially* *feasible* if

$$
\begin{aligned}
x_i^{\text{ref}}[0|0] &= x_{0,i} \\
x_i^{\text{ref}}[\kappa+1|0] &= \hat{f}_i(x_i^{\text{ref}}[\kappa|0], u_i^{\text{ref}}[\kappa|0])
\end{aligned} \tag{21a}
$$

$$
h_i(x_i) \leqslant \mathbf{0} \quad \forall x_i \in x_i^{\text{ref}}[\kappa|0] \oplus \mathcal{C}_i \tag{21b}
$$

$$
\begin{aligned}
c_{i,r}(x_i, x_{j\in\mathcal{N}_{i,r}}) \leqslant \mathbf{0} \quad & \forall x_i \in x_i^{\text{ref}}[\kappa|0] \oplus \mathcal{C}_i \\
& \forall x_j \in x_j^{\text{ref}}[\kappa|0] \oplus \mathcal{C}_j,\ j \in \mathcal{N}_{i,r} \\
& \forall r \in \{1, \ldots, R_i\}
\end{aligned} \tag{21c}
$$

$$
u_i^{\text{ref}}[\kappa|0] \in \widehat{\mathcal{U}}_i \tag{21d}
$$

$$
x_i^{\text{ref}}[N|0] \in \widehat{\mathcal{X}}_i^f \tag{21e}
$$

where $\widehat{\mathcal{X}}_i^f$ satisfies Ass. 7.

3.3 Guarantees of the Distributed MPC Problems

Now, we show that the local optimization problems always remain feasible given that there exist initially feasible trajectories, and that the closed-loop system is robust asymptotically stable with respect to ξ_i, $i \in \mathcal{V}$. The dynamics of the closed-loop system resulting from the application of u_i^{MPC} defined in (17) are given as

$$
x_i[k+1] = f_i(x_i[k], u_i^{\text{MPC}}(x_i[k]), w_i[k]), \quad i \in \mathcal{V}. \tag{22}
$$

Theorem 9 *Let Ass. 2, 7 and 8 hold, and let there be a set of initially feasible reference trajectories $x_i^{ref}[\cdot|0]$, $i \in \mathcal{V}$. Then for each subsystem $i \in \mathcal{V}$, the local optimization problems* (15) *subject to* (16) *are recursively feasible with respect to the closed-loop dynamics* (22) *for any $w_i[k] \in \mathcal{W}_i$. Moreover, the closed-loop dynamics* (22) *are robustly asymptotically stable with respect to ξ_i, $i \in \mathcal{V}$.*

PROOF. In a first step, we prove recursive feasibility. Thereafter robust asymptotic stability is proven.

Recursive Feasibility: We have to show that there exist feasible solutions to the local optimization problems (15) subject to (16) for $k > 0$ given that there exist initially feasible reference trajectories at $k = 0$. To this end, we recursively construct candidate trajectories $\hat{x}_i^{\text{c}}[\kappa|k]$ for $\kappa = k, \ldots, k+N$ and $\hat{u}_i^{\text{c}}[\kappa|k]$ for $\kappa = k, \ldots, k+N-1$ that satisfy constraints (16).

Firstly, consider $k=0$ and choose candidate trajectories

$$
\begin{aligned}
\hat{x}_i^{\text{c}}[\kappa|0] &= x_i^{\text{ref}}[\kappa|0] \quad \text{for } \kappa = 0, \ldots, N, \\
\hat{u}_i^{\text{c}}[\kappa|0] &= u_i^{\text{ref}}[\kappa|0] \quad \text{for } \kappa = 0, \ldots, N-1
\end{aligned}
$$

for all $i \in \mathcal{V}$ where $x_i^{\text{ref}}[\cdot|0]$ denotes an initially feasible reference trajectory and $u_i^{\text{ref}}[\cdot|0]$ its corresponding input trajectory which together satisfy (21). Now, we show that $\hat{x}_i^{\text{c}}[\cdot|0]$ and $\hat{u}_i^{\text{c}}[\cdot|0]$ also satisfy (16).

Since $x_i^{\text{ref}}[\cdot|0], u_i^{\text{ref}}[\cdot|0]$ satisfy (21a), it holds that

$$
\hat{x}_i^{\text{c}}[0|0] = x_i^{\text{ref}}[0|0] \overset{(21\text{a})}{=} x_{0,i} = x_i[0]
$$

and it follows that (16a) and (16b) are satisfied by $\hat{x}_i^{\text{c}}[\cdot|0], \hat{u}_i^{\text{c}}[\cdot|0]$ for all $i \in \mathcal{V}$. Because $\hat{x}_i^{\text{c}}[\cdot|0] = x_i^{\text{ref}}[\cdot|0]$ and $\mathcal{C}_i$ is a closed neighborhood of the origin, it follows that (16c) is also satisfied by $\hat{x}_i^{\text{c}}[\cdot|0]$. The satisfaction of (16d)-(16e) trivially follows from (21d)-(21e). Thereby, we have shown that there exist feasible solutions to the local optimization problem (15)-(16) at $k=0$.

In a next step, we show that there exist feasible solutions for $k > 0$ using induction. Assume that for all $i \in \mathcal{V}$ at time-step k, there exist predicted nominal state and input trajectories, namely $\hat{x}_i^*[\cdot|k]$ and $\hat{u}_i^*[\cdot|k]$, that solve (15) subject to constraints (16) (induction hypothesis). As we have shown in the previous paragraph, the induction hypothesis is initially fulfilled at $k = 0$. We now show that given the induction hypothesis holds, there also exist feasible candidate trajectories $\hat{x}_i^{\text{c}}[\cdot|k+1]$, $\hat{u}_i^{\text{c}}[\cdot|k+1]$ for all $i \in \mathcal{V}$ at time $k+1$ (induction step). Therefore, we construct for time-steps $k+1$, $k \geqslant 0$, and all subsystems $i \in \mathcal{V}$ candidate trajectories

$$
\hat{x}_i^{\text{c}}[\kappa|k+1] = \begin{cases} \hat{x}_i^*[\kappa|k] & \text{for } \kappa = k+1, \ldots, k+N \\ \hat{f}_i(\hat{x}_i^*[k+N|k], k_i^f(\hat{x}_i^*[k+N|k])) & \\ & \text{for } \kappa = k+N+1 \end{cases} \tag{24a}
$$

$$
\hat{u}_i^{\text{c}}[\kappa|k+1] = \begin{cases} \hat{u}_i^*[\kappa|k] & \text{for } \kappa = k+1, \ldots, k+N-1 \\ k_i^f(\hat{x}_i^*[k+N|k]) & \text{for } \kappa = k+N \end{cases} \tag{24b}
$$

where k_i^f denotes the auxiliary controller associated with $\widehat{\mathcal{X}}_i^f$ and J_i^f as specified in Ass. 7. Now, we show that $\hat{x}_i^{\text{c}}[\cdot|k+1]$ and $\hat{u}_i^{\text{c}}[\cdot|k+1]$ satisfy (16). At first, we obtain for the initial value of $\hat{x}_i^{\text{c}}[\cdot|k+1]$

$$
\begin{aligned}
\hat{x}_i^{\text{c}}[k+1|k+1] &= \hat{x}_i^*[k+1|k] = \hat{f}_i(\hat{x}_i^*[k|k], \hat{u}_i^*[k|k]) \\
&= \hat{f}_i(\hat{x}_i^*[k|k], u_i^{\text{MPC}}(x_i[k])) \\
&\overset{(7)}{=} f_i(x_i[k], u_i^{\text{MPC}}(x_i[k]), w_i[k]) - p_i[k+1] \\
&\overset{(1)}{=} x_i[k+1] - p_i[k+1] \in x_i[k+1] \oplus (-\mathcal{P}_i)
\end{aligned}
$$

and (16a) is satisfied. As $\hat{x}_i^*[\cdot|k]$, $\hat{u}_i^*[\cdot|k]$ satisfy (16b) according to the induction assumption, candidate trajectories (24) satisfy (16b) as well. Due to Ass. 8.2, the candidate trajectory $\hat{x}_i^{\text{c}}[\kappa|k+1] = \hat{x}_i^*[\kappa|k+1]$, $\kappa = k, \ldots, k+N-1$, satisfies (16c) for all $i \in \mathcal{V}$. Besides, as $\hat{u}_i^{\text{c}}[\kappa|k+1] = \hat{u}_i^*[\kappa|k]$, (16d) is trivially satisfied for $\kappa = k+1, \ldots, k+N-1$. Moreover, $\hat{u}_i^{\text{c}}[k+N|k+1] = k_i^f(\hat{x}_i^*[k+N|k]) \in \widehat{\mathcal{U}}_i$ holds due to Ass. 7.2 which applies because $\hat{x}_i^*[k+N|k] \in \widehat{\mathcal{X}}_i^f$, and the satisfaction of (16d) also follows for $\kappa = k+N$. Finally, it follows from Ass. 7.3 that

$$
\hat{x}_i^{\text{c}}[k+N+1|k+1] = \hat{f}_i(\hat{x}_i^*[k+N|k], k_i^f(\hat{x}_i^*[k+N|k])) \in \widehat{\mathcal{X}}_i^f.
$$

Thus, (16e) is satisfied as well. Now, as the induction hypothesis is also initially fulfilled, it inductively follows that for all $k > 0$ there exists a feasible solution to the optimization problems (15)-(16), and we conclude recursive feasibility.

Robust asymptotic stability: At first, we define the auxiliary cost function

$$V_i(x_i[k]) := \min_{\hat{u}_i[\cdot|k]} J_i(\hat{x}_i[\cdot|k], \hat{u}_i[\cdot|k]) \tag{25}$$

subject to $\hat{x}_i[k|k] = x_i[k]$ (which implies (16a)) and (16b)–(16e); J_i is the same as in (15). Then, it holds

$$J_i^*(x_i[k+1]) = \min_{z \in x_i[k+1] \oplus (-\mathcal{P}_i)} V_i(z) \leqslant V_i(\hat{x}_i^*[k+1|k]). \tag{26}$$

Note that here $\hat{x}_i^*[k+1|k] \in x_i[k+1] \oplus (-\mathcal{P}_i)$, and hence the inequality follows from the suboptimality of $\hat{x}_i^*[k+1|k]$ with respect to the minimization. Using this preliminary result, we show in the remainder of the proof that J_i^* is a Lyapunov function in terms of [34, Thm. 1].

By the positive definiteness of $\boldsymbol{Q}_i, \boldsymbol{R}_i$, it holds for the stage cost function $l_i(\xi_i, u_{\xi_i}) = 0$ and $l_i(\hat{x}_i, \hat{u}_i) > 0$ for all $\hat{x}_i \neq \xi_i$, $\hat{u}_i \neq u_{\xi_i}$. Thus, there exists a class $\mathcal{K}$ function α such that $\alpha(0) = 0$ and $0 < \alpha(d_H(x_i, \xi_i \oplus \mathcal{P}_i)) \leqslant J_i^*(x_i)$ for all $x_i \notin \xi \oplus \mathcal{P}_i$. This is because

$$\begin{aligned}
J_i^*(x_i) &\geqslant \min_{\substack{\hat{x}_i[k|k] \in x_i \oplus (-\mathcal{P}_i) \\ \hat{u}_i[k|k] \in \mathcal{U}_i}} l_i(\hat{x}_i[k|k], \hat{u}_i[k|k]) \\
&\geqslant \min_{\hat{x}_i \in x_i \oplus (-\mathcal{P}_i)} ||\hat{x}_i - \xi_i||_{\boldsymbol{Q}_i} \\
&\geqslant \sqrt{\lambda_{\min}(\boldsymbol{Q}_i)} \min_{\hat{x}_i \in x_i \oplus (-\mathcal{P}_i)} ||\hat{x}_i - \xi_i|| \\
&= \sqrt{\lambda_{\min}(\boldsymbol{Q}_i)}\, d_H(x_i, \xi_i \oplus \mathcal{P}_i)
\end{aligned}$$

and we identify $\alpha(z) = \sqrt{\lambda_{\min}(\boldsymbol{Q}_i)} z$ where $\lambda_{\min}(\boldsymbol{Q}_i)$ denotes the smallest eigenvalue of $\boldsymbol{Q}_i$. Moreover, for $x_i[k] \in \xi_i \oplus \mathcal{P}_i$, the unique optimal solution to the local optimization problem (15)-(16) is $\hat{x}_i^*[k|k] = \xi_i$, $\hat{x}_i^*[\cdot|k] \equiv \xi_i$, $\hat{u}_i^*[\cdot|k] \equiv u_{\xi_i}$. Therefore, $J_i^*(x_i) = 0$ for all $x_i[k] \in \xi_i \oplus \mathcal{P}_i$. Thus, by the feasibility of (15)-(16) as shown in the recursive feasibility proof, there exists a class $\mathcal{K}$ function β such that $J_i^*(x_i) \leqslant \beta(d_H(x_i, \xi_i \oplus \mathcal{P}_i))$.

Next, we investigate the descent on J_i^* for the closed-loop system. At first, we observe

$$\begin{aligned}
J_i^*(x_i[k+1]) - J_i^*(x_i[k]) &\overset{(26)}{\leqslant} V_i(\hat{x}_i^*[k+1|k]) - J_i^*(x_i[k]) \\
&\leqslant J_i(\hat{x}_i^{\mathrm{c}}[\cdot|k+1], \hat{u}_i^{\mathrm{c}}[\cdot|k+1]) - J_i^*(x_i[k])
\end{aligned} \tag{27}$$

where the latter inequality follows from definition (25) and the suboptimality of the candidate trajectories. From this, we further derive

$$\begin{aligned}
&J_i(\hat{x}_i^{\mathrm{c}}[\cdot|k+1], \hat{u}_i^{\mathrm{c}}[\cdot|k+1]) - J_i^*(x_i[k]) \\
&= \sum_{\kappa=k+1}^{k+N} l_i(\hat{x}_i^{\mathrm{c}}[\kappa|k+1], \hat{u}_i^{\mathrm{c}}[\kappa|k+1]) + J_i^f(\hat{x}_i^{\mathrm{c}}[k+N+1|k+1]) \\
&\quad - \sum_{\kappa=k}^{k+N-1} l_i(\hat{x}_i^*[\kappa|k], \hat{u}_i^*[\kappa|k]) - J_i^f(\hat{x}_i^*[k+N|k]) \\
&\overset{(24)}{=} l_i(\hat{x}_i^*[k+N|k], k_i^f(\hat{x}_i^*[k+N|k])) - l_i(\hat{x}_i^*[k|k], \hat{u}_i^*[k|k]) \\
&\quad + J_i^f(\hat{f}_i(\hat{x}_i^*[k+N|k], k_i^f(\hat{x}_i^*[k+N|k]))) - J_i^f(\hat{x}_i^*[k+N|k]) \\
&\overset{\text{Ass. 7.4}}{\leqslant} -l_i(\hat{x}_i^*[k|k], \hat{u}_i^*[k|k]) \\
&\leqslant -||\hat{x}_i^*[k|k] - \xi_i||_{\boldsymbol{Q}_i}^2 = -\gamma_{V_i}(||\hat{x}_i^*[k|k] - \xi_i||)
\end{aligned} \tag{28}$$

where γ_{V_i} is a class $\mathcal{K}$ function. Note that

$$\begin{aligned}
||\hat{x}_i^*[k|k] - \xi_i|| &= d_H(\hat{x}_i^*[k|k] + p_i[k|k], \xi_i + p_i[k|k]) \\
&\geqslant d_H(\hat{x}_i^*[k|k] + p_i[k|k], \xi_i \oplus \mathcal{P}_i) = d_H(x_i[k], \xi_i \oplus \mathcal{P}_i)
\end{aligned}$$

where $p_i[k|k] = x_i[k] - \hat{x}_i^*[k|k] \in \mathcal{P}_i$, which finally yields, together with (27)-(28),

$$J_i^*(x_i[k+1]) - J_i^*(x_i[k]) \leqslant -\gamma_{V_i}(d_H(x_i[k], \xi_i \oplus \mathcal{P}_i)) < 0$$

for all $x_i[k] \notin \xi_i \oplus \mathcal{P}_i$. Thus, by [34, Thm. 1], we have shown that J_i^* is a Lyapunov function. We further conclude the asymptotic stability of $\xi_i \oplus \mathcal{P}_i$ under closed-loop dynamics (22), and equivalently the robust asymptotic stability of ξ_i for all $i \in \mathcal{V}$. □

Next, we show that state and input trajectories of the closed-loop system (22) satisfy all constraints.

Proposition 10 *Let the same premises hold as in Thm. 9, i.e., let Ass. 2, 7 and 8 hold, and let there be a set of initially feasible reference trajectories $x_{i \in \mathcal{V}}^{ref}[\cdot|0]$. Then for all $i \in \mathcal{V}$, the state trajectory $x_i[k]$, $k \geqslant 0$, of the closed-loop system (22) and the corresponding input trajectory $u_i[k] = u_i^{MPC}(x_i[k])$, $k \geqslant 0$, satisfy state constraints (3) and input constraints (4).*

PROOF. From Thm. 9, it follows that the local optimization problems are recursively feasible and hence state and input trajectories $x_i[k]$ and $u_i[k]$ of the closed-loop dynamics (22) are well-defined for $k \geqslant 0$. According to (17), the input trajectory is given by $u_i[k] = u_i^{\mathrm{MPC}}(x_i[k]) = \hat{u}_i^*[k|k] + K_i(x_i[k], \hat{x}_i^*[k|k])$ where $\hat{u}_i^*[k|k]$ is such that (16d) holds, and by definition (10), we have $K_i(\cdot,\cdot) \in \Delta\mathcal{U}_i$. Then, it follows that

$$u_i[k] \in \hat{\mathcal{U}}_i \oplus \Delta\mathcal{U}_i = (\mathcal{U}_i \ominus \Delta\mathcal{U}_i) \oplus \Delta\mathcal{U}_i \overset{\text{Lem. 5}}{\subseteq} \mathcal{U}_i.$$

and input constraint (4) holds. Next, due to consistency constraint (16c) and $p_i[k] = x_i[k] - \hat{x}_i[k] \in \mathcal{P}_i$ (cf. (9)), it follows that for all $i \in \mathcal{V}$

$$x_i[k] = \hat{x}_i[k] + p_i[k] \in (x_i^{\text{ref}}[k|k] \oplus \widehat{\mathcal{C}}_i) \oplus \mathcal{P}_i = x_i^{\text{ref}}[k|k] \oplus \mathcal{C}_i.$$

Then due to Ass. 8, the satisfaction of state constraints (3a) and coupled state constraints (3b) follows from (18)-(19). □

3.4 Discussion

If reference trajectories x_i^{ref} and consistency constraint sets $\mathcal{C}_i$ satisfy Ass. 8.1, then state constraint satisfaction is implied via consistency constraint (16d). Therefore, no further state constraints apart from the consistency constraint need to be considered in the local optimization problem (15)-(16). Hence, the local optimization problems are subject to fewer constraints compared to other DMPC schemes allowing for coupled state constraints, see [14–17].

Additionally, if the consistency constraint sets, namely $\widehat{\mathcal{C}}_i$, are chosen to be convex, then the local optimization problems (15)-(16) may be convex even if the original state constraints (3) are non-convex. In particular, this is the case if dynamics (1) are linear and the input constraint sets $\mathcal{U}_i$ convex; note that linear dynamics give rise to convex $\mathcal{P}_i$ and $\widehat{\mathcal{X}}_i^f$. However, even if dynamics (1) are nonlinear, the local optimization problems (15)-(16) approximately constitute a convex problem for sufficiently small $\widehat{\mathcal{C}}_i$. This is because consistency constraint (16c) restricts the optimal solutions to a neighborhood of a reference trajectory in which the system dynamics are approximately linear.

Moreover, we do not require that reference trajectories $x_{i\in\mathcal{V}}^{\text{ref}}[\cdot|k]$ satisfy any dynamics (cf. Ass. 8) after initialization, i.e., for $k > 0$. This is in contrast to other DMPC approaches employing consistency constraints [17, 18, 35]. Moreover, we allow that reference trajectories can be updated at every time step which allows for enhanced performance compared to algorithms with fixed reference trajectories (see performance comparison Sec. 5.3). We detail the reference trajectory update in the next section.

4 Parallelized DMPC Algorithm

In the previous section, we have formulated local optimization problems for all subsystems and assumptions that allow for their parallelized evaluation while ensuring recursive feasibility. While the satisfaction of Ass. 1, 2, 7 needs to be ensured during the initialization, Ass. 8 is the only assumption that needs to be taken into account online. In this section, we detail the initialization procedure, present a recursive algorithm to update reference trajectories such that Ass. 8 is satisfied, and state the overall DMPC algorithm.

4.1 Initialization

In order to solve the local optimization problems, some parameters need to be chosen offline. Therefore, we suggest the following initialization procedure:

Step 1 (Problem Formulation): Formulate the control problem such that neighboring agents have those constraints that couple their states in common (Ass. 1). This can be always achieved by adding coupled state constraints to neighboring subsystems.

Step 2 (RPI sets): For each subsystem $i \in \mathcal{V}$, determine an RPI set $\mathcal{P}_i$ and the corresponding state-feedback controller K_i. For each subsystem, the computations are independent of the other subsystems due to the decoupled dynamics. Available methods are reviewed in Remark 3. The existence of $\mathcal{P}_i$ and K_i, $i \in \mathcal{V}$, is assumed in Ass. 2.

Step 3 (Terminal constraints, stage cost function, and consistency constraint set): For each subsystem $i \in \mathcal{V}$, choose symmetric positive-definite matrices $\boldsymbol{Q}_i, \boldsymbol{R}_i$ and a stage cost function $l_i(x,u) = ||x - \xi_i||^2_{\boldsymbol{Q}_i} + ||u - u_{\xi_i}||^2_{\boldsymbol{R}_i}$. Thereafter, determine terminal sets $\widehat{\mathcal{X}}_i^f$ and corresponding terminal cost functions J_i^f such that Ass. 7 holds. To this end, we first construct auxilliary sets $\widetilde{\mathcal{X}}_i^f$, $i \in \mathcal{V}$, that satisfy Ass. 7.2-7.4. These can be computed independently for each subsystem due to the decoupled dynamics. For systems whose linearization is stabilizable, an approach using the discrete-time algebraic Riccati equation can be chosen. Let

$$\hat{f}_i(\hat{x}_i, \hat{u}_i) = \boldsymbol{A}_i \hat{x}_i + \boldsymbol{B}_i \hat{u}_i + \hat{f}_i'(\hat{x}_i, \hat{u}_i)$$

for some matrices $\boldsymbol{A}_i$, $\boldsymbol{B}_i$ with respective dimensions and $\hat{f}_i' : \mathbb{R}^{n_i} \times \mathbb{R}^{m_i} \to \mathbb{R}^{n_i}$. By solving the discrete-time algebraic Riccati equation

$$\begin{aligned}\boldsymbol{P}_i =& \boldsymbol{A}_i^T \boldsymbol{P}_i \boldsymbol{A}_i \\ &- (\boldsymbol{A}_i^T \boldsymbol{P}_i \boldsymbol{B}_i)(\boldsymbol{R}_i + \boldsymbol{B}_i^T \boldsymbol{P}_i \boldsymbol{B}_i)^{-1}(\boldsymbol{B}_i^T \boldsymbol{P}_i \boldsymbol{A}_i) + \boldsymbol{Q}_i,\end{aligned}$$

a positive definite matrix $\boldsymbol{P}_i$ is obtained. Choose $k_i^f(x_i) = -(\boldsymbol{R}_i + \boldsymbol{B}_i^T \boldsymbol{P}_i \boldsymbol{B}_i)^{-1} \boldsymbol{B}_i^T \boldsymbol{P}_i \boldsymbol{A}_i x_i$ and $J_i^f(x_i) = \sigma_i x_i^T \boldsymbol{P}_i x_i$ with a scalar $\sigma_i \geqslant 0$. Then, there exists a sufficiently small scalar $\tilde{\gamma}_i \geqslant 0$ such that Ass. 7.2, 7.4 hold for all $x_i \in \widetilde{\mathcal{X}}_i^f = \{x | J_i^f(x) \leqslant \tilde{\gamma}_i\}$ [3, Remark 5.15]. At last, choose $\widehat{\mathcal{C}}_i$ as some neighborhood of the origin and set $\mathcal{C}_i = \widehat{\mathcal{C}}_i \oplus \mathcal{P}_i$ (this choice might need to be refined later). Then select in a centralized way [2] $\widehat{\mathcal{X}}_i^f = \{x | J_i^f(x) \leqslant \gamma_i\} \subseteq \widetilde{\mathcal{X}}_i^f$ with a sufficiently small scalar $\gamma_i \in [0, \tilde{\gamma}_i]$ such that the constraints in Ass. 7.1

[2] Also a distributed computation of $\gamma_{i\in\mathcal{V}}$ is possible. For example, each subsystem incrementally increases γ_i and checks at each step constraint satisfaction. For each subsystem, the largest γ_i is chosen that still satisfies the constraints. Alternatively, distributed iterative optimization algorithms can be considered [36].

hold for all $i \in \widehat{\mathcal{X}}_i^f$ and all subsystems $i \in \mathcal{V}$. Note that Ass. 7.3 is satisfied due to the choice of $\widehat{\mathcal{X}}_i^f$ as a super-level set of J_i^f.

Step 4 (Initially feasible reference trajectories): Determine initially feasible reference trajectories $x_i^{\mathrm{ref}}[\cdot|0]$, $i \in \mathcal{V}$, that satisfy (21), e.g. by solving a centralized optimization problem subject to (21). If no such trajectories could be found, choose a smaller set $\widehat{\mathcal{C}}_i$.

Note that (21b)-(21c) can often be simplified. If $h_i, c_{i,r}$ are *linear* and $\mathcal{C}_{i\in\mathcal{V}}$ are polytopes, then (21b)-(21c) can be expressed as linear algebraic inequalities. Libraries for computations with polytopes are MPT3 (Matlab) [37], Polyhedra (Julia) [38] and Polytope (Python) [39]. Alternatively, if $h_i, c_{i,r}$ are nonlinear but still *Lipschitz continuous*, then there exist scalars H_i and $C_{i,r}$ such that [3]

$$\begin{aligned} |h_i(x') - h_i(x'')| &\leqslant H_i\|x' - x''\| \\ |c_{i,r}(y') - c_{i,r}(y'')| &\leqslant C_{i,r}\|y' - y''\| \end{aligned}$$

where $x', x'' \in \mathbb{R}^{n_i}$ and $y', y'' \in \times_{j\in\{i\}\cup\mathcal{N}_{i,r_i}} \mathbb{R}^{n_j}$. Define the maximal distance of any point in some compact set $\mathcal{A} \subset \mathbb{R}^n$ to some point $x \in \mathbb{R}^n$ as $d_{\max}(x, \mathcal{A}) := \sup_{z\in\mathcal{A}} \|x - z\|$. Then, there exist scalars $\nu_{h_i} := d_{\max}(\mathbf{0}, \mathcal{C}_i)$ and $\nu_{c_{i,r}} := d_{\max}(\mathbf{0}, \times_{j\in\{i\}\cup\mathcal{N}_{i,r}} \mathcal{C}_j)$, and it holds

$$\begin{aligned} &\sup_{x\in x_i^{\mathrm{ref}}\oplus\mathcal{C}_i} |h_i(x_i^{\mathrm{ref}}[\kappa|0]) - h_i(x)| \leqslant H_i\nu_{h_i} \\ &\sup_{x\in \times_{j\in\{i\}\cup\mathcal{N}_{i,r}} x_j^{\mathrm{ref}}\oplus\mathcal{C}_j} |c_{i,r}(x_{j\in\{i\}\cup\mathcal{N}_{i,r}}^{\mathrm{ref}}[\kappa|0]) - c_{i,r}(x)| \leqslant C_{i,r}\nu_{c_{i,r}}. \end{aligned}$$

Thus, we can replace (21b)-(21c) by

$$\begin{aligned} h_i(x_i^{\mathrm{ref}}[\kappa|0]) &\leqslant -H_i\nu_{h_i} \\ c_{i,r}(x_i^{\mathrm{ref}}[\kappa|0], x_{j\in\mathcal{N}_{i,r}}^{\mathrm{ref}}[\kappa|0]) &\leqslant -C_{i,r}\nu_{c_{i,r}}. \end{aligned}$$

4.2 Online Determination of Reference Trajectories

In Ass. 8, general conditions are stated that allow for the parallelized evaluation of the local optimization problems (15) subject to (16) while preserving recursive feasibility (cf. Thm. 9) and ensuring state constraint satisfaction (cf. Prop. 10). Algorithms that ensure the satisfaction of Ass. 8 are essential to the proposed DMPC scheme. The generality of Ass. 8 allows for various such algorithms. In this section, we propose such an algorithm that updates a previous reference trajectory for each subsystem $i \in \mathcal{V}$. The algorithm is distributed.

In particular, let each subsystem $i \in \mathcal{V}$ be initialized with a set of initially feasible reference trajectories $x_i^{\mathrm{ref}}[\cdot|0]$. Then for $k \geqslant 0$, each subsystem $i \in \mathcal{V}$ checks for all $\kappa \in \{k+1, \ldots, k+N\}$ separately if

$$h_i(x_i) \leqslant \mathbf{0} \qquad \forall x_i \in \hat{x}_i^*[\kappa|k] \oplus \mathcal{C}_i \tag{29}$$

and

$$\begin{aligned} c_{i,r}(x_i, x_{j\in\mathcal{N}_{i,r}}) \leqslant \mathbf{0} \qquad &\forall x_i \in \hat{x}_i^*[\kappa|k] \oplus \mathcal{C}_i, \\ \forall x_j \in (\hat{x}_j^*[\kappa|k] \oplus \mathcal{C}_j) \cup (&x_j^{\mathrm{ref}}[\kappa|k] \oplus \mathcal{C}_j),\ j \in \mathcal{N}_{i,r} \end{aligned} \tag{30}$$

for all $r = 1, \ldots, R_i$.

Remark 11 *From a practical point of view,* (29)-(30) *can be efficiently evaluated analogously to Step 4 in the previous section. Following the same reasoning, if* $h_i, c_{i,r}$ *are* linear *and* $\mathcal{C}_{i\in\mathcal{V}}$ *are polytopes, then* (29)-(30) *can be expressed as linear algebraic inequalities. Alternatively, if* $h_i, c_{i,r}$ *are only* Lipschitz continuous, *then* (29)-(30) *can be replaced by*

$$\begin{aligned} h_i(\hat{x}_i^*[\kappa|k]) &\leqslant -H_i\nu_{h_i} \\ c_{i,r}(\hat{x}_i^*[\kappa|k], \hat{x}_{j\in\mathcal{N}_{i,r}}^*[\kappa|k]) &\leqslant -C_{i,r}\nu_{c_{i,r}} \\ c_{i,r}(\hat{x}_i^*[\kappa|k], x_{j\in\mathcal{N}_{i,r}}^{ref}[\kappa|k]) &\leqslant -C_{i,r}\nu_{c_{i,r}} \end{aligned}$$

where H_i, $C_{i,r}$, ν_{h_i} *and* $\nu_{c_{i,r}}$ *are as before. In other cases, the evaluation is more elaborate.*

Based on conditions (29)-(30), the updated reference trajectories are defined for each subsystem $i \in \mathcal{V}$ and $\kappa = k+1, \ldots, k+N-1$ as

$$x_i^{\mathrm{ref}}[\kappa|k+1] := \begin{cases} \hat{x}_i^*[\kappa|k] & \text{if (29)-(30) hold for } \kappa, \\ x_i^{\mathrm{ref}}[\kappa|k] & \text{otherwise,} \end{cases} \tag{31a}$$

$$x_i^{\mathrm{ref}}[k+N|k+1] := \hat{x}_i^*[k+N|k]. \tag{31b}$$

Intuitively, each subsystem attempts to change its previous reference state $x_i^{\mathrm{ref}}[\kappa|k]$ to the predicted state $\hat{x}_i^*[\kappa|k]$ for any κ. The reference states, however, are only changed if no state within a $\mathcal{C}_i$-neighborhood of the predicted state violates any of the state constraints (3) which is checked by conditions (29) and (30). In condition (30) on the coupled state constraints, both the predicted states $\hat{x}_{j\in\mathcal{N}_{i,r}}^*$ and the previous reference states $x_{j\in\mathcal{N}_{i,r}}^{\mathrm{ref}}$ of the neighboring subsystems are considered. This allows for the parallelized computation of reference trajectories. The reference trajectory update of subsystem i for $k \geqslant 0$ is summarized in Algorithm 1. It guarantees the satisfaction of Ass. 8 as stated by the following proposition.

Proposition 12 *Let Ass. 1 and Ass. 7.1 be satisfied, and let there be a set of initially feasible reference trajectories* $x_i^{ref}[\cdot|0]$, $i \in \mathcal{V}$. *Then for all* $i \in \mathcal{V}$, *reference trajectories* $x_i^{ref}[\cdot|0]$ *and* $x_i^{ref}[\cdot|k]$, $k > 0$, *recursively defined in* (31), *satisfy Ass. 8 for all times* $k \geqslant 0$.

[3] For simplicity, h_i and $c_{i,r}$ are assumed to be scalar. In the case of vectors, subsequent calculations hold for row-wise evaluation.

Algorithm 1 Reference Trajectory Update of Subsystem i

Input $\hat{x}_i^*[\cdot|k]$, $\hat{x}_{j\in\mathcal{N}_i}^*[\cdot|k]$, $x_i^{\text{ref}}[\cdot|k]$, $x_{j\in\mathcal{N}_i}^{\text{ref}}[\cdot|k]$
Output $x_i^{\text{ref}}[\cdot|k+1]$

```
for κ = k+1, ..., k+N do
    if (29) and (30) hold for all r = 1, ..., R_i then
        x_i^ref[κ|k+1] := x̂_i^*[κ|k];
    else
        x_i^ref[κ|k+1] := x_i^ref[κ|k];
    end if
end for
x_i^ref[k+N|k+1] := x̂_i^*[k+N|k];
```

PROOF. *Satisfaction of Ass. 8.1:* At first, we show by induction that $x_i^{\text{ref}}[\cdot|k]$ satisfies Ass. 8.1 for all $k \geqslant 0$. Note that initially feasible reference trajectories $x_i^{\text{ref}}[\cdot|0]$ satisfy Ass. 8.1 for $k=0$ by definition. Next, assuming that $x_i^{\text{ref}}[\kappa|k]$, $\kappa = k,\dots,k+N-1$, satisfies Ass. 8.1, we show that also $x_i^{\text{ref}}[\kappa|k+1]$, $\kappa = k+1,\dots,k+N$, satisfies Ass. 8.1. We split the proof in two parts: at first, we show that $x_i^{\text{ref}}[\kappa|k+1]$, $\kappa = k+1,\dots,k+N-1$, as defined in (31a) satisfies Ass. 8.1. Thereafter, we show that also $x_i^{\text{ref}}[k+N|k+1]$ as defined in (31b) satisfies Ass. 8.1 for $\kappa = k+N$. From this, we can conclude, that $x_i^{\text{ref}}[\cdot|k+1]$ overall satisfies Ass. 8.1 at time $k+1$.

Part 1: Consider $x_i^{\text{ref}}[\kappa|k+1]$, $\kappa = k+1,\dots,k+N-1$, as defined in (31a). Note that $x_i^{\text{ref}}[\kappa|k+1] \in \{\hat{x}_i^*[\kappa|k], x_i^{\text{ref}}[\kappa|k]\}$ for all $i \in \mathcal{V}$. We consider two cases:
Case 1: Let conditions (29)-(30) be satisfied. Then observe that (29) is equivalent to (18). Moreover, as for any neighbor $j \in \mathcal{N}_i$, it holds that $x_j^{\text{ref}}[\kappa|k+1] \in \{\hat{x}_j^*[\kappa|k], x_j^{\text{ref}}[\kappa|k]\}$, (30) implies (19). We conclude the satisfaction of Ass. 8.1 in case 1.
Case 2: If conditions (29)-(30) are *not* satisfied for κ, then $x_i^{\text{ref}}[\kappa|k+1] = x_i^{\text{ref}}[\kappa|k]$. As $x_i^{\text{ref}}[\kappa|k]$ satisfies (18) at time-step k, it does so at $k+1$ since (18) is time-invariant and does not depend on other subsystems. In order to determine the satisfaction of (19), we assume at first that conditions (29)-(30) are *not* satisfied at κ and any of the neighbors $j \in \mathcal{N}_i$. Then $x_j^{\text{ref}}[\kappa|k+1] = x_j^{\text{ref}}[\kappa|k]$ for all $j \in \mathcal{N}_i$, and the satisfaction of (19) trivially follows from the previous time step k when $x_j^{\text{ref}}[\kappa|k]$ satisfies (19) by assumption. Next, assume that for all $j \in \tilde{\mathcal{N}}_i$, where $\tilde{\mathcal{N}}_i \subseteq \mathcal{N}_i$ is an arbitrary subset of $\mathcal{N}_i$, conditions (29)-(30) are satisfied. Then, $x_j^{\text{ref}}[\kappa|k+1] = \hat{x}_j^*[\kappa|k]$ for all $j \in \tilde{\mathcal{N}}_i$. Let r be any $r \in \{1,\dots,R_i\}$ and consider the coupled state constraint $c_{i,r}$. By Ass. 1, there exist constraints $c_{j,r'}$, $r' \in \{1,\dots,R_j\}$, for all neighbors $j \in \mathcal{N}_{i,r}$ of subsystem i such that

$$c_{j,r'}(x_j, x_{j'\in\mathcal{N}_{j,r'}}) \equiv c_{i,r}(x_i, x_{i'\in\mathcal{N}_{i,r}}). \qquad (32)$$

For these $c_{j,r'}$, the satisfaction of (30) implies

$$\begin{aligned} c_{j,r'}(x_j, x_{j'\in\mathcal{N}_{j,r'}}) \leqslant \mathbf{0} \qquad & \forall x_j \in x_j^{\text{ref}}[\kappa|k+1] \oplus \mathcal{C}_j, \\ & \forall x_{j'} \in x_{j'}^{\text{ref}}[\kappa|k+1] \oplus \mathcal{C}_{j'},\ j' \in \mathcal{N}_{j,r'}. \end{aligned}$$

Due to (32) and since $i \in \mathcal{N}_{j,r'}$, we can rewrite the latter equation as

$$\begin{aligned} c_{i,r}(x_i, x_{j\in\mathcal{N}_{i,r}}) \leqslant \mathbf{0} \qquad & \forall x_i \in x_i^{\text{ref}}[\kappa|k+1] \oplus \mathcal{C}_i, \\ & \forall x_j \in x_j^{\text{ref}}[\kappa|k+1] \oplus \mathcal{C}_j,\ j \in \mathcal{N}_{i,r} \end{aligned}$$

where $\mathcal{N}_{i,r} = (\mathcal{N}_{j,r'} \cup \{j\})\backslash\{i\}$. This is equal to (19) at time $k+1$. Thus, we have shown that even in the case that conditions (29)-(30) are satisfied for some of the neighbors of subsystem i, namely $j \in \tilde{\mathcal{N}}_i$, (19) still holds. Thereby, Ass. 8.1 is also satisfied in case 2.

Part 2: Consider $x_i^{\text{ref}}[\kappa|k+1]$ at $\kappa = k+N$ where $x_i^{\text{ref}}[k+N|k+1] := \hat{x}_i^*[k+N|k]$ by (31b). Since $\hat{x}_i^*[k+N|k] \in \hat{\mathcal{X}}_i^f$ according to terminal constraint (16e), the choice $x_i^{\text{ref}}[\kappa|k+1]$ satisfies Ass. 8.1 due to Ass. 7.1.

Satisfaction of Ass. 8.2: At last, we show the satisfaction of Ass. 8.2. To this end, we consider the two cases for $x_i^{\text{ref}}[\kappa|k+1]$, $\kappa = k+1,\dots,k+N-1$, from part 1 again. In case 1, if conditions (29)-(30) are satisfied for κ, then $x_i^{\text{ref}}[\kappa|k+1] = \hat{x}_i^*[\kappa|k]$ which trivially implies (20). In case 2, if conditions (29)-(30) are *not* satisfied, we have

$$\hat{x}_i^*[\kappa|k] \overset{(16c)}{\in} x_i^{\text{ref}}[\kappa|k] \oplus \mathcal{C}_i = x_i^{\text{ref}}[\kappa|k+1] \oplus \mathcal{C}_i$$

which is equivalent to (20). For $x_i^{\text{ref}}[\kappa|k+1]$ at $\kappa = k+N$, (20) is trivially satisfied. Altogether, we also conclude the satisfaction of Ass. 8.2. □

Other strategies to update the reference trajectories are also possible; e.g., sets $\mathcal{C}_i$ could be varied in addition.

4.3 Distributed MPC Algorithm

The distributed MPC algorithm is initialized as detailed in Sec. 4.1. Then, the distributed MPC algorithms as given in Algorithm 2 are executed in parallel by all subsystems $i \in \mathcal{V}$. Note that Algorithm 1 can be replaced by any other algorithm that ensures the satisfaction of Ass. 8 for all $k \geqslant 0$.

Remark 13 (Iterative version) *If Algorithm 2 is initialized with suboptimal initially feasible reference trajectories, then an iterative version of the proposed DMPC scheme can lead to an improved performance. Therefore, Algorithm 2 is modified as follows: After solving local optimization problem* (15)-(16)*, a new reference trajectory* $x_i^{ref,+}[\kappa|k]$, $\kappa = k,\dots,k+N-1$, *is computed as*

$$x_i^{ref,+}[\kappa|k] := \begin{cases} \hat{x}_i^*[\kappa|k] & (29)\text{-}(30) \textit{ hold for } \kappa, \\ x_i^{ref}[\kappa|k] & \textit{otherwise.} \end{cases}$$

Algorithm 2 Distributed MPC Algorithm for Subsystem i

Input $\xi_i, \mathcal{P}_i, \mathcal{C}_i, K_i, l_i, J_i^f, \hat{\mathcal{X}}_i^f, x_i^{\mathrm{ref}}[\cdot|0]$

$k \leftarrow 0$;
while $k \geqslant 0$ **do**
    measure $x_i[k]$;
    solve (15) subject to (16);
    communicate $\hat{x}_i^*[\cdot|k]$ to all neighbors $j \in \mathcal{N}_i$;
    receive $\hat{x}_j^*[\cdot|k]$, $j \in \mathcal{N}_i$;
    apply u_i^{MPC} as given in (17);
    compute $x_i^{\mathrm{ref}}[\cdot|k+1]$ with Algorithm 1;
    communicate $\hat{x}_i^{\mathrm{ref}}[\cdot|k+1]$ to all neighbors $j \in \mathcal{N}_i$;
    receive $\hat{x}_j^{\mathrm{ref}}[\cdot|k+1]$, $j \in \mathcal{N}_i$;
    $k \leftarrow k+1$;
end while

Then, the local optimization problems are repeatedly solved. Depending on the available computation time, this procedure can be repeated multiple times, before applying the last computed control input $u_i^{MPC}[k]$*. An example is presented in Sec. 5.*

Remark 14 (Dynamic couplings) *The proposed approach is not directly generalizable to systems with dynamic couplings. The main challenges are twofold. Firstly, a parallelized non-iterative DMPC scheme presumably only yields approximate asymptotic convergence to a neighborhood of the desired states, which [35] alleges. Secondly, the heterogeneity in the subsystems' constraints hardens the update of the reference trajectories (to circumvent this, [17, 18] employ fixed reference trajectories). It is expected that further conditions that account for this heterogeneity in the case of systems with bounded dynamic couplings need to be invoked to ensure recursive feasability. This however is a non-trivial problem on its own and left for future research. For systems with bounded dynamic couplings, however, the proposed DMPC scheme can be extended (though conservatively) along the lines of [40, Sec. 4.6].*

5 Simulation

In this section, we investigate the performance of the proposed algorithm with respect to computation time and optimality. To this end, we consider mobile robots subject to connectivity and collision avoidance constraints, which are often considered coupled state constraints in the literature [14, 41, 42].

In particular, we consider the kinematic model of three-wheeled omni-directional robots. The state of robot i is given as $\mathbf{x}_i := [x_i, y_i, \psi_i]$ where x_i, y_i denote the position coordinates and ψ_i the orientation; its position is defined as $\mathbf{x}_i^{\mathrm{pos}} := [x_i, y_i]^T$. The dynamics of robot i are given as

$$\dot{\mathbf{x}}_i = \boldsymbol{R}(\psi_i)\,(\boldsymbol{B}_i^T)^{-1}\, r_i\, u_i + w_i \tag{33}$$

where

$$\boldsymbol{R}(\psi_i) = \begin{bmatrix} \cos(\psi_i) & -\sin(\psi_i) & 0 \\ \sin(\psi_i) & \cos(\psi_i) & 0 \\ 0 & 0 & 1 \end{bmatrix},\ \boldsymbol{B}_i = \begin{bmatrix} 0 & \cos(\pi/6) & -\cos(\pi/6) \\ -1 & \sin(\pi/6) & \sin(\pi/6) \\ l_i & l_i & l_i \end{bmatrix},$$

$l_i = 0.2$ is the radius of the robot body, $r_i = 0.02$ the wheel radius, $u_i = [u_{i,1}, u_{i,2}, u_{i,3}]^T$ the angular velocity of the wheels, and $w_i = [w_{i,1}, w_{i,2}, w_{i,3}]^T \in \mathcal{W}_i \subset \mathbb{R}^3$ a bounded uniformly distributed disturbance. The corresponding nominal dynamics are

$$\dot{\hat{\mathbf{x}}}_i = \boldsymbol{R}(\hat{\psi}_i)\,(\boldsymbol{B}_i^T)^{-1}\, r_i\, \hat{u}_i \tag{34}$$

where $\hat{\mathbf{x}}_i := [\hat{x}_i, \hat{y}_i, \hat{\psi}_i]$ is the nominal state of robot i and $\hat{u}_i$ the nominal input. The nominal position is defined as $\hat{\mathbf{x}}_i^{\mathrm{pos}} := [\hat{x}_i, \hat{y}_i]^T$.

5.1 Controller Design

We follow the four initialization steps in Sec. 4.1 as presented next in great detail. Let us consider three robots $\mathcal{V} = \{1, 2, 3\}$ with nonlinear dynamics (33) which shall move from an initial formation x_0 to a target formation ξ. All robots $i \in \mathcal{V}$ are subject to connectivity constraints

$$||\mathbf{x}_i^{\mathrm{pos}} - \mathbf{x}_j^{\mathrm{pos}}|| \leqslant d^{\max}, \qquad j \in \mathcal{N}_i := \mathcal{V}\backslash\{i\} \tag{35}$$

with $d^{\max} = 2.9$, and input constraints $||u_i||_\infty \leqslant 15$ where $||\cdot||_\infty$ denotes the maximum norm. It can be easily verified that the coupled state constraints satisfy Ass. 1 by design (step 1). The continuous-time nominal dynamics (34) are discretized with time-step $\Delta t = 1/3$ using a 4th-order Runge-Kutta integration algorithm where the nominal inputs are applied as zero-order hold. The control input applied to the robots is

$$u_i(t) = \hat{u}_i(t) + K_i(\mathbf{x}_i(t), \hat{\mathbf{x}}_i(t)) \tag{36}$$

where $\hat{u}_i(t) = \hat{u}_i^{\mathrm{MPC}}(\mathbf{x}_i[k])$, $t \in [k\,\Delta t, (k+1)\Delta t)$.

Next, we construct RPI sets $\mathcal{P}_i$, $i \in \mathcal{V}$ (step 2). To this end, we choose the continuous-time auxiliary controller K_i as

$$\begin{aligned} K_i(\mathbf{x}_i(t), \hat{\mathbf{x}}_i(t)) &= \boldsymbol{B}_i^T \boldsymbol{R}(-\psi_i)\, \boldsymbol{\Lambda}_i\, (\mathbf{x}_i(t) - \hat{\mathbf{x}}_i(t)) \\ &= \boldsymbol{B}_i^T \boldsymbol{R}(-\psi_i)\, \boldsymbol{\Lambda}_i\, p_i(t) \end{aligned} \tag{37}$$

where $\boldsymbol{\Lambda}_i \in \mathbb{R}^{3\times3}$ is Hurwitz. Substituting (37) into (36), (36) into (33), and computing $\dot{p}_i = \dot{\mathbf{x}}_i - \dot{\hat{\mathbf{x}}}_i$ yields

$$\dot{p}_i(t) = \boldsymbol{\Lambda}_i p_i(t) + w_i(t). \tag{38}$$

We choose $\boldsymbol{\Lambda}_i = -\mathrm{diag}(6, 6, 5.5)$ for all $i \in \mathcal{V}$. To numerically compute the RPI set $\mathcal{P}_i$, we exactly discretize the continuous-time deviation dynamics (38) and obtain

$$p_i[k+1] = \boldsymbol{\Lambda}_i^{\mathrm{d}} p_i[k] + w_i[k] \tag{39}$$

with $\mathbf{\Lambda}_i^{\mathrm{d}} = e^{\mathbf{\Lambda}_i \Delta t}$, and $w_i[k] \in \mathcal{W}_i^d := \int_{\tau=0}^{\Delta t} e^{\mathbf{\Lambda}_i \tau} d\tau \; \mathcal{W}_i$. We assume that $\mathcal{W}_i^d = \{w_i \,|\, |w_{i,1}| < 0.1, \; |w_{i,2}| < 0.1, \; |w_{i,3}| < \pi/32\}$, or equivalently, $\mathcal{W}_i = \{w_i \,|\, |w_{i,1}| < 0.6940, \; |w_{i,2}| < 0.6940, \; |w_{i,3}| < 0.6429\}$. Then, we can compute $\mathcal{P}_i$ as in [24]. The resulting RPI set $\mathcal{P}_i$ is a box given as $\mathcal{P}_i = \{p_i \,|\, |p_{i,1}| < \bar{p}_{i,1} = 0.1157, \; |p_{i,2}| < \bar{p}_{i,2} = 0.1157, |p_{i,3}| < \bar{p}_{i,3} = 0.1169\}$. An outer approximation of $\Delta\mathcal{U}_i$ is computed in accordance with (10). Then, Ass. 2 is satisfied. For required set computations, we use the MPT3 toolbox [37] for Matlab and YALMIP [43].

Now, we choose the performance matrices as $\boldsymbol{Q}_1 = \mathrm{diag}(100, 100, 100)$, $\boldsymbol{Q}_2 = \boldsymbol{Q}_3 = \mathrm{diag}(1, 1, 50)$, $\boldsymbol{R}_1 = \mathrm{diag}(1, 1, 1)$, $\boldsymbol{R}_2 = \boldsymbol{R}_3 = \mathrm{diag}(5, 5, 5)$, and consistency constraint sets $\widehat{\mathcal{C}}_i = \{\zeta \in \mathbb{R}^3 \,|\, ||[\zeta_1, \zeta_2]^T|| \leqslant \sqrt{2}\,\bar{c}_i\}$ for all $i \in \mathcal{V}$ where $\bar{c}_i = 0.125$. Matrices $\boldsymbol{Q}_1$ and $\boldsymbol{R}_1$ are chosen such that subsystem 1 tends faster to its desired state than the other subsystems. Thereby, subsystem 1 attempts to violate the connectivity constraint, which is of interest in the performance analysis conducted later (Sec. 5.3). The terminal cost functions $J_{i\in\mathcal{V}}^f$ and terminal sets $\hat{\mathcal{X}}_{i\in\mathcal{V}}^f$ are computed via the discrete-time algebraic Riccati equation as outlined in Sec. 4.1, step 3, and Ass. 7.2-7.4 are satisfied. By choosing terminal sets $\hat{\mathcal{X}}_{i\in\mathcal{V}}^f$ sufficiently small, also Ass. 7.1 is satisfied.

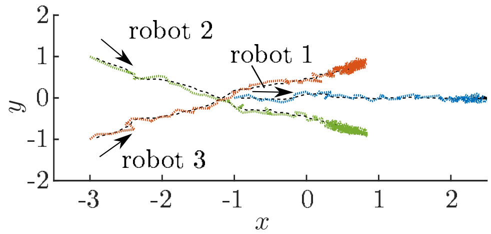
At last, we determine initially feasible reference trajectories (step 4). Therefore, observe that $c_{ij}(\mathbf{x}_i, \mathbf{x}_j) = ||\mathbf{x}_i^{\mathrm{pos}} - \mathbf{x}_j^{\mathrm{pos}}|| - d^{\max}$ is Lipschitz continuous with $C = 1$ for all $i, j \in \mathcal{V}$. Then following the discussion on Lipschitz continuous constraints in Sec. 4.1, step 4, we can rewrite (21c) for all $i, j \in \mathcal{V}$ more conservatively as

$$c_{ij}(\hat{\mathbf{x}}_i, \hat{\mathbf{x}}_j) = ||\hat{\mathbf{x}}_i^{\mathrm{pos}} - \hat{\mathbf{x}}_j^{\mathrm{pos}}|| - d^{\max} \leqslant -\nu_{c_{ij}} \qquad (40)$$

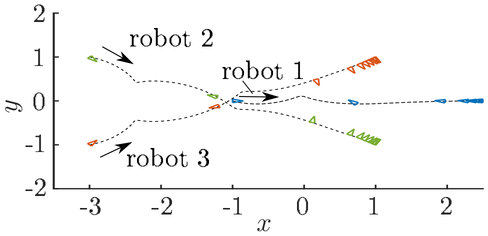
where $\nu_{c_{ij}} = 2(\sqrt{2}\,\bar{c}_i + \bar{p}_i^{\mathrm{pos}})$ with $\bar{p}_i^{\mathrm{pos}} := (\bar{p}_{i,1}^2 + \bar{p}_{i,2}^2)^{0.5} = 0.1636$. By solving an optimization subject to (21), we compute initially feasible reference trajectories x_i^{ref}, $i \in \mathcal{V}$, where (21c) is implemented as (40). To conclude the controller design, we implement the local optimization problems (15)-(16) where consistency constraint (16c) is implemented as a box constraint using the inner approximation $\bar{\mathcal{C}}_i = \{x \in \mathbb{R}^3 \,|\, ||[x_1, x_2]||_\infty \leqslant \bar{c}_i\}$ of $\widehat{\mathcal{C}}_i$. Reference trajectories are updated at every time-step by Algorithm 1 which ensures the satisfaction of Ass. 8 (cf. Prop. 12). Thereby, the satisfaction of all assumptions is ensured by the controller design.

5.2 *Simulation Results*

The three mobile robots start in the initial formation $\mathbf{x}_{0,1} = [-1, 0, 0]^T$, $\mathbf{x}_{0,2} = [-3, 1, 7\pi/4]^T$, $\mathbf{x}_{0,3} = [-3, -1, \pi/4]^T$, and move to the target formation $\xi_1 = [\xi_{11}, 0, \pi]^T$, $\xi_2 = [1, -1, \pi/4]^T$, $\xi_3 = [1, 1, 7\pi/4]^T$ where $\xi_{11} \in \{1.5, 2.0, 2.5, 3.0\}$. Observe that for increasing ξ_{11}, the inter-robot distances in the target formation increase. The prediction time is chosen as $T = 12s$ and the prediction horizon as $N = 36$.



(a) Trajectories of the disturbed system with $w_i \in \mathcal{W}_i$. The black dashed line denotes $\hat{\mathbf{x}}_i[k|k]$, the colored dotted lines the actual trajectories.



(b) Trajectories of the undisturbed system, i.e., $w_i \equiv 0$. The markers denote the robots' orientation. Nominal and actual trajectories coincide.

Fig. 2. Trajectories of robots for $\xi_{11} = 2.5$.

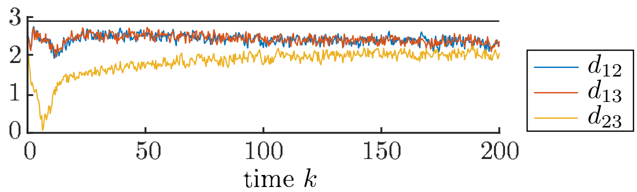


Fig. 3. Actual inter-robot distance $d_{ij} = ||\hat{\mathbf{x}}_i^{\mathrm{pos}} - \hat{\mathbf{x}}_j^{\mathrm{pos}}||$ for $\xi_{11} = 3.0$. The black line denotes $d^{\max} = 2.9$.

For $\xi_{11} = 2.5$, the resulting trajectories are depicted in Fig. 2. Fig. 2(a) shows how the actual state trajectories $\mathbf{x}_i[k]$ oscillate around the nominal trajectories $\hat{\mathbf{x}}_i[k|k]$ due to the disturbances. For comparison, Fig. 2(b) shows the state trajectories in the absence of disturbances; for a detailed discussion of this case, we refer to [23]. Fig. 3 shows that the actual inter-robot distances satisfy the coupled state constraint (35).

5.3 *Performance Analysis*

In order to evaluate the performance of the proposed algorithm with respect to computation time and actual cost, we compare it with two other robust DMPC algorithms: (1) Algorithm 2 with fixed reference trajectories

$$x_i^{\mathrm{ref}}[\kappa|k+1] := \hat{x}_i[\kappa|k] \quad \text{for } \kappa = k+1, \ldots, k+N.$$

This choice of reference trajectories corresponds to the choice in [17]. (2) Sequential DMPC [14] which is based on [22, Sec. 2] and [15]. The DMPC controllers are implemented using Casadi [44], Ipopt and Matlab; simulations are performed on an Intel Core i5-10310U, 16GB RAM.

The simulation results for the relative actual costs and computational times are summarized in Tables 1 and 2. For robot i, the actual cost is computed over the simulated time interval as $J_i^a = \sum_{\kappa=0}^{T_{\mathrm{sim}}} ||\hat{\mathbf{x}}_i[\kappa] - \xi_i||_{\boldsymbol{Q}_i} +$

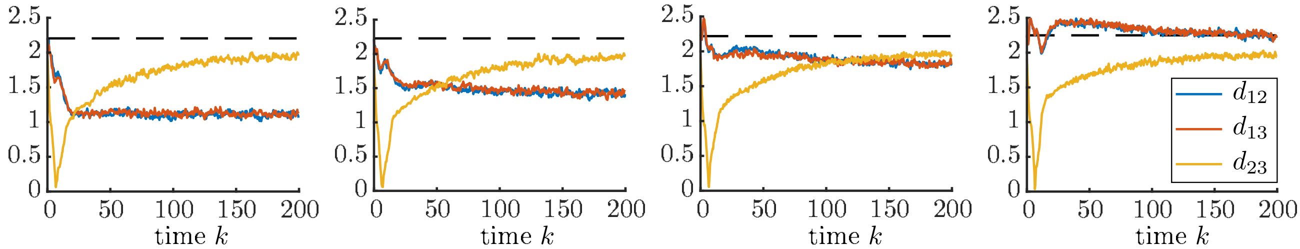


Fig. 4. Nominal inter-robot distances. Distance $\hat{d}_{ij}[k|k] = ||\hat{\mathbf{x}}_i^{\text{pos}}[k|k] - \hat{\mathbf{x}}_j^{\text{pos}}[k|k]||$ denotes the nominal distance between robot i and j; the dashed line indicates $d^{\max} - \nu_{c_{ij}}$.

ξ_{11}	Proposed DMPC	DMPC with fixed reference	Sequential DMPC [14]
1.5	1.00, 1.00, 1.00	0.93, 1.45, 1.45	1.91, 0.93, 0.94
2.0	1.00, 1.00, 1.00	1.02, 1.41, 1.37	1.75, 0.91, 0.90
2.5	1.00, 1.00, 1.00	1.07, 1.12, 1.14	1.37, 0.70, 0.72
3.0	1.00, 1.00, 1.00	1.07, 1.00, 1.02	1.04, 0.61, 0.63

Table 1
Relative actual cost for subsystems 1, 2 and 3.

ξ_{11}	Proposed DMPC	DMPC with fixed reference	Sequential DMPC [14]
1.5	0.0261	0.0255	0.1116
2.0	0.0267	0.0266	0.1155
2.5	0.0254	0.0253	0.1106
3.0	0.0265	0.0262	0.1157

Table 2
Average computational times for calculating control inputs in seconds.

$||\hat{u}_i[\kappa] - u_{\xi_i}||_{\boldsymbol{R}_i}$ where $T_{\text{sim}} = 200$; the i-th entry in each field of Table 1 is the actual cost J_i^a normed with the actual cost J_i^a of the proposed DMPC. The presented numbers are the average from 100 simulations.

In Fig. 4, the nominal inter-robot distances $\hat{d}_{ij} = ||\hat{\mathbf{x}}_i^{\text{pos}} - \hat{\mathbf{x}}_j^{\text{pos}}||$ are depicted for various ξ_{11}. For the pair $(\hat{\mathbf{x}}_i^{\text{pos}}, \hat{\mathbf{x}}_j^{\text{pos}})$, (40) can be rewritten as

$$\hat{d}_{ij} - d^{\max} \leqslant -\nu_{c_{ij}} \tag{41}$$

which is a condition on the nominal inter-robot distance. Intuitively, this means that if the graph of $\hat{d}_{ij}$ exceeds the dashed line in Fig. 4, then (41) is violated. As a consequence, the reference state at the respective time is not changed in order to prevent a potential violation of the coupled state constraints (cf. Sec. 4.2).

This observation can be related to the relative actual costs in Table 1. The closer the nominal distance gets to the dashed line or even exceeds it, the closer is the performance of the proposed DMPC to that of the DMPC with fixed reference (cf. $\xi_{11} = 3.0$). However, if $\hat{d}_{ij}$ does not exceed the dashed line, the performance of the proposed DMPC scheme is significantly improved compared to that of the DMPC with fixed reference where the relative costs are up to 45% higher (cf. $\xi_{11} = 1.5, 2.0$). Compared to sequential DMPC, the performance of the proposed DMPC scheme also tends to be better in these cases. This indicates that the parallelized evaluation of the local optimization problems can be beneficial over a sequential one despite the need of a consistency constraint. In particular, the proposed DMPC computes the control inputs more than 4 times faster (Table 2) than sequential DMPC. This is due to the parallel evaluation of the local optimization problems in the proposed DMPC and the reduced number of constraints. This ratio further improves in favor of the proposed DMPC if more subsystems are added.

5.4 Collision Avoidance Constraints

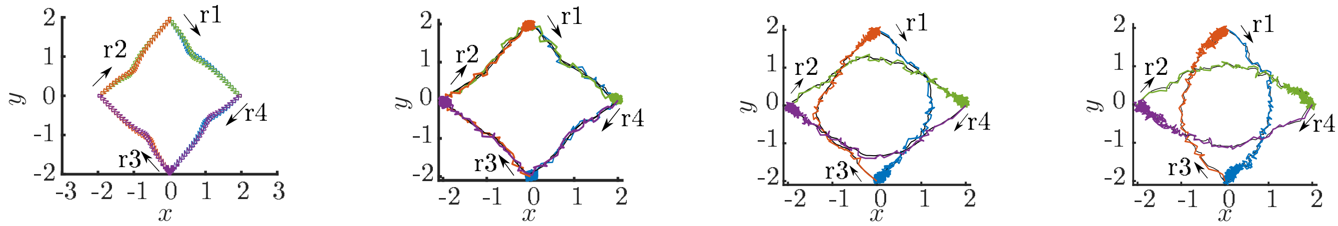
Instead of (35), we now consider the collision avoidance constraint

$$||\mathbf{x}_i^{\text{pos}} - \mathbf{x}_j^{\text{pos}}|| \geqslant d^{\min}, \qquad j \in \mathcal{N}_i := \mathcal{V}\backslash\{i\} \tag{42}$$

where $d^{\min} = 0.5$. Therefore, we replace (40) by

$$c_{ij}(\mathbf{x}_i, \mathbf{x}_j) = d^{\min} - ||\mathbf{x}_i^{\text{pos}} - \mathbf{x}_j^{\text{pos}}|| \leqslant -\nu_{c_{ij}}. \tag{43}$$

We consider four mobile robots governed by (33) as before. The initial formation is $x_{0,1} = [0, 2, -\pi/2]^T$, $x_{0,2} = [-2, 0, 0]^T$, $x_{0,3} = [0, -2, \pi/2]^T$, $x_{0,4} = [2, 0, \pi]^T$; the target formation is $\xi_1 = [0, -2, -3\pi/2]^T$, $\xi_2 = [2, 0, -\pi]^T$, $\xi_3 = [0, 2, -\pi/2]^T$, $\xi_4 = [-2, 0, 0]^T$. Everything else remains unchanged. We initialize the DMPC algorithms with reference trajectories where the robots move clockwise to their target states on the opposite side of the formation as depicted in Fig. 5(a). Because it is generally difficult to determine optimal initially feasible reference trajectories in the presence of concave constraints, it can be beneficial to employ the iterative DMPC scheme as outlined in Rem. 13. As it can be seen from Table 3, the actual cost J_i^a reduces with an increasing number of iterations. Fig. 5 illustrates how the state trajectories of the closed-loop system improve with an increasing number of iterations. Observe that even though constraint (42) is non-convex, the local optimization problems in our proposed approach are only



(a) Initialization (b) Fixed reference trajectories. (c) 1 iteration (Algorithm 2) (d) 4 iterations (algorithm Rem. 13)

Fig. 5. Formation control problem with collision avoidance constraints. Robots 1, 2, 3, 4 are denoted by blue, green, red, violet.

iterations	Robot 1	Robot 2	Robot 3	Robot 4
Fixed ref.	5.2028	5.1523	5.1659	5.1530
1	2.4460	2.4502	2.4832	2.4474
4	1.7965	1.7948	1.7609	1.7633
6	1.4752	1.4779	1.5088	1.5011

Table 3
Actual cost J_i^a ($\times 10^4$) in dependence of the number of iterations, average of 100 simulations. Obtained for $T_{\text{sim}} = 100$.

subject to convex state constraints. That is because the satisfaction of all state constraints is ensured by the consistency constraint which is convex by choice.

6 Conclusion

We presented a robust DMPC algorithm that allows for the parallel evaluation of the local optimization problems in the presence of coupled state constraints while it admits to alter and improve already established reference trajectories. For the case of dynamically decoupled systems subject to coupled constraints, we thereby provide a novel DMPC scheme that allows for a faster distributed control input computation compared to sequential DMPC schemes. Theoretical guarantees on recursive feasibility and robust asymptotic convergence are provided. Moreover, we briefly commented on an iterative extension of the algorithm. In the end, we demonstrated the algorithm's applicability and compared its performance to other DMPC algorithms.

References

[1] H. Chen and F. Allgöwer, "A quasi-infinite horizon nonlinear model predictive control scheme with guaranteed stability," *Automatica*, vol. 34, no. 10, pp. 1205 – 1217, 1998.

[2] G. De Nicolao, L. Magni, and R. Scattolini, "Stabilizing receding-horizon control of nonlinear time-varying systems," *IEEE Transactions on Automatic Control*, vol. 43, no. 7, pp. 1030–1036, 1998.

[3] L. Grüne and J. Pannek, *Nonlinear Model Predictive Control Theory and Algorithms*. Communications and Control Engineering, Springer, London, 1st. ed., 2011.

[4] D. Mayne, J. Rawlings, C. Rao, and P. Scokaert, "Constrained model predictive control: Stability and optimality," *Automatica*, vol. 36, no. 6, pp. 789 – 814, 2000.

[5] D. Q. Mayne, "Model predictive control: Recent developments and future promise," *Automatica*, vol. 50, no. 12, pp. 2967–2986, 2014.

[6] E. Camponogara, D. Jia, B. Krogh, and S. Talukdar, "Distributed model predictive control," *IEEE Control Systems Magazine*, vol. 22, no. 1, pp. 44–52, 2002.

[7] J. R. D. Frejo and E. F. Camacho, "Global versus local mpc algorithms in freeway traffic control with ramp metering and variable speed limits," *IEEE Transactions on Intelligent Transportation Systems*, vol. 13, no. 4, pp. 1556–1565, 2012.

[8] R. M. Hermans, A. Jokić, M. Lazar, A. Alessio, P. P. van den Bosch, I. A. Hiskens, and A. Bemporad, "Assessment of non-centralised model predictive control techniques for electrical power networks," *International Journal of Control*, vol. 85, no. 8, pp. 1162–1177, 2012.

[9] M. A. Müller and F. Allgöwer, "Economic and distributed model predictive control: Recent developments in optimization-based control," *SICE Journal of Control, Measurement, and System Integration*, vol. 10, no. 2, pp. 39 – 52, 2017.

[10] R. Negenborn and J. Maestre, "Distributed model predictive control: An overview and roadmap of future research opportunities," *IEEE Control Systems Magazine*, vol. 34, no. 4, pp. 87–97, 2014.

[11] R. Negenborn and J. Maestre, eds., *Distributed Model Predictive Control Made Easy*, vol. 69 of *Intelligent Systems, Control and Automation: Science and Engineering*. Dordrecht: Springer Netherlands, 2014 ed., 2014.

[12] 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.

[13] 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.

[14] A. Nikou and D. V. Dimarogonas, "Decentralized tube-based model predictive control of uncertain nonlinear multiagent systems," *International Journal of Robust and Nonlinear Control*, vol. 29, no. 10, pp. 2799–2818, 2019.

[15] A. Richards and J. P. How, "Robust distributed model predictive control," *International Journal of Control*, vol. 80, no. 9, pp. 1517–1531, 2007.

[16] 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.

[17] 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.

[18] M. Farina, G. Betti, and R. Scattolini, "Distributed predictive control of continuous-time systems," *Systems & Control Letters*, pp. 32 – 40, 2014.

[19] S. Riverso, M. Farina, and G. Ferrari-Trecate, "Plug-and-play decentralized model predictive control," in *2012 IEEE 51st IEEE Conference on Decision and Control (CDC)*, pp. 4193 – 4198, IEEE, 2012.

[20] 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.

[21] S. Riverso, M. Farina, and G. Ferrari-Trecate, "Plug-and-play model predictive control based on robust control invariant sets," *Automatica*, vol. 50, no. 8, pp. 2179 – 2186, 2014.

[22] 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.

[23] A. Wiltz, F. Chen, and D. V. Dimarogonas, "A consistency constraint-based approach to coupled state constraints in distributed model predictive control," in *2022 IEEE 61st Conference on Decision and Control (CDC)*, pp. 3959–3964, 2022.

[24] S. Rakovic, P. Grieder, M. Kvasnica, D. Mayne, and M. Morari, "Computation of invariant sets for piecewise affine discrete time systems subject to bounded disturbances," in *2004 43rd IEEE Conference on Decision and Control (CDC)*, vol. 2, pp. 1418–1423, 2004.

[25] D. Mayne, M. Seron, and S. Raković, "Robust model predictive control of constrained linear systems with bounded disturbances," *Automatica*, vol. 41, no. 2, pp. 219 – 224, 2005.

[26] S. Raković and D. Mayne, "A simple tube controller for efficient robust model predictive control of constrained linear discrete-time systems subject to bounded disturbances," *IFAC Proceedings Volumes*, vol. 38, no. 1, pp. 241 – 246, 2005. 16th IFAC World Congress.

[27] S. V. Rakovic, A. R. Teel, D. Q. Mayne, and A. Astolfi, "Simple robust control invariant tubes for some classes of nonlinear discrete time systems," in *Proceedings of the 45th IEEE Conference on Decision and Control*, pp. 6397–6402, 2006.

[28] J. B. Rawlings and D. Q. Mayne, *Model predictive control: theory and design.* Madison, Wis.: Nob Hill Publ., 1. printing ed., 2009.

[29] D. Q. Mayne, E. C. Kerrigan, E. J. van Wyk, and P. Falugi, "Tube-based robust nonlinear model predictive control," *International Journal of Robust and Nonlinear Control*, vol. 21, no. 11, pp. 1341–1353, 2011.

[30] S. Yu, C. Maier, H. Chen, and F. Allgöwer, "Tube mpc scheme based on robust control invariant set with application to lipschitz nonlinear systems," *Systems & Control Letters*, vol. 62, no. 2, pp. 194 – 200, 2013.

[31] J. Köhler, R. Soloperto, M. A. Müller, and F. Allgöwer, "A computationally efficient robust model predictive control framework for uncertain nonlinear systems," *IEEE Transactions on Automatic Control*, vol. 66, no. 2, pp. 794–801, 2021.

[32] T. Berger, D. Dennstädt, A. Ilchmann, and K. Worthmann, "Funnel model predictive control for nonlinear systems with relative degree one," *SIAM Journal on Control and Optimization*, vol. 60, no. 6, pp. 3358–3383, 2022.

[33] R. Schneider, *Convex bodies : the Brunn-Minkowski theory.* Encyclopedia of mathematics and its applications, 2nd ed., 2014.

[34] R. E. Kalman and J. E. Bertram, "Control system analysis and design via the "second method" of lyapunov: Ii–discrete-time systems," *J. Basic Eng*, vol. 82, pp. 394–400, June 1960.

[35] W. B. Dunbar, "Distributed receding horizon control of dynamically coupled nonlinear systems," *IEEE Transactions on Automatic Control*, vol. 52, no. 7, pp. 1249–1263, 2007.

[36] A. Falsone, K. Margellos, S. Garatti, and M. Prandini, "Dual decomposition for multi-agent distributed optimization with coupling constraints," *Automatica*, vol. 84, pp. 149–158, 2017.

[37] M. Herceg, M. Kvasnica, C. Jones, and M. Morari, "Multi-Parametric Toolbox 3.0," in *Proc. of the European Control Conference*, (Zürich, Switzerland), pp. 502–510, 2013. http://control.ee.ethz.ch/ mpt.

[38] B. Legat, R. Deits, G. Goretkin, T. Koolen, J. Huchette, D. Oyama, and M. Forets, "Juliapolyhedra/polyhedra.jl: v0.6.16," June 2021.

[39] "Caltech Control and Dynamical Systems", "polytope 0.2.3," Nov. 2020.

[40] A. Wiltz, "Distributed control for spatio-temporally constrained systems," 2023. Licentiate Thesis.

[41] A. Bono, G. Fedele, and G. Franzè, "A swarm-based distributed model predictive control scheme for autonomous vehicle formations in uncertain environments," *IEEE Transactions on Cybernetics*, pp. 1–11, 2021.

[42] I. B. Hagen, D. K. M. Kufoalor, E. F. Brekke, and T. A. Johansen, "Mpc-based collision avoidance strategy for existing marine vessel guidance systems," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 7618–7623, 2018.

[43] J. Löfberg, "Yalmip : A toolbox for modeling and optimization in matlab," in *In Proceedings of the CACSD Conference*, (Taipei, Taiwan), 2004.

[44] J. A. E. Andersson, J. Gillis, G. Horn, J. B. Rawlings, and M. Diehl, "CasADi – A software framework for nonlinear optimization and optimal control," *Mathematical Programming Computation*, vol. 11, no. 1, pp. 1–36, 2019.