A Consistency Constraint-Based Approach to Coupled State Constraints in Distributed Model Predictive Control

Adrian Wiltz, Fei Chen, Dimos V. Dimarogonas

Abstract— **In this paper, we present a distributed model predictive control (DMPC) scheme for dynamically decoupled systems which are subject to state constraints, coupling state constraints and input constraints. In the proposed control scheme, neighbor-to-neighbor communication suffices and all subsystems solve their local optimization problem in parallel. The approach relies on consistency constraints which define a neighborhood around each subsystem's reference trajectory where the state of the respective subsystem is guaranteed to stay in. Reference trajectories and consistency constraints are known to neighboring subsystems. Contrary to other relevant approaches, the reference trajectories are improved iteratively. Besides, the presented approach allows the formulation of convex optimization problems even in the presence of non-convex state constraints. The algorithm's effectiveness is demonstrated with a simulation.**

I. INTRODUCTION

Since the publication of the first distributed model predictive control (DMPC) schemes [3], their development became a thriving branch in the research on model predictive control (MPC). The motivation behind the development of DMPC is that centralized MPC [16] becomes computationally intractable for large-scale systems, and in the case of spatially distributed systems a reliable communication with a central control-unit is difficult to realize [9], [12].

In [17], 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 [17]. As remarked in [19], 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 [20]. Many available DMPC schemes follow in their proofs of recursive feasibility and asymptotic stability an argumentation similar to [4] for continuous-time systems, and [5] for discrete-time systems.

Especially the distribution of MPC problems subject to coupled state constraints turned out to be complicated [13], and most available DMPC schemes that are capable of handling them cannot avoid a sequential scheme [14], [21], [22], [23]. However, sequential schemes have the drawback that the computation of the control input of all subsystems becomes very time-consuming for highly connected networks when each subsystem has a large number of neighbors. A notable exception that does not rely on a sequential scheme can be found in [8], 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. However, [8] does not allow to modify the reference trajectory once it is established which restricts the possibility to optimize the system's performance significantly.

In this paper, we overcome this limitation. Taking inspiration from [8], we employ consistency constraints in our approach. However, we allow to vary and further improve an already established reference trajectory at each time-step. For guaranteeing recursive feasibility, we check some conditions and correspondingly update the reference trajectories afterwards. All computations are carried out in a distributed fashion. The proposed algorithm only requires neighbor-to-neighbor communication. Interestingly, the usage of consistency constraints allows us to formulate the local optimization problems as convex problems even in the presence of non-convex state and coupled state constraints. In contrast to [8], we formulate the algorithm for dynamically decoupled systems.

The remainder is structured as follows: In Section II, we present the partitioned system and the control objective. In Section III, we define the local optimization problems, the proposed DMPC algorithm, and derive guarantees. In Section IV, the algorithm's effectiveness is demonstrated, and in Section V some conclusions are drawn.

Notation: A continuous function $\gamma : \mathbb{R}_+ \to \mathbb{R}_+$ is a $\mathcal{K}_\infty$ function if $\gamma(0) = 0$, it is strictly increasing and $\gamma(t) \to \infty$ as $t \to \infty$. If the domain of a trajectory $x[\kappa]$ with $\kappa = a,\ldots,b$, $a, b \in \mathbb{N}$, 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 short-hand $x_{i\in\mathcal{V}}$ is equivalent to $\{x_i\}_{i\in\mathcal{V}}$ where $\mathcal{V}$ is an index set. The Minkowski sum is denoted by $\oplus$. Finally, $\mathbf{0}, \mathbf{1}$ denote vectors of all zeros or ones, and for $x \in \mathbb{R}^n$ and $P \in \mathbb{R}^{n\times n}$, we define the weighted norm $||x||_P = x^T P x$.

This work was supported by the ERC Consolidator Grant LEAFHOUND, the Swedish Foundation for Strategic Research (SSF) COIN, the Swedish Research Council (VR) and the Knut and Alice Wallenberg Foundation.

The authors are with the Division of Decision and Control Systems, KTH Royal Institute of Technology, SE-100 44 Stockholm, Sweden {wiltz,fchen,dimos}@kth.se.

II. Problem Formulation

A. 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

$$x_i[k+1] = f_i(x_i[k], u_i[k]), \qquad x_i[k_0] = x_{0,i} \tag{1}$$

where $x_i \in \mathbb{R}^{n_i}$ and $u_i \in \mathbb{R}^{m_i}$. The dynamics of the overall system are denoted by

$$x[k+1] = f(x[k], u[k]), \qquad x[k_0] = x_0 \tag{2}$$

with stack vectors $x = [x_1^T, \dots, x_{|\mathcal{V}|}^T]^T$, $u = [u_1^T, \dots, u_{|\mathcal{V}|}^T]^T$, and $x_0 = [x_{0,1}^T, \dots, x_{0,|\mathcal{V}|}^T]^T$. All subsystems $i \in \mathcal{V}$ are subject to state constraints $x_i \in \mathcal{X}_i \subseteq \mathbb{R}^{n_i}$ and input constraints

$$u_i \in \mathcal{U}_i \subseteq \mathbb{R}^{m_i}. \tag{3}$$

Moreover, some of the subsystems are coupled to each other by coupled state constraints. For subsystems i and j, if there exists a coupled state constraint $x_i[k] \in \mathcal{X}_{ij}(x_j[k])$, then we call subsystem j a *neighbor* of subsystem i and write $j \in \mathcal{N}_i$, where $\mathcal{N}_i$ is the set of all neighboring subsystems of i. We define the state constraints via inequalities, namely

$$h_i(x_i) \leq \mathbf{0} \tag{4a}$$

$$c_{ij}(x_i, x_j) \leq \mathbf{0}, \qquad j \in \mathcal{N}_i \tag{4b}$$

where h_i, c_{ij} are continuous; $h_i : \mathbb{R}^{n_i} \to \mathbb{R}^{r_i}$ defines the state constraint with r_i as the number of inequalities defining the state constraint of subsystem i; $c_{ij} : \mathbb{R}^{n_i} \times \mathbb{R}^{n_j} \to \mathbb{R}^{s_i}$ defines the coupled state constraints with s_i as the number of inequalities defining the coupled state constraints that couple subsystem i with j. The state constraint sets are defined as

$$\mathcal{X}_i := \{x_i \in \mathbb{R}^{n_i} \,|\, h_i(x_i) \leq \mathbf{0}\}$$

$$\mathcal{X}_{ij}(x_j) := \{x_i \in \mathbb{R}^{n_i} \,|\, c_{ij}(x_i, x_j) \leq \mathbf{0}\}, \qquad j \in \mathcal{N}_i$$

where $\mathcal{X}_{ij} : \mathbb{R}^{n_j} \to \mathcal{P}(\mathbb{R}^{n_i})$ is a set valued function and $\mathcal{P}(\mathbb{R}^{n_i})$ denotes the power set of $\mathbb{R}^{n_i}$.

A state x_i of subsystem i is called *infeasible* if (4) is not satisfied. Besides, we assume that neighboring subsystems that are coupled by state constraints both respect these constraints. In particular, we assume that for all subsystems $i \in \mathcal{V}$ and their neighbors $j \in \mathcal{N}_i$ we have

$$c_{ij}(x_i, x_j) = c_{ji}(x_j, x_i).$$

Otherwise, this would result in an asymmetry in the subsystems capabilities which is beyond the scope of this paper.

B. 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 $\mathcal{E}$ is the set of edges defined as $\mathcal{E} := \{(i,j) \,|\, j \in \mathcal{N}_i\}$. We assume that $\mathcal{E}$ defines the communication links among the subsystems $\mathcal{V}$ and hence that neighboring subsystems can communicate with each other.

C. Control Objective

Let $\xi_i = f_i(\xi_i, u_{\xi_i})$ be a steady state of subsystem i for a constant input u_{ξ_i} and denote by $\xi = [\xi_1^T, \dots, \xi_{|\mathcal{V}|}^T]^T$ the stack vector of all steady states. The control objective is to steer subsystems $i \in \mathcal{V}$ to target states $\{\xi_i\}_{i \in \mathcal{V}}$ which satisfy $\xi \in \Xi := \{\xi | \xi_i \in \mathcal{X}_i,\ \xi_i \in \mathcal{X}_{ij}(\xi_j),\ u_{\xi_i} \in \mathcal{U}_i,\ \ \forall i \in \mathcal{V}, \forall j \in \mathcal{N}_i\}$.

III. Main Results

A. Local Optimization Problems

In the proposed DMPC scheme, a subsystem i predicts a state-trajectory $x_i[\kappa|k]$ for $\kappa = k, \dots, k+N$ and a corresponding input-trajectory $u_i[\kappa|k]$ for $\kappa = k, \dots, k+N-1$ based on dynamics (1) at every time-step k minimizing a cost function $J_i(x_i[\cdot|k], u_i[\cdot|k])$ where N denotes the prediction horizon. Input trajectories $u_{i\in\mathcal{V}}[\cdot|k]$ are determined such that they satisfy input constraints (3), and state trajectories $x_{i\in\mathcal{V}}[\cdot|k]$ such that they start in $x_{i\in\mathcal{V}}[k]$ measured at time k and it holds $x_i[k|k] = x_i[k]$, $i \in \mathcal{V}$. The satisfaction of state constraints (4) is ensured by *consistency constraints*

$$x_i[\kappa|k] \in x_i^{\text{ref}}[\kappa|k] \oplus \mathcal{C}_i \tag{5}$$

where $i \in \mathcal{V}$, $\kappa = k, \dots, k+N$ and $\mathcal{C}_i \subseteq \mathbb{R}^{n_i}$ is a closed neighborhood of the origin. Thereby, subsystem i guarantees to its neighbors $j \in \mathcal{N}_i$ that its predicted state trajectory $x_i[\kappa|k]$ always stays in a neighborhood $\mathcal{C}_i$ of reference states $x_i^{\text{ref}}[\kappa|k]$ for $\kappa = k, \dots, k+N$. Reference states $x_i^{\text{ref}}[\kappa|k]$ are determined at every time-step and communicated to all neighbors $j \in \mathcal{N}_i$. Contrary to other DMPC approaches employing consistency constraints [6], [7], [8], here the reference trajectories $x_{i\in\mathcal{V}}^{\text{ref}}[\cdot|k]$ need not to satisfy dynamics (2).

The local optimization problem of subsystem i at time-step k is given as

$$J_i^*(x_i[k]) = \min_{u_i[\cdot|k]} J_i(x_i[\cdot|k], u_i[\cdot|k]) \tag{6}$$

where

$$J_i(x_i[\cdot|k], u_i[\cdot|k]) = \sum_{\kappa=k}^{k+N-1} l_i(x_i[\kappa|k], u_i[\kappa|k]) + J_i^f(x_i[k+N|k])$$

$$l_i(x_i[\kappa|k], u_i[\kappa|k]) = ||x_i[\kappa|k] - \xi_i||_{Q_i} + ||u_i[\kappa|k] - u_{\xi_i}||_{R_i}$$

with stage-cost-function l_i, terminal cost-function J_i^f, and quadratic positive-definite matrices Q_i, R_i. By $x_i^*[\cdot|k]$ and $u_i^*[\cdot|k]$, we denote those trajectories that minimize J_i. The local optimality criterion (6) is subject to

$$\begin{aligned} x_i[k|k] &= x_i[k] \\ x_i[\kappa+1|k] &= f_i(x_i[\kappa|k], u_i[\kappa|k]) \end{aligned} \tag{7a}$$

$$x_i[\kappa|k] \in x_i^{\text{ref}}[\kappa|k] \oplus \mathcal{C}_i \tag{7b}$$

$$u_i[\kappa|k] \in \mathcal{U}_i \tag{7c}$$

$$x_i[k+N|k] \in \mathcal{X}_i^f \tag{7d}$$

for $\kappa = k, \dots, k+N-1$ and with terminal set $\mathcal{X}_i^f$. The optimal control input applied by subsystem i at time k given current state $x_i[k]$ is $\mu_i(x_i[k]) := u_i^*[k|k]$.

In order to ensure recursive feasibility and that the consistency constraint do not admit states that result in a violation of state constraints (4), the reference states $x_i^{\text{ref}}[\kappa|k]$, $\kappa = k, \dots, k+N$, are updated in every time-step k. We impose the following assumption on reference states and consistency constraint sets.

Assumption 1. For $\kappa = k, \dots, k+N-1$ and $k \geq k_0$, reference states $x_i^{\text{ref}}[\kappa|k]$ are chosen

(A1.1) such that

$$h_i(x_i) \leq \mathbf{0} \quad \forall x_i \in x_i^{\text{ref}}[\kappa|k] \oplus \mathcal{C}_i \tag{8}$$

and for all $j \in \mathcal{N}_i$

$$c_{ij}(x_i, x_j) \leq \mathbf{0} \quad \forall x_i \in x_i^{\text{ref}}[\kappa|k] \oplus \mathcal{C}_i, \quad \forall x_j \in x_j^{\text{ref}}[\kappa|k] \oplus \mathcal{C}_j; \tag{9}$$

(A1.2) and if $k > k_0$ additionally such that

$$x_i^*[\kappa|k-1] \in x_i^{\text{ref}}[\kappa|k] \oplus \mathcal{C}_i. \tag{10}$$

Whereas (A1.1) ensures the satisfaction of (4), (A1.2) leads to recursive feasibility of the local optimization problems. In Section III-D, we show how to satisfy Assumption 1.

Moreover, we impose the following standard assumptions on terminal sets $\mathcal{X}_i^f$ and terminal cost J_i^f [8], [10] in a slightly modified way.

Assumption 2. For the terminal sets $\mathcal{X}_i^f$, the terminal costs $J_i^f : \mathcal{X}_i^f \to \mathbb{R}_{\geq 0}$ with $i \in \mathcal{V}$, and a state-feedback controller $k_i^{\text{aux}}(x_i)$, we assume that

(A2.1) all $\mathcal{X}_i^f$, $i \in \mathcal{V}$, are feasible, i.e., there exist $\alpha_i \in \mathbb{R}_{>1}$, $i \in \mathcal{V}$, such that $\alpha_i \mathcal{X}_i^f \subseteq \mathcal{X}_i$ and $c_{ij}(x_i, x_j) \leq \mathbf{0}$ $\forall x_i \in \alpha_i \mathcal{X}_i^f, \forall x_j \in \alpha_j \mathcal{X}_j^f$ for all $j \in \mathcal{N}_i$;

(A2.2) $k_i^{\text{aux}} : \mathcal{X}_i^f \to \mathcal{U}_i$;

(A2.3) $f_i(x_i, k_i^{\text{aux}}(x_i)) \in \mathcal{X}_i^f$;

(A2.4) $J_i^f(f(x_i, k_i^{\text{aux}}(x_i))) + l_i(x_i, k_i^{\text{aux}}(x_i)) \leq J_i^f(x_i)$.

By (A2.1), it is ensured that state constraints (4) are satisfied for any states in the terminal regions $\mathcal{X}_{i\in\mathcal{V}}^f$ and a neighborhood around them. We suggest to determine J_i^f and $\mathcal{X}_i^f$ as outlined in [10, Remark 5.15].

B. Distributed MPC Algorithm

First, initial state trajectories $x_i^{\text{init}}[\kappa|k_0]$, $\kappa = k_0, \ldots, k_0 + N$, and input trajectories $u_i^{\text{init}}[\kappa|k_0]$, $\kappa = k_0, \ldots, k_0 + N - 1$, must be determined for all $i \in \mathcal{V}$ such that

$$x_i^{\text{init}}[k_0|k_0] = x_{0,i}$$
$$x_i^{\text{init}}[\kappa+1|k_0] = f_i(x_i^{\text{init}}[\kappa|k_0], u_i^{\text{init}}[\kappa|k_0]) \tag{11a}$$
$$h_i(x_i^{\text{init}}[\kappa|k_0]) \leq -\beta_i \mathbf{1} \tag{11b}$$
$$c_{ij}(x_i^{\text{init}}[\kappa|k_0], x_j^{\text{init}}[\kappa|k_0]) \leq -\beta_i \mathbf{1} \tag{11c}$$
$$u_i^{\text{init}}[\kappa|k_0] \in \mathcal{U}_i \tag{11d}$$
$$x_i^{\text{init}}[k_0 + N|k_0] \in \mathcal{X}_i^f \tag{11e}$$

where $\beta_i \in \mathbb{R}_{>0}$. Then, we define initial reference trajectories as $x_i^{\text{ref}}[\cdot|k_0] := x_i^{\text{init}}[\cdot|k_0]$ for all $i \in \mathcal{V}$. Note, that (11b) and (11c) imply (4) but are stricter. Similarly, (A2.1) ensures that for each state in $\mathcal{X}_i^f$ there exists a neighborhood whose states satisfy (4) as well. Hence, there always exist sufficiently small sets $\mathcal{C}_i$ for all $i \in \mathcal{V}$ such that (8) and (9) hold with $x_i^{\text{ref}}[\kappa|k_0] = x_i^{\text{init}}[\kappa|k_0]$, and such that

$$\mathcal{X}_i^f \oplus \mathcal{C}_i \subseteq \alpha_i \mathcal{X}_i^f. \tag{12}$$

We call trajectories $x_{i\in\mathcal{V}}^{\text{init}}[\cdot|k_0], u_{i\in\mathcal{V}}^{\text{init}}[\cdot|k_0]$ that satisfy (11) *initially feasible*. We define $\mathcal{X}_N^0$ as the set of states x_0 for which initially feasible trajectories exist when the prediction horizon is N. For computing the sets $\mathcal{C}_i$, $i \in \mathcal{V}$, we suggest to start with an initial guess of $\mathcal{C}_i$ denoted by $\hat{\mathcal{C}}_i$, and check if (8), (9) and (12) are satisfied. If these hold, enlarge $\hat{\mathcal{C}}_i$ by multiplication with a scalar $\rho_i > 1$; otherwise, shrink $\hat{\mathcal{C}}_i$ by multiplication with $1/\rho_i$. Then, we choose $\mathcal{C}_i = \rho_i^{\iota_i^{\max}} \hat{\mathcal{C}}_i$ where $\iota_i^{\max} \in \mathbb{Z}$ is the largest integer such that (8), (9) and (12) are still satisfied. If $\hat{\mathcal{C}}_i$ is defined as the intersection of half-spaces (polytopes), and h_i and c_{ij} are linear, then (8), (9) and (12) can be efficiently evaluated using common libraries for computations with polyhedra, e.g., MPT3 (Matlab) [11], Polyhedra (Julia) [15] and Polytope (Python) [2]. In (8) and (9), $\mathcal{C}_i$ and $\mathcal{C}_j$ can be replaced by outer approximations which often results in more conservative conditions, but allows for an easier evaluation. An example is given in Section IV.

The entire distributed MPC algorithm, which comprises initialization steps 1 and the DMPC routine 2, is as follows:

1.1. Set $k = k_0$ and determine J_i^f, $\mathcal{X}_i^f$ and k_i^{aux} for all $i \in \mathcal{V}$.
1.2. Compute $x_i^{\text{init}}[\kappa|k_0]$ for $\kappa = k_0, \ldots, k_0 + N$ and $u_i^{\text{init}}[\kappa|k_0]$ for $\kappa = k_0, \ldots, k_0 + N - 1$ for all $i \in \mathcal{V}$ such that (11) holds. All subsystems $i \in \mathcal{V}$ communicate $x_i^{\text{init}}[\cdot|k_0], u_i^{\text{init}}[\cdot|k_0]$ to their neighbors $j \in \mathcal{N}_i$.
1.3. For all $i \in \mathcal{V}$, set $x_i^{\text{ref}}[\kappa|k_0] = x_i^{\text{init}}[\kappa|k_0]$ for $\kappa = k_0, \ldots, k_0 + N - 1$ and compute $\mathcal{C}_i \subseteq \mathbb{R}^{n_i}$ as a closed neighborhood of the origin such that (8), (9) and (12) hold. All subsystems $i \in \mathcal{V}$ communicate $x_i^{\text{ref}}[\cdot|k_0]$ and $\mathcal{C}_i$ to their neighbors $j \in \mathcal{N}_i$.
2.1. All $i \in \mathcal{V}$ measure $x_i[k]$, solve their local optimization problem (6) subject to (7) in parallel and thereby determine $x_i^*[\kappa|k]$ for $\kappa = k, \ldots, k + N$ and $u_i^*[\kappa|k]$ for $\kappa = k, \ldots, k + N - 1$. All $i \in \mathcal{V}$ communicate $x_i^*[\cdot|k]$ to their neighbors $j \in \mathcal{N}_i$.
2.2. All $i \in \mathcal{V}$ apply $\mu_i(x_i[k]) := u_i^*[k|k]$.
2.3. All $i \in \mathcal{V}$ determine reference states $x_i^{\text{ref}}[\kappa|k+1]$ for $\kappa = k + 1, \ldots, k + N$ such that Assumption 1 holds (see Section III-D). All $i \in \mathcal{V}$ communicate $x_i^{\text{ref}}[\cdot|k+1]$ to their neighbors $j \in \mathcal{N}_i$.
2.4. Set $k \leftarrow k + 1$ and go to step 2.1.

C. Guarantees and Properties of the DMPC Algorithm

At first, we observe that the proposed DMPC algorithm ensures the satisfaction of the constraints introduced in Section II-A which each subsystem has to satisfy.

Lemma 1. Let Assumption 1 and 2 hold. For all $i \in \mathcal{V}$, let $x_i^*[\kappa|k]$, $\kappa = k, \ldots, k+N$, and $u_i^*[\kappa|k]$, $\kappa = k, \ldots, k+N-1$, be the solution to local optimization problem (6) with constraints (7). Then, constraints (3) and (4) are satisfied by $x_i^*[\cdot|k]$ and $u_i^*[\cdot|k]$.

Proof. This result is straightforward: (7c) implies input constraint (3), consistency constraints (7b) imply state constraints (4) for $\kappa = k, \ldots, k + N - 1$ due to (A1.1), and (7d) and (A2.1) together imply (4) for $\kappa = k + N$. ☐

Next, we prove recursive feasibility of the proposed DMPC algorithm and show that the states $x_{i\in\mathcal{V}}$ asymptotically converge to their target states $\xi_{i\in\mathcal{V}}$.

Theorem 2. For $i \in \mathcal{V}$, let $x_i^{\text{init}}[\kappa|k_0]$ for $\kappa = k_0, \ldots, k_0+N$ and $u_i^{\text{init}}[\kappa|k_0]$ for $\kappa = k_0, \ldots, k_0+N-1$ be initially feasible trajectories that satisfy (11). Besides, let Assumptions 1 and 2 hold. Then, the DMPC algorithm comprising steps 1.1 to 2.4 is recursively feasible, ξ is an asymptotically stable equilibrium of the closed-loop system

$$x[k+1] = f(x[k], \mu_N(x[k])),$$

on $\mathcal{X}_N^0$ and x asymptotically converges to ξ.

Proof. In a first step, we prove recursive feasibility and thereafter asymptotic stability and asymptotic convergence.

Recursive Feasibility: In order to show recursive feasibility, we have to show that for all $k \geq k_0$, there exist candidate trajectories $x_i^{\mathrm{c}}[\kappa|k]$ for $\kappa = k, \ldots, k+N$ and $u_i^{\mathrm{c}}[\kappa|k]$ for $\kappa = k, \ldots, k+N-1$ that satisfy (7). Thereby it is ensured, that there always exist feasible solutions to the local optimization problems (6)-(7).

First, consider $k = k_0$ and choose candidate trajectories

$$x_i^{\mathrm{c}}[\kappa|k_0] = x_i^{\mathrm{init}}[\kappa|k_0] \quad \text{for } \kappa = k_0, \ldots, k_0+N,$$
$$u_i^{\mathrm{c}}[\kappa|k_0] = u_i^{\mathrm{init}}[\kappa|k_0] \quad \text{for } \kappa = k_0, \ldots, k_0+N-1$$

for all $i \in \mathcal{V}$ where $x_i^{\mathrm{init}}[\cdot|k_0], u_i^{\mathrm{init}}[\cdot|k_0]$ denote the initially feasible trajectories determined in step 1.2. Since $x_i^{\mathrm{init}}[\cdot|k_0], u_i^{\mathrm{init}}[\cdot|k_0]$ satisfy (11a), it trivially follows that $x_i^{\mathrm{c}}[\cdot|k_0], u_i^{\mathrm{c}}[\cdot|k_0]$ satisfy (7a) for all $i \in \mathcal{V}$. Because $x_i^{\mathrm{ref}}[\cdot|k_0] = x_i^{\mathrm{init}}[\cdot|k_0] = x_i^{\mathrm{c}}[\cdot|k_0]$ and $\mathcal{C}_i$ is a closed neighborhood of the origin, it also follows that $x_i^{\mathrm{c}}[\kappa|k_0]$ satisfies (7b). The satisfaction of (7c)-(7d) trivially follows from (11d)-(11e). Thereby, we have shown that there exists at least one feasible solution to the optimization problem (6)-(7) for $k = k_0$.

In a next step, we show the existence of feasible solutions for $k > k_0$. Therefore, consider the candidate trajectories

$$x_i^{\mathrm{c}}[\kappa|k] = \begin{cases} x_i^*[\kappa|k-1] & \text{for } \kappa = k, \ldots, k+N-1 \\ f_i(x_i^*[\kappa-1|k-1], k_i^{\mathrm{aux}}(x_i^*[\kappa-1|k-1])) \\ & \text{for } \kappa = k+N \end{cases} \tag{14a}$$

$$u_i^{\mathrm{c}}[\kappa|k] = \begin{cases} u_i^*[\kappa|k-1] & \text{for } \kappa = k, \ldots, k+N-2 \\ k_i^{\mathrm{aux}}(x_i[\kappa|k-1]) & \text{for } \kappa = k+N-1 \end{cases} \tag{14b}$$

for all $i \in \mathcal{V}$. Since

$$\begin{aligned} x_i^{\mathrm{c}}[k|k] &= x_i^*[k|k-1] \\ &= f_i(x_i^*[k-1|k-1], u_i^*[k-1|k-1]) \\ &= f_i(x_i[k-1], u_i[k-1]) = x_i[k] \end{aligned}$$

and since $x_i^*[\cdot|k-1], u_i^*[\cdot|k-1]$ satisfy (7a), it follows from (14) that also $x_i^{\mathrm{c}}[\cdot|k], u_i^{\mathrm{c}}[\cdot|k]$ satisfy (7a). Because $x_i^{\mathrm{c}}[\kappa|k] = x_i^*[\kappa|k-1]$ for $\kappa = k, \ldots, k+N-1$ and (A1.2) holds, $x_i^{\mathrm{c}}[\cdot|k]$ satisfies (7b). Besides, as $u_i^{\mathrm{c}}[\kappa|k+1] = u_i^*[\kappa|k]$, (7c) is trivially satisfied for $\kappa = k, \ldots, k+N-2$. Furthermore, $u_i^{\mathrm{c}}[k+N-1|k] = k_i^{\mathrm{aux}}(x_i^*[k+N-1|k-1]) \in \mathcal{U}_i$ holds due to (A2.2) because $x_i^*[k+N-1|k-1] \in \mathcal{X}_i^f$, and the satisfaction of (7c) is shown. At last, because $x_i^{\mathrm{c}}[k+N|k] = f_i(x_i^*[k+N-1|k-1], k_i^{\mathrm{aux}}(x_i^*[k+N-1|k-1]))$ and due to (A2.3), also $x_i^{\mathrm{c}}[k+N|k] \in \mathcal{X}_i^f$ holds and (7d) is satisfied. Thereby, we have shown that there also exists for all $k > k_0$ at least one feasible solution to optimization problem (6)-(7), and we conclude that the algorithm is recursively feasible.

Asymptotic stability and asymptotic convergence (Sketch): Using standard arguments [10], we can derive that $V_i(x_i) := J_i^*(x_i)$ is a Lyapunov function for (1) by showing that

$$\begin{aligned} V_i(x_i[k+1]) - V_i(x_i[k]) &\leq -l_i(x_i[k], \mu_N(x_i[k])) \\ &\leq ||x_i[k] - \xi_i||_{Q_i} = -\gamma_{V_i}(x_i[k]) \end{aligned}$$

where γ_{V_i} is a $\mathcal{K}_\infty$ function. Moreover, we can show that $V(x) := \sum_{i \in \mathcal{V}} V_i(x)$ is a Lyapunov function for the overall system (2) on $\mathcal{X}_N^0$. Then, the asymptotic stability of ξ on $\mathcal{X}_N^0$ and the asymptotic convergence to ξ follows, cf. e.g. [10, Theorem 2.19]. □

D. Determination of Reference States

For $k = k_0$, reference states $x_{i \in \mathcal{V}}^{\mathrm{ref}}[\cdot|k]$ are determined in step 1.3 of the DMPC algorithm such that Assumption 1 holds. For $k > k_0$, we can determine the reference states $x_i^{\mathrm{ref}}[\kappa|k]$ for $\kappa = k, \ldots, k+N-1$ as follows:

1. For $\kappa = k, \ldots, k+N-2$, check if
$$h_i(x_i) \leq \mathbf{0} \qquad \forall x_i \in x_i^*[\kappa|k-1] \oplus \mathcal{C}_i \tag{15}$$
and
$$\begin{aligned} &c_{ij}(x_i, x_j) \leq \mathbf{0} \qquad \forall x_i \in x_i^*[\kappa|k-1] \oplus \mathcal{C}_i, \\ &\forall x_j \in (x_j^*[\kappa|k-1] \oplus \mathcal{C}_j) \cup (x_j^{\mathrm{ref}}[\kappa|k-1] \oplus \mathcal{C}_j) \end{aligned} \tag{16}$$
for all $j \in \mathcal{N}_i$ hold.
2. For $\kappa = k, \ldots, k+N-2$, if (15) and (16) hold for all $j \in \mathcal{N}_i$, then set $x_i^{\mathrm{ref}}[\kappa|k] = x_i^*[\kappa|k-1]$, else set $x_i^{\mathrm{ref}}[\kappa|k] = x_i^{\mathrm{ref}}[\kappa|k-1]$.
3. For $\kappa = k+N-1$, set $x_i^{\mathrm{ref}}[\kappa|k] = x_i^*[\kappa|k-1]$.

Proposition 3. Let Assumption 1 be satisfied at time-step k_0. Then Assumption 1 holds for $k > k_0$, if $x_i^{\mathrm{ref}}[\kappa|k]$ is determined according to steps 1 to 3 above for $\kappa = k, \ldots, k+N-1$ and all $i \in \mathcal{V}$.

Proof. This result follows recursively. To start with, consider $\kappa = k, \ldots, k+N-2$. At first note that the reference states of all subsystems $i \in \mathcal{V}$ are chosen such that $x_i^{\mathrm{ref}}[\kappa|k] \in \{x_i^*[\kappa|k-1], x_i^{\mathrm{ref}}[\kappa|k-1]\}$. In step 1, each subsystem $i \in \mathcal{V}$ checks if $x_i^{\mathrm{ref}}[\kappa|k] = x_i^*[\kappa|k-1]$ is a viable choice such that (A1.1) is fulfilled for any $x_j^{\mathrm{ref}}[\kappa|k] \in \{x_j^*[\kappa|k-1], x_j^{\mathrm{ref}}[\kappa|k-1]\}$. Therefore, (15) and (16) imply (8) and (9), respectively, and (10) trivially follows from $x_i^{\mathrm{ref}}[\kappa|k] = x_i^*[\kappa|k-1]$. Thus, Assumption 1 holds if (15) and (16) hold.

Next, we show that $x_i^{\mathrm{ref}}[\kappa|k] = x_i^{\mathrm{ref}}[\kappa|k-1]$ fulfills Assumption 1 independently of the satisfaction of (15) and (16). As $x_i^{\mathrm{ref}}[\kappa|k-1]$ satisfied (8) at time-step $k-1$, it does so at k since (8) is decoupled. Next, consider any neighbor $j \in \mathcal{N}_i$ of subsystem i. If subsystem j sets $x_j^{\mathrm{ref}}[\kappa|k] = x_j^*[\kappa|k-1]$, then (16) holds according to step 2 and the satisfaction of (10) follows. If however subsystem j sets $x_j^{\mathrm{ref}}[\kappa|k] = x_j^{\mathrm{ref}}[\kappa|k-1]$, the satisfaction of (9) follows recursively from the previous time-step. The initial existence of reference states x_i^{ref} that satisfy (9) is ensured by the assumption that $x_i^{\mathrm{ref}}[\kappa|k_0]$ satisfies Assumption 1. At last, (10) is satisfied because the consistency constraint (7b) constraining the optimization problem (6)-(7) at time-step $k-1$ implies $x_i^*[\kappa|k-1] \in x_i^{\mathrm{ref}}[\kappa|k-1] \oplus \mathcal{C}_i = x_i^{\mathrm{ref}}[\kappa|k] \oplus \mathcal{C}_i$ which is equivalent to (10). Hence, it is shown that Assumption 1 is also fulfilled, even if (15) or (16) do not hold. At last, consider $\kappa = k+N-1$. Since $x_i^*[k+N-1|k-1] \in \mathcal{X}_i^f$ according to (7d), the choice $x_i^{\mathrm{ref}}[\kappa|k] = x_i^*[\kappa|k-1]$ ensures the satisfaction of (A1.1) due to (12) and (A2.1). (A1.2) is trivially satisfied. □

Remark 1. The aforementioned libraries for computations with polyhedra can be also used for the evaluation of conditions (15) and (16). In particular, note that if sets $\mathcal{C}_i$, $i \in \mathcal{V}$, are polytopes, and h_i and c_{ij}, $j \in \mathcal{N}_i, i \in \mathcal{V}$, are linear, then conditions (15) and (16) reduce to simple algebraic inequalities. Besides, observe that $\mathcal{C}_i$ and $\mathcal{C}_j$ in (15) and (16) can be replaced by an outer approximation which results in more conservative conditions but does not change the result of Proposition 3. This is possible because the choice $x_i^{\text{ref}}[\kappa|k] = x_i^{\text{ref}}[\kappa|k-1]$ always ensures the fulfillment of Assumption 1 independently of (15) and (16) as it can be seen from the proof of Proposition 3. Outer approximations can help to simplify conditions (15) and (16) especially if h_i, c_{ij} are nonlinear. We give an example in the next section.

IV. Simulation

We consider three-wheeled omni-directional robots, which behave according to their kinematic model with states $x_i := [p_{i,x}, p_{i,y}, \psi_i]^T$ where $p_{i,x}$ and $p_{i,y}$ denote the position coordinates and ψ_i the orientation of robot i. The position of robot i is denoted by $p := [p_{i,x}, p_{i,y}]^T$. Its dynamics are

$$\dot{x}_i = R(\psi_i)\,(B_i^T)^{-1}\, r_i\, u_i$$

where

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

l_i is the radius of the robot body, r_i is the wheel radius, and $u_i = [u_{i,1}, u_{i,2}, u_{i,3}]^T$ the angular velocity of the wheels.

In the sequel, we consider three robots which shall move from an initial formation x_0 to a target formation ξ, where all robots $i \in \mathcal{V}$ are subject to connectivity constraints

$$||p_i - p_j|| \le d_{\max}, \qquad j \in \mathcal{N}_i := \mathcal{V} \setminus \{i\} \qquad (17)$$

with $d_{\max} = 2.6$, and input constraints $||u_i||_\infty \le 15$ where $||\cdot||_\infty$ denotes the maximum norm. The performance matrices are chosen as $Q_1 = \text{diag}(100, 100, 100)$, $Q_2 = Q_3 = \text{diag}(1, 1, 50)$, $R_1 = \text{diag}(1, 1, 1)$, $R_2 = R_3 = \text{diag}(5, 5, 5)$; the terminal cost functions $J^f_{i\in\mathcal{V}}$ and terminal sets $\mathcal{X}^f_{i\in\mathcal{V}}$ are computed via the algebraic Riccati equation as outlined in [10]. The consistency constraint set is chosen as a box $\mathcal{C}_i = \{\eta \in \mathbb{R}^3 \,|\, ||[\eta_1, \eta_2]||_\infty \le c_i\}$ with $c_i = 0.125$ for all $i \in \mathcal{V}$; since ψ_i is unconstrained, the third coordinate η_3 is unconstrained as well. A suitable outer approximation of $\mathcal{C}_i$ is $\bar{\mathcal{C}}_i := \{\bar{\eta} \in \mathbb{R}^3 \,|\, ||[\bar{\eta}_1, \bar{\eta}_2]|| \le \bar{c}_i\}$ with $\bar{c}_i = \sqrt{2}\, c_i$ where the box constraint on the position coordinates is replaced by a circle that encloses it. By using the outer approximations $\bar{\mathcal{C}}_i$, we rewrite (9) more conservatively as

$$c_{ij}(x_i, x_j) = ||p_i - p_j|| - d_{\max} \le -2\bar{c}_i, \qquad (18)$$

i.e., the satisfaction of (18) implies the satisfaction of (9). Thereby, it is sufficient to check a simple inequality and set valued operations can be avoided. Although (9) cannot be simplified in all cases as much as in this example, a problem dependent outer approximation of $\mathcal{C}_i$ can often still lead to significant simplifications. If terminal sets $\mathcal{X}^f_{i\in\mathcal{V}}$ are chosen sufficiently small such that (18) is satisfied by all $x_i \in \mathcal{X}^f_i$ for all $i \in \mathcal{V}$, then (A2.1) is satisfied. Furthermore, if $\beta_i = 2\bar{c}_i$ in (11c), then $x_i^{\text{ref}}[\cdot|k_0] = x_i^{\text{init}}[\cdot|k_0]$ fulfills also Assumption 1. We compute initially feasible trajectories $x^{\text{init}}_{i\in\mathcal{V}}$ using a sequential MPC approach.

In the scenario considered in the simulation, three robots solve a formation control task. Starting in the initial formation $x_{0,1} = [-1, 0, 0]^T$, $x_{0,2} = [-3, 1, 7\pi/4]^T$, $x_{0,3} = [-3, -1, \pi/4]^T$, the robots 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 $\xi_{11} \in \{2.0, 2.5, 2.75, 3.0\}$. The prediction time is chosen as $T = 12s$ and the prediction horizon as $N = 36$. We discretize the time-continuous dynamics using a 4-th order Runge-Kutta method with a time-step $\Delta t = T/N$.

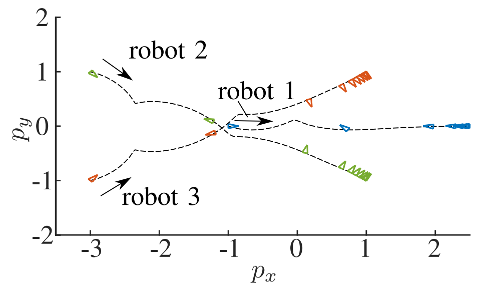


Fig. 1. Trajectories of robots for $\xi_{11} = 2.5$. The markers indicate the robots' orientations.

ξ_{11}	Proposed DMPC	DMPC with fixed reference [8]	Sequential DMPC [18]
2.0	1.00,1.00,1.00	0.94,1.42,1.42	0.87,0.71,0.71
2.5	1.00,1.00,1.00	1.00,1.35,1.32	0.93,0.68,0.67
2.75	1.00,1.00,1.00	1.00,1.18,1.17	0.94,0.60,0.59
3.0	1.00,1.00,1.00	1.01,1.06,1.05	0.93,0.53,0.53

TABLE I

Relative actual cost for subsystems 1, 2 and 3.

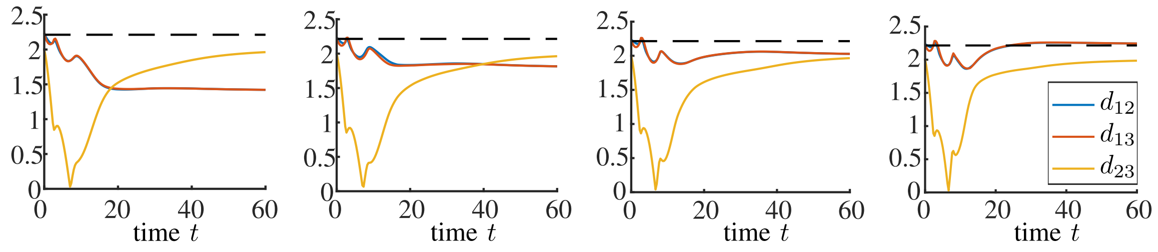
ξ_{11}	Proposed DMPC	DMPC with fixed reference [8]	Sequential DMPC [18]
2.0	0.0234	0.0253	0.0935
2.5	0.0237	0.0259	0.0975
2.75	0.0233	0.0262	0.1060
3.0	0.0224	0.0249	0.1405

TABLE II

Average computional times in seconds.

In order to evaluate the performance of the proposed algorithm with respect to computation time and actual cost, we compare it with two other common DMPC algorithms: (1) DMPC algorithm of [8] with fixed reference trajectories which cannot be updated once they are established; (2) Sequential DMPC as presented in [18, Section 2] which is based on the sequential scheme proposed in [22]. The MPC controllers have been implemented using Casadi [1] and Matlab, the simulations have been performed on an Intel Core i5-10310U with 16GB RAM.

The resulting trajectories of the robots are depicted in Figure 1, the inter-robot distances in Figure 2. From Figure 2, it can be seen that distance constraints (17) are clearly satisfied. However, simulations using an MPC algorithm that does not take the coupled state constraints (17) into account resulted for $\xi_{11} \in \{2.5, 2.75, 3.0\}$ in a violation of (17) which emphasizes the importance of explicitly taking coupled state constraints into account. If the inter-robot distances exceed the dashed-line in Figure 2, then (16) is not satisfied anymore. As a result, the reference state at the respective time is not changed in order to prevent a potential violation of the coupled state constraints (cf. Section III-D). This causes some conservativeness which also can be seen from the relative actual costs listed in Table I. For robot i, the actual cost is computed over the simulated time interval as $J_i^a = \sum_{\kappa=0}^{60} ||x_i[\kappa] - \xi_i||_{Q_i} + ||u_i[\kappa] - u_{\xi_i}||_{R_i}$; the i-th entry in each field of Table I is the actual cost J_i^a obtained with



(a) $\xi_{11} = 2.0$ (b) $\xi_{11} = 2.5$ (c) $\xi_{11} = 2.75$ (d) $\xi_{11} = 3.0$

Fig. 2. Inter robot distances. Distance d_{ij} denotes the distance between robot i and j.

the respective DMPC algorithm normed with the actual cost J_i^a obtained using the proposed DMPC. Comparing Figure 2 and Table I shows, that the more often a subsystem is close to a violation of one of its coupled state constraints and (16) is violated (in this example: the more often the distance curve exceeds the dashed line), the closer is its performance to the performance of fixed reference DMPC (cf. $\xi_{11} = 3.0$). However, if the distance curves do not exceed the dashed line, or only for small time intervals, the proposed algorithm results in a notably improved performance compared to fixed reference DMPC where the actual costs are 17-42% higher. Most importantly, however, the proposed DMPC computes the control inputs more than 4-6 times faster (Table II) than the sequential DMPC. This is due to the parallel evaluation of the local optimization problems in the proposed DMPC and the reduced number of constraints because all state constraints are incorporated into the consistency constraint. This ratio further improves in favor of the proposed DMPC if more subsystems are added because computations of neighboring subsystems must be carried out sequentially.

V. Conclusion

We presented a DMPC algorithm that allows for the parallel evaluation of the local optimization problems in the presence of coupled state constraints while it admits to iteratively alter and improve already established reference trajectories at every time-step. 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. Guarantees are provided when applying it to the nominal, i.e., undisturbed and known dynamics. In a next step, we plan to extend the presented basic algorithm such that model uncertainties and disturbances can be handled.

References

[1] 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.

[2] "Caltech Control and Dynamical Systems", "polytope 0.2.3," Nov. 2020. [Online]. Available: https://github.com/tulip-control/polytope

[3] 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.

[4] 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.

[5] 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.

[6] 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.

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

[8] 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.

[9] 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.

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

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

[12] 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.

[13] 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.

[14] 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.

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

[16] 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.

[17] 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.

[18] 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.

[19] 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.

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

[21] 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.

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

[23] 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.