Distributed Learning Model Predictive Control for Linear Systems*

Yvonne R. Stürz[1], Edward L. Zhu[1], Ugo Rosolia[2], Karl H. Johansson[3], Francesco Borrelli[1]

***Abstract*— This paper presents a distributed learning model predictive control (DLMPC) scheme for distributed linear time invariant systems with coupled dynamics and state constraints. The proposed solution method is based on an online distributed optimization scheme with nearest-neighbor communication. If the control task is iterative and data from previous feasible iterations are available, local data are exploited by the subsystems in order to construct the local terminal set and terminal cost, which guarantee recursive feasibility and asymptotic stability, as well as performance improvement over iterations. In case a first feasible trajectory is difficult to obtain, or the task is non-iterative, we further propose an algorithm that efficiently explores the state-space and generates the data required for the construction of the terminal cost and terminal constraint in the MPC problem in a safe and distributed way. In contrast to other distributed MPC schemes which use structured positive invariant sets, the proposed approach involves a control invariant set as the terminal set, on which we do not impose any distributed structure. The proposed iterative scheme converges to the global optimal solution of the underlying infinite horizon optimal control problem under mild conditions. Numerical experiments demonstrate the effectiveness of the proposed DLMPC scheme.**

I. Introduction

Complex systems composed of multiple subsystems are present in many control applications. The large scale and spatial distribution of these systems often make the control by a centralized unit intractable due to limitations in computation and communication. Research has therefore focused on proposing design schemes for local controllers which compute control actions for the individual subsystems based on only local information in decentralized schemes, and on communicated information from neighboring subsystems in distributed control schemes. One line of research has focused on exploiting the interconnection structure of the system in order to design interconnected controllers based on a convex reformulation involving linear matrix inequalities in a scalable way [1]. If constraints need to be accounted for, distributed model predictive control (DMPC) techniques can be employed. They can mainly be categorized into non-cooperative, such as tube-based [2], and cooperative schemes [3], [4], [5], [6], [7]. The latter often involve distributed optimization techniques [8] where the subsystems communicate local information and agree on a solution, thus solving the optimization problem cooperatively.

The main challenge in DMPC schemes is to enable distributed computation by decomposing the optimization problem into subproblems for the individual subsystems. Most of the DMPC approaches in the literature therefore impose the distributed structure of the system on the terminal set and cost function of the MPC problem [2], [4], [5], [9], [10], [11]. In particular, they first design a structured terminal controller and cost based on Lyapunov stability and then design structured positive invariant sets under this terminal controller, satisfying the constraints. Two aspects in these schemes can lead to conservatism: (1) Imposing structure on the terminal controllers and terminal sets, and (2) computing positive invariant sets for one specific choice of terminal controller which is fixed in the design phase, lead to a possibly small inner approximation of the maximal control invariant set. In order to mitigate the conservatism introduced by the imposed structure, some works have proposed to adapt the terminal sets based on the states of the subsystems in operation [10], [11], [5]. In [6], [7], terminal sets also computed online within the MPC problem.

Recent research has focused on learning-based and data-driven DMPC schemes, [12], [13], [14], [15], [16]. In [17], a data-driven MPC scheme, Learning MPC (LMPC), was introduced, where previously seen data are exploited in order to construct the terminal components of the MPC problem. In [18] this framework was extended to uncertain systems, and it was shown how the LMPC scheme can be used to iteratively enlarge the domain of the policy.

In this paper, we propose a distributed LMPC (DLMPC) scheme. The contributions of the paper are the following:

- We present a novel DLMPC scheme for linear systems able to handle coupled dynamics, coupled state constraints and coupled cost functions. The main improvement w.r.t. existing DMPC approaches is fully distributed computations without imposing any structure on the terminal cost function or constraint set. This is achieved by exploiting previously seen local data by the individual subsystems in order to build local data driven terminal sets and terminal cost functions. A consensus on specific parameters in the construction of the local costs and constraints is achieved by distributed optimization which guarantees that the local terminal sets are a control invariant set and the sum of the

[1]The authors are with the Model Predictive Control Laboratory, University of California, Berkeley, CA 94709, USA. Email address: y.stuerz@berkeley.edu

[2]The author is with the Department of Mechanical and Civil Engineering, California Institute of Technology, Pasadena, CA 91125 USA

[3]The author is with the School of Electrical Engineering and Computer Science, KTH Royal Institute of Technology, Stockholm, Sweden.

*This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Sklodowska-Curie grant agreement No. 846421. This work was also sponsored by the Office of Naval Research. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the Office of Naval Research or the US government.

local terminal cost functions is a Lyapunov function for the global system. This can considerably reduce conservatism w.r.t. DMPC schemes that rely on finding a positive invariant terminal set under a fixed structured stabilizing terminal controller.
- For iterative control tasks, given a first feasible trajectory, the proposed scheme provides recursive feasibility and asymptotic stability. Furthermore, we prove that the proposed DLMPC has a non-increasing control performance over iterations and, under mild conditions, converges to the global centralized optimal solution.
- For non-iterative control tasks, or if an initial feasible trajectory is difficult to obtain, we further present an algorithm that by iteratively performing the proposed DLMPC scheme with changing starting conditions leads to an enlargement of the domain of the DLMPC policy. This can be used to safely explore the state space and to generate the required data in a sample efficient and distributed way.

The paper is structured as follows. Section II introduces the model of the distributed systems and the control task considered. Section III provides a brief review of the LMPC in [19]. A decomposed formulation of the LMPC according to the distributed system structure is presented in Section IV. The fully distributed solution is proposed in Section V. Section VI provides numerical experiments before Section VII concludes the paper.

Notation: Let $\mathbb{R}$ denote the set of real numbers. $\mathbb{N}$ and $\mathbb{N}_+$ denote the set of non-negative and positive natural numbers. We denote the transpose of a vector $v \in \mathbb{R}^n$ as $v^\top$, and its Euclidean norm as $\|v\|$. The matrix $M = \mathrm{diag}(M_1, ..., M_m)$ is the block-diagonal matrix with submatrices M_i on its diagonal. The symbol $\succcurlyeq$ is used to indicate elementwise inequality. The identitiy matrix of dimension n is denoted as I_n and the vector of all ones is denoted as $\mathbf{1}$.

II. Problem Formulation

In this section, we present the model of the distributed systems considered in this paper, and then state the control problem formulation.

A. Dynamically Coupled Constrained Linear Systems

We consider the discrete-time linear time-invariant system with dynamics given by

$$x_{t+1} = Ax_t + Bu_t, \tag{1}$$

where $x_t \in \mathbb{R}^n$ and $u_t \in \mathbb{R}^m$ are the system state and input at time $t \in \mathbb{N}$. The system matrices A and B are assumed to be known. The system states and inputs are subject to linear constraints

$$x_t \in \mathcal{X}, \qquad u_t \in \mathcal{U}, \tag{2}$$

which are formulated as

$$Gx_t \leq g, \qquad Lu_t \leq l, \tag{3}$$

with G, L, g, and l given matrices and vectors, respectively.

We consider systems in (1) which have a structure that admits a decomposition into subsystems $\mathcal{N} = \{1, ..., M\}$ which may be coupled in their state dynamics. The state of the ith subsystem is $x_{i,t} \in \mathbb{R}^{n_i}$, and we assume that the ith input $u_{i,t} \in \mathbb{R}^{m_i}$ affects only the i-th state. Thus, the system states and inputs are partitioned as

$$x_t = \left[x_{1,t}^\top \ \dots \ x_{M,t}^\top\right]^\top, \quad u_t = \left[u_{1,t}^\top \ \dots \ u_{M,t}^\top\right]^\top. \tag{4}$$

For each subsystem $i \in \mathcal{N}$, we define the set of neighboring subsystems $\mathcal{N}_i \subseteq \mathcal{N}$ which contains all those subsystems that are coupled to subsystem i over the dynamics, constraints or cost. We define the state vector $x_{\mathcal{N}_i,t} \in \mathbb{R}^{n_{\mathcal{N}_i}}$ containing the local states of subsystem i and its neighboring subsystems in $\mathcal{N}_i$, which can be expressed as $x_{\mathcal{N}_i,t} = X_{\mathcal{N}_i} x_t$, with $X_{\mathcal{N}_i}$ being a projection matrix, i.e., a binary matrix $X_{\mathcal{N}_i} \in \{0,1\}^{n_{\mathcal{N}_i} \times n}$. Similarly, the projection matrices $X_i \in \{0,1\}^{n_i \times n}$ and $U_i \in \{0,1\}^{m_i \times m}$, are defined such that $x_{i,t} = X_i x_t$ and $u_{i,t} = U_i u_t$. The dynamics of subsystem i is then reformulated as

$$x_{i,t+1} = A_{\mathcal{N}_i} x_{\mathcal{N}_i,t} + B_i u_{i,t}, \quad \forall i \in \mathcal{N}, \tag{5}$$

with

$$A_{\mathcal{N}_i} = X_i A X_{\mathcal{N}_i}^\top, \quad B_i = X_i B U_i^\top. \tag{6}$$

The local state and input constraints are defined as

$$\begin{aligned} &x_{\mathcal{N}_i,t} \in \mathcal{X}_{\mathcal{N}_i} = \{x_{\mathcal{N}_i,t} \in \mathbb{R}^{n_{\mathcal{N}_i}} : G_{\mathcal{N}_i} x_{\mathcal{N}_i} \leq g_{\mathcal{N}_i}\}, \\ &u_{i,t} \in \mathcal{U}_i = \{u_{i,t} \in \mathbb{R}^{m_i} : L_i u_i \leq l_i\}, \end{aligned} \tag{7}$$

with $L_i = U_i L U_i^\top$ and $l_i = U_i l$, and $G_{\mathcal{N}_i} = X_{\mathcal{N}_i} G X_{\mathcal{N}_i}^\top$ and $g_{\mathcal{N}_i} = X_{\mathcal{N}_i} g$.

B. Control Problem Formulation

Let us consider system (1). We are given an iterative task, where the trajectories of the subsystems start at the same initial states at each iteration. We will discuss the case of non-iterative tasks in Section V-C. In the following, we denote the iteration by a superscript q and the initial state at iteration q by

$$x_{i,0}^q = x_{i,S}, \quad \forall i \in \mathcal{N}, \tag{8}$$

where the overall initial state $x_0^q = x_S$ is defined as a stacked vector similar to (4).

The goal is to solve the following infinite horizon optimal control problem (IHOCP) at each iteration

$$\begin{aligned} J_{0\to\infty}^*(x_S) = \min_{u_0,u_1,\dots} \quad & \sum_{t=0}^{\infty} h(x_t, u_t) \\ \text{s.t.} \quad & x_{t+1} = Ax_t + Bu_t, \ \forall t \geq 0 \\ & x_t \in \mathcal{X}, \ \forall t \geq 0 \\ & u_t \in \mathcal{U}, \ \forall t \geq 0 \\ & x_0 = x_S. \end{aligned} \tag{9}$$

In the following, we consider problems that involve decomposable stage costs in (9), i.e., where $h(x_t, u_t)$ is given as a sum of local stage costs $h_i(x_{\mathcal{N}_i,t}, u_{i,t})$ as

$$h(x_t, u_t) = \sum_{i=1}^{M} h_i(x_{\mathcal{N}_i,t}, u_{i,t}). \tag{10}$$

We assume that the local stage costs $h_i(\cdot,\cdot)$ are continuous, jointly convex and satisfy

$$\begin{cases} h_i(x_{\mathcal{N}_i,F},0)=0, \\ h_i(x^q_{\mathcal{N}_i,t},u^q_{i,t})\geq 0, \quad \forall\ x^q_{\mathcal{N}_i,t}\in\mathbb{R}^{n_{\mathcal{N}_i}}\backslash\{x_{\mathcal{N}_i,F}\}, \\ \qquad\qquad\qquad\qquad \forall\ u^q_{i,t}\in\mathbb{R}^{m_i}\backslash\{0\}, \end{cases} \tag{11}$$

where the final state x_F is a feasible equilibrium for system (1) under no input, i.e., $Ax_F=x_F$.

Remark 1: While the local stage costs $h_i(x_{\mathcal{N}_i,t},u_{i,t})$ can account for coupling between the subsystems, this formulation includes the special case of completely separable cost functions with local stage costs given as $h_i(x_{i,t},u_{i,t})$.

Remark 2: A specific choice of the stage cost $h(x_t,u_t)$ can be the quadratic function

$$h(x_t,u_t)=x_t^\top Qx_t+u_t^\top Ru_t,$$

with positive semi-definite and positive definite weighting matrices $Q\in\mathbb{R}^{n\times n}$ and $R\in\mathbb{R}^{m\times m}$, respectively. In this case, the local stage costs are given by

$$h_i(x_{\mathcal{N}_i,t},u_{i,t})=x_{\mathcal{N}_i,t}^\top Q_{\mathcal{N}_i}x_{\mathcal{N}_i,t}+u_{i,t}^\top R_iu_{i,t},$$

with $Q_{\mathcal{N}_i}$ and R_i such that the global weighting matrices Q and R are given by $Q=\sum_{i\in\mathcal{N}}X_{\mathcal{N}_i}^\top Q_{\mathcal{N}_i}X_{\mathcal{N}_i}$ and $R=\sum_{i\in\mathcal{N}}U_i^\top R_iU_i$. A completely separable quadratic stage cost is then defined as

$$h_i(x_{i,t},u_{i,t})=x_{i,t}^\top Q_ix_{i,t}+u_{i,t}^\top R_iu_{i,t}, \tag{12}$$

with Q_i such that $Q=\sum_{i\in\mathcal{N}}X_i^\top Q_iX_i$ and R_i as before.

III. Background on LMPC

We review the LMPC problem formulation for the global system in (1) from [19]. For this, we define the vectors that collect all inputs applied to system (1) and its resulting states for all time steps t of iteration q as

$$\begin{aligned} \mathbf{u}^q&=[u_0^{q\top},u_1^{q\top},...,u_t^{q\top},...]^\top, \\ \mathbf{x}^q&=[x_0^{q\top},x_1^{q\top},...,x_t^{q\top},...]^\top. \end{aligned} \tag{13}$$

A. Convex Safe Set

In order to guarantee stability of MPC laws, an N-step controllable set to a control invariant set can be used. Computing such a set is usually numerically challenging or even intractable for nonlinear systems or large scale distributed systems. To alleviate this problem, we will as [17] exploit previously seen trajectories that successfully completed the iterative task. Since they represent a subset of the maximal stabilizable set, the sampled safe set $\mathcal{SS}^q$ is defined over the realized trajectories of the system from previous iterations

$$\mathcal{SS}^q=\left\{\bigcup_{l\in T^q}\bigcup_{t=0}^{\infty}x_t^l,\right\}, \tag{14}$$

where T^q collects all iteration indices from previous successful iterations, i.e., which were feasible and converged to x_F, defined as

$$T^q=\left\{l\in[0,q]:\lim_{t\to\infty}x_t^l=x_F\right\}. \tag{15}$$

Because of the convexity of the constraints $\mathcal{X}$ and $\mathcal{U}$, any convex combination of the elements in the safe set $\mathcal{SS}^q$ is again a control invariant set for system (1), i.e., for any element in the convex safe set

$$\begin{aligned} &\mathcal{CS}^q=\mathrm{conv}(\mathcal{SS}^q) \\ &=\left\{\sum_{l\in T^q}\sum_{t=0}^{\infty}\alpha_t^lx_t^l:\ \alpha_t^l\geq 0,\ \sum_{l\in T^q}\sum_{t=0}^{\infty}\alpha_t^l=1,\ x_t^l\in\mathcal{SS}^q\right\}, \end{aligned} \tag{16}$$

there exists a sequence of control inputs that steers the system (1) to x_F [20]. If all previous successful trajectories are taken into account, then it holds that the sets are growing over the iterations, i.e., $T^{q-1}\subseteq T^q$ and therefore

$$\mathcal{CS}^{q-1}\subseteq\mathcal{CS}^q. \tag{17}$$

B. Terminal Cost

For the qth realized trajectory $\mathbf{x}^q$ and associated input sequence $\mathbf{u}^q$ in (13), the cost-to-go from time t onwards is given by

$$J^q_{t\to\infty}(x_t^q)=\sum_{k=t}^{\infty}h(x_k^q,u_k^q). \tag{18}$$

The performance of the qth trajectory is defined as the cost from time $t=0$, i.e.,

$$J^q_{0\to\infty}(x_0^q)=\sum_{t=0}^{\infty}h(x_t^q,u_t^q). \tag{19}$$

The barycentric function [21] is used as the terminal cost in the LMPC for linear systems in [17]. It is defined as

$$\begin{aligned} V^{q,*}(x)=\min_{\boldsymbol{\alpha}^q}\quad & \sum_{l=0}^{q}\sum_{t=0}^{\infty}\alpha_t^lJ^l_{t\to\infty}(x_t^l) \\ \text{s.t.}\quad & \sum_{l=0}^{q}\sum_{t=0}^{\infty}\alpha_t^l=1 \\ & \sum_{l=0}^{q}\sum_{t=0}^{\infty}\alpha_t^lx_t^l=x, \\ & \alpha_t^l\geq 0,\quad \forall t\in\mathbb{N}, \end{aligned} \tag{20}$$

with x_t^l being the realized state at time t of the lth iteration, and where $\boldsymbol{\alpha}^q$ comprises all $\alpha_t^l,\forall l\in\{0,...,q\},\forall t\in\mathbb{N}$. The function $V^{q,*}$ thus assigns to every point in the convex safe set the corresponding convex combination of minimum costs-to-go along the previous trajectories in the safe set.

Remark 3: In practical applications, the iterations will have a finite time duration. For simplicity, we adopt the infinite time formulation in this paper.

An LMPC [17] for a centralized linear system then solves at each time step t the following finite horizon optimal

control problem (FHOCP),

$$
\begin{aligned}
&J^{\mathrm{LMPC},q}_{t\to t+N}(x^q_t) = \\
&\min_{\mathbf{x}_{t,N},\mathbf{u}_{t,N-1},\boldsymbol{\alpha}^{q-1}} \left[\sum_{k=t}^{t+N-1} h(x_{k|t},u_{k|t}) + V^{q-1,*}(x_{k+N|t})\right] \\
&\quad\text{s.t.} \quad x_{k+1|t} = Ax_{k|t} + Bu_{k|t}, \\
&\qquad\quad x_{k|t} \in \mathcal{X}, \\
&\qquad\quad u_{k|t} \in \mathcal{U},\ k = t,...,t{+}N{-}1 \\
&\qquad\quad x_{t|t} = x^q_t, \\
&\qquad\quad x_{t+N|t} \in \mathcal{CS}^{q-1},
\end{aligned} \tag{21}
$$

with

$$
\begin{aligned}
\mathbf{x}_{t,N} &= [x^\top_{t|t},...,x^\top_{t+N|t}]^\top, \\
\mathbf{u}_{t,N-1} &= [u^\top_{t|t},...,u^\top_{t+N-1|t}]^\top.
\end{aligned} \tag{22}
$$

Let us denote the optimal solution to (21) by

$$
\begin{aligned}
\mathbf{x}^*_{t,N} &= [x^{*\top}_{t|t},...,x^{*\top}_{t+N|t}]^\top, \\
\mathbf{u}^*_{t,N-1} &= [u^{*\top}_{t|t},...,u^{*\top}_{t+N-1|t}]^\top.
\end{aligned} \tag{23}
$$

At time t, the first input is applied to the system, i.e., $u^q_t = u^{*q}_{t|t}$, and the problem (21) is solved again for the next time step in a receding horizon fashion.

Under the assumption that at iteration $q = 1$ the convex safe set is non-empty, i.e., $\mathcal{CS}^{q-1} = \mathcal{CS}^0 \neq \emptyset$, recursive and iterative feasibility, asymptotic stability and non-decreasing performance over the iterations are proved in [19].

IV. DLMPC

In the following, we present the problem formulation of DLMPC, which extends the LMPC approach to distributed systems. Let us consider the coupled constrained linear distributed system from (1). We define the vectors that collect all inputs applied to subsystem i in (5) and its resulting states for all time steps t of iteration q as

$$
\begin{aligned}
\mathbf{u}^q_i &= [u^{q\top}_{i,0}, u^{q\top}_{i,1},..., u^{q\top}_{i,t},...]^\top, \\
\mathbf{x}^q_i &= [x^{q\top}_{i,0}, x^{q\top}_{i,1},..., x^{q\top}_{i,t},...]^\top.
\end{aligned} \tag{24}
$$

We further define the local sampled safe sets for subsystems $i \in \mathcal{N}$ over the realized trajectories of the subsystem from all successful previous iterations up to q as

$$
\mathcal{SS}^q_i = \left\{\bigcup_{l\in T^q}\bigcup_{t=0}^{\infty} x^l_{i,t},\right\}, \tag{25}
$$

with T^q as defined before for (14). Moreover, we note that we can decompose the safe set from (16) into the following local convex safe sets

$$
\begin{aligned}
\mathcal{CS}^q_i = \Bigg\{ &\sum_{l\in T^q}\sum_{t=0}^{\infty} \alpha^l_{i,t} x^l_{i,t} : \\
&\alpha^l_{i,t} \geq 0,\ \sum_{l\in T^q}\sum_{t=0}^{\infty} \alpha^l_{i,t} = 1,\ x^l_{i,t} \in \mathcal{SS}^q_i \Bigg\},
\end{aligned} \tag{26}
$$

where the coefficients $\alpha^l_{i,t}, \forall l \in \{0,...,q-1\}, \forall t \geq 0$ will be optimized over in problems (28) and (30).

We note the following relation of the convex safe set $\mathcal{CS}^q$ in (16) and the local convex safe sets $\mathcal{CS}^q_i$ in (26), which will be important for the decomposition of the problem in (21):

$$
\begin{aligned}
&x = [x^\top_1,...,x^\top_M]^\top \in \mathcal{CS}^q \iff \\
&x_i \in \mathcal{CS}^q_i, \quad \boldsymbol{\alpha}^q_i = \boldsymbol{\alpha}^q_j, \forall i \neq j, i,j \in \mathcal{N},
\end{aligned} \tag{27}
$$

with $\boldsymbol{\alpha}^q_i$ comprising $\alpha^l_{i,t}, \forall l \in \{0,...,q-1\}, \forall t \geq 0$ in (26).

Based on the assumption before that the global system is decomposable into M coupled subsystems, the global LMPC problem in (21) can equivalently be decomposed into the following subproblems

$$
\begin{aligned}
&J^{\mathrm{LMPC},q}_{i,t\to t+N}(x^q_{i,t}) = \\
&\min_{\substack{\mathbf{x}_{\mathcal{N}_i,t,N},\\ \mathbf{u}_{i,t,N-1},\\ \boldsymbol{\alpha}^{q-1}_i}} \left[\sum_{k=t}^{t+N-1} h_i(x_{\mathcal{N}_i,k|t}, u_{i,k|t}) + V^{q-1,*}_i(x_{i,t+N|t})\right] \\
&\quad\text{s.t.} \quad x_{i,k+1|t} = A_{\mathcal{N}_i} x_{\mathcal{N}_i,k|t} + B_i u_{i,k|t}, \\
&\qquad\quad x_{\mathcal{N}_i,k|t} \in \mathcal{X}_{\mathcal{N}_i}, \\
&\qquad\quad u_{i,k|t} \in \mathcal{U}_i,\ k = t,...,t{+}N{-}1 \\
&\qquad\quad x_{\mathcal{N}_i,t|t} = x_{\mathcal{N}_i,t}, \\
&\qquad\quad x_{i,t+N|t} \in \mathcal{CS}^q_i,
\end{aligned} \tag{28}
$$

with

$$
\begin{aligned}
\mathbf{x}_{\mathcal{N}_i,t,N} &= [x^\top_{\mathcal{N}_i,t|t},...,x^\top_{\mathcal{N}_i,t+N-1|t}]^\top, \\
\mathbf{u}_{i,t,N-1} &= [u^\top_{i,t|t},...,u^\top_{i,t+N-1|t}]^\top,
\end{aligned} \tag{29}
$$

with $\boldsymbol{\alpha}^{q-1}_i$ comprising $\alpha^l_{i,t}, \forall l \in \{0,...,q-1\}, \forall t \in \mathbb{N}$, and with $V^{q-1,*}_i(x_{i,k+N|t})$ being defined as in (20), but with $J^l_{t\to\infty}(x^l_t)$ replaced by

$$
J^l_{i,t\to\infty}(x^l_{i,t}) = \sum_{k=t}^{\infty} h_i(x^l_{\mathcal{N}_i,k}, u^l_{i,k}).
$$

In order to guarantee that the decomposed problem in (28) is an exact reformulation of the global problem in (21), i.e., to guarantee that they have the same solutions, the following consensus constraints need to be introduced

$$
\begin{aligned}
\boldsymbol{\alpha}^{q-1}_i &= \boldsymbol{\alpha}^{q-1}_j, && \forall i,j \in \mathcal{N}, i \neq j, \\
\mathbf{x}_{\mathcal{N}_i,t,N} &= X_{\mathcal{N}_i}\,\mathbf{x}_{t,N}, && \forall i \in \mathcal{N},\ t \geq 0,
\end{aligned} \tag{30}
$$

with $\mathbf{x}_{t,N}$ the planned state trajectory of the global system as defined in (22). The consensus constraint in the first line of (30) ensures the condition in (27), and the one in the second line ensures that overlapping parts of state variables from neighboring subsystems in $\mathbf{x}_{\mathcal{N}_i,t,N}$, i.e., variables of different subsystems that have the same physical meaning, are the same. The local FHOCPs in (28) are solved in a receding horizon fashion, i.e., the first local inputs $u^q_{i,t} = u^{*q}_{i,t|t}$ are applied to the subsystems at time t. The next section presents a distributed solution method to solve the subproblems (28).

V. Distributed Synthesis for DLMPC

In this section, we present a distributed solution method for the local decomposed subproblems in (28) coupled over the consensus constraints in (30). Various distributed optimization algorithms can be employed [8]. We propose a

distributed solution scheme based on the alternating direction method of multipliers (ADMM) because of its fast convergence in practice [9], [22]. A consensus algorithm involving a central coordinator [22] could be implemented, which requires communication to every subsystem and therefore might not be tractable in practice. We propose a scheme, where only nearest-neighbor communication and no global coordination is required. A similar scheme has been presented before for distributed controller synthesis of large-scale systems in [23].

A. Distributed Synthesis for DLMPC

Let us define the local variable vector of subsystem i as

$$s_i = [\mathbf{x}_{\mathcal{N}_i,t,N}^\top,\ \boldsymbol{\alpha}_i^{q-1\top},\ \mathbf{u}_{i,t,N-1}^\top]^\top, \tag{31}$$

and the projection matrices E_{ij}, which project s_i onto those variables over which a consensus needs to be achieved between subsystem i and its neighboring subsystems $j \in \mathcal{N}_i$, i.e., $E_{ij}s_i = E_{ji}s_j$ is the consensus constraint from (30) for subsystem i with j. The decomposed problem in (28) and (30) can now be formulated as the following M subproblems for all $i \in \mathcal{N}$

$$\begin{aligned} &\min_{s_i}\ J^q_{i,t\to t+N}(s_i) + g_i(s_i) \\ &\text{s.t. } E_{ij}s_i = E_{ji}s_j, \quad \forall j \in \mathcal{N}_i. \end{aligned} \tag{32}$$

with $J^q_{i,t\to t+N}(s_i)$ being the cost function in (28), and $g_i(s_i)$ being the indicator function for the constraints in (28), i.e.,

$$g_i(s_i) = \begin{cases} 0 & \text{if } s_i \text{ satisfies the constraints in (28),} \\ +\infty & \text{otherwise.} \end{cases}$$

In order to derive the ADMM steps, we formulate the augmented Lagrangian, which allows a decomposition into the following sum of local terms

$$\mathcal{L}_\rho = \sum_{i\in\mathcal{N}} \mathcal{L}_{\rho,i}, \tag{33}$$

with

$$\begin{aligned} \mathcal{L}_{\rho,i} = {} & J^q_{i,t\to t+N}(s_i) + g_i(s_i) \\ & + \sum_{j\in\mathcal{N}_i} \left(\lambda_{ij}^\top (E_{ij}s_i - E_{ji}s_j) + \frac{\rho}{2}\|E_{ij}s_i - E_{ji}s_j\|_2^2\right). \end{aligned} \tag{34}$$

The modified ADMM update steps are summarized in Algorithm 1. The derivation can be found in [23]. The update steps require communication only between neighboring subsystems, i.e., subsystems that are coupled through their dynamics, constraints, or costs.

The DLMPC for iterative tasks, with distributed solution of the subproblems by Algorithm 1, is given in Algorithm 2.

Algorithm 1: Distributed computation of local input $u^q_{i,t} = u^{*q}_{i,t|t}$ for subsystem i at time t of iteration q

Input: Iteration q, time t, $\rho > 0$, set of neighboring subsystems $\mathcal{N}_i$, current subsystems states $x^q_{i,t}$, initial values $s_i^{(0)}$, $\forall i \in \mathcal{N}$

for $i \in \mathcal{N}$ **do**
  Initialization: Set $\kappa \leftarrow 0$, $\lambda_i^{(0)} \leftarrow 0$;
  while *not converged* **do**
    Communicate $E_{ij}\, s_i^{(\kappa)}$ to neighboring nodes $j \in \mathcal{N}_i$;
    $\lambda_i^{(\kappa+1)} \leftarrow \lambda_i^{(\kappa)} + \rho \sum_{j\in\mathcal{N}_i}(T_{ij}\, s_i^{(\kappa)} - T_{ji}\, s_j^{(\kappa)})$;
    $$s_i^{(\kappa+1)} \leftarrow \underset{s_i}{\operatorname{argmin}} \Big\{ J^q_{i,t\to t+N}(s_i) + g_i(s_i) + s_i^\top \lambda_i^{(\kappa+1)} + \rho \sum_{j\in\mathcal{N}_i} \|T_{ij}\, s_i - \frac{T_{ij}\, s_i^{(\kappa)} + T_{ji}\, s_j^{(\kappa)}}{2}\|^2 \Big\};$$
  end
  $\kappa \leftarrow \kappa + 1$;
  Set $s_i^* = s_i^\kappa$;
  Return $u^{*q}_{i,t|t}$ from s_i^*;
end

Output: local inputs $u^q_{i,t}, \quad \forall i \in \mathcal{N}$

Algorithm 2: DLMPC

Input: Initial states $x_{i,S}$, target states $x_{i,F}$, sets of neighboring subsystems $\mathcal{N}_i$, initial successful feasible trajectories $\mathbf{x}^0_{i,t}$, $\mathbf{u}^0_{i,t}$ with $\mathcal{CS}^0_i$ and $J^0_{i,t\to\infty}(x_{i,t}), \forall x_{i,t} \in \mathbf{x}^0_{i,t}$, $\forall i \in \mathcal{N}$, $q_{\max}$

for *iteration* $q = 1$ *to* $q_{\max}$ **do**
  for $i \in \mathcal{N}$ **do**
    while $x_{i,t} \neq x_{i,F}$ **do**
      Solve local problem (32) via Algorithm 1;
      Apply local input $u^q_{i,t} = u^{*q}_{i,t|t}$;
      Obtain local state $x^q_{i,t}$;
    end
    Update $\mathcal{CS}^q_i$ by adding $\mathbf{x}^*_{i,t}$;
    Compute and save $J^q_{i,t\to\infty}(x^q_{i,t}), \forall x^q_{i,t} \in \mathbf{x}^*_{i,t}$;
  end
  $q = q + 1$;
end

Output: Closed-loop trajectories $\mathbf{x}^*_{i,t}$, $\mathbf{u}^*_{i,t}$, $\forall i \in \mathcal{N}$

B. Properties of the DLMPC

Next, we present our main result on the properties of Algorithm 2. We make the following assumptions.

Assumption 1: We have access to feasible trajectories $\mathbf{x}^q_i$ at iteration $q = 0$ converging to $x_{i,F}$ for all subsystems $i \in \mathcal{N}$, and therefore the convex safe sets at iteration $q = 1$, $\mathcal{CS}^{q-1}_i = \mathcal{CS}^0_i$, are non-empty.

Assumption 2: We assume that the local cost functions $J^q_{i,t\to t+N}(\cdot) + g_i(\cdot)$ in (32) are closed, proper and convex for all subsystems $i \in \mathcal{N}$, and that the unaugmented Lagrangian

$$\mathcal{L}_i = J^q_{i,t\to t+N}(s_i) + g_i(s_i) + \sum_{j\in\mathcal{N}_i} \lambda_{ij}^\top (E_{ij}s_i - E_{ji}s_j)$$

has a saddle point, and that the ADMM update steps in Algortihm 1 are feasible.

In addition to the classical MPC properties, namely, persistent feasibility in each iteration, and asymptotic stability of the equilibria $x_{i,F}$, the following properties hold for the DLMPC in Algorithm 2.

Theorem 1: Consider system (1), with distributed structure (5) and (7). Let Assumptions 1 and 2 hold. Then, the DLMPC in Algorithm 2 has the following properties:

1) The DLMPC is feasible for all $t \geq 0$ and at every iteration $q \geq 1$. The equilibrium points $x_{i,F}$ are asymptotically stable for the closed-loop coupled subsystems under the DLMPC law.
2) The iteration cost $J^q_{0\to\infty}(x_S)$ of the closed-loop system does not increase with the iteration index q, i.e., $J^{q+1}_{0\to\infty} \leq J^q_{0\to\infty}$.
3) If the closed-loop system under the DLMPC converges to the steady-state inputs $\mathbf{u}_i^\infty = \lim_{q\to\infty} \mathbf{u}_i^q$ and the related steady-state trajectories $\mathbf{x}_i^\infty = \lim_{q\to\infty} \mathbf{x}_i^q$, for all subsystems $i \in \mathcal{N}$, and the conditions from [17, Theorem 3] are satisfied, then, $\mathbf{u}_i^\infty$ and $\mathbf{x}_i^\infty$ are global optimal solutions for the IHOCP (9).

Proof: The properties of Theorem 1 have been proven in [17] and [19] for a single system. It therefore suffices to show that the proposed decomposed problem solved in Algorithm 2 is an exact reformulation of the global centralized problem and that the distributed solution method in Algorithm 1 converges to the global optimal solution.

It can easily be seen that the local subproblems in (28) together with the consensus constraints in (30), and their reformulation into the subproblems in (32) are exact reformulations of the global problem in (21). This follows from the decomposability of the cost function in (10) and the structure of the system in (5) and (7), together with the definitions of $\mathcal{CS}_i^q$ and $V_i^{q,*}$ in (26) and (28) with the consensus constraints in (30). For linear system dynamics and convex constraints, the problems (both the global and the local ones) are convex and therefore admit a global optimal solution.

The proposed distributed solution method in Algorithm 1 is equivalent to the update steps of consensus ADMM in [22]. This equivalence has been shown in the derivation of the steps of Algorithm A.1 in [24]. Under Assumption 2, the residuals $E_{ij}s_i - E_{ji}s_j,\ \forall j \in \mathcal{N}_i,\ \forall i \in \mathcal{N}$ in Algorithm 1 asymptotically converge to zero and the cost $\sum_{i\in\mathcal{N}}(J^q_{i,t\to t+N}(\cdot) + g_i(\cdot))$ from (32) asymptotically converges to the global optimal solution. This is true in each time step and therefore $J^q_{0\to\infty}(x_S) = \sum_{i\in\mathcal{N}} J^q_{i,0\to\infty}(x_{i,S})$ converges to the global optimal solution. With the previous results, this is equivalent to the global optimal solution of the global centralized problem (21).

Therefore, the proofs in [17] and [19] for a single system can be applied to the global centralized problem and thus the properties in Theorem 1 hold. ■

Note that the properties in Theorem 1 hold for the global system in (1), i.e., for the ensemble of all coupled subsystems. In particular, property 2) guarantees a decrease in the iteration cost $J^q_{0\to\infty}(x_S) = \sum_{i\in\mathcal{N}} J^q_{i,0\to\infty}(x_{i,S})$ of the sum of costs of all subsystems over iterations, rather than a decrease in the iteration costs $J^q_{i,0\to\infty}(x_{i,S})$ of the individual subsystems. Similarly, the optimal cost function $J^{\mathrm{LMPC}}_{t\to t+N}(\cdot) = \sum_{i\in\mathcal{N}} J^{\mathrm{LMPC}}_{i,t\to t+N}(\cdot)$, is a Lyapunov function for the equilibrium point x_F of the closed loop system (1) rather than the individual cost functions $J^{\mathrm{LMPC}}_{i,t\to t+N}(\cdot)$ for the individual subsystems. Furthermore, Algorithm 1 enables a distributed implementation of the global terminal constraint set $\mathcal{CS}^{q-1}$ on which no distributed structure is imposed. The approach presented in this paper therefore captures the couplings between the subsystems and thus reduces conservatism w.r.t. other approaches of distributed MPC in the literature which impose structure on the terminal cost or terminal constraint sets.

The size of the decomposed local FHOCPs in (28) are of the size of the individual subsystems and are independent of the number of subsystems. Since only nearest-neighbor communication is required in Algorithm 1, also the solution method scales well with the number of subsystems. The number of data points for the construction of the convex safe sets in (26) grows in each iteration with adding the most recent closed loop trajectories to the safe sets in (25). In order to reduce the required computational effort, the set of data points can be truncated, i.e., not all previously seen data points need to be included in the safe sets in (25). For example only the most recent trajectories, or only the previous trajectory can be chosen to be included.

C. Safe and Efficient Data Generation and Domain Enlargement of the DLMPC Policy

In order to use Algorithm 2 for a (possibly iterative) task, data from at least one set of successful feasible trajectories of the subsystems are required to construct the local terminal sets and cost functions, which guarantee the properties in Theorem 1. While successful feasible trajectories might be easy to obtain for distributed systems in some applications, such as by locally or manually controlling multiple loosely coupled subsystems in a non-optimal way, in other applications, such as for tightly coupled subsystems with safety-critical constraints, these data might be difficult to generate. We therefore propose in the following a distributed algorithm which allows the safe and efficient generation of the data required for the computation of the terminal sets and costs in Algorithm 2. We present this data generation method for a control task from given initial states $x_{i,0}^{\mathrm{des}}$ to the target states $x_{i,F}$ of the subsystems. Let us define the following FHOCP, which is similar to the one in (28) except for a different cost function, and with the initial states $x_{i,0}$ being optimization variables

$$\begin{aligned}
&\min_{\substack{\mathbf{x}_{\mathcal{N}_i,t,N},\\ \mathbf{u}_{i,t,N-1},\\ \boldsymbol{\alpha}_i^{q-1}}} \|x_{i,0}^q - x_{i,0}^{\mathrm{des}}\|_2^2 \\
\text{s.t.}\quad & x_{i,k+1|t} = A_{\mathcal{N}_i} x_{\mathcal{N}_i,k|t} + B_i u_{i,k|t},\ k = 0,...,N-1 \\
& x_{\mathcal{N}_i,k|t} \in \mathcal{X}_{\mathcal{N}_i},\ k = t,...,t+N-1 \\
& u^i_{k|t} \in \mathcal{U}_i,\ k = t,...,t+N-1, \\
& x_{i,t+N|t} \in \mathcal{CS}_i^{q-1},
\end{aligned} \tag{35}$$

Algorithm 3: Efficient and safe distributed data generation for the DLMPC policy

Input: Terminal states $x_{i,F}$, sets of neighboring systems $\mathcal{N}_i$, desired initial states $x_{i,0}^{\text{des}}$, $\forall i \in \mathcal{N}$, $r_{\max}$

1 **Initialize:** Set $\mathcal{CS}_i^0 = x_{i,F}$;
2 Set iteration count $r = 1$;
3 **for** $i \in \mathcal{N}$ **do**
4 **while** $\|x_{i,0}^r - x_{i,0}^{\text{des}}\|_2^2 \leq \epsilon$ *and* $r \leq r_{\max}$ **do**
5 Solve (35) to obtain $x_{i,0}^r$;
6 Compute $\mathbf{x}_{i,t}^*$, $\mathbf{u}_{i,t}^*$ via Algorithm 2 with inputs: $x_{i,S} = x_{i,0}^r$, $\mathcal{CS}_i^0 = \mathcal{CS}_i^{r-1}$, $J_{i,t\to\infty}^0 = J_{i,t\to\infty}^r$, $q_{\max} = 1$, until line 7 of Algorithm 2, then break;
7 Update $\mathcal{CS}_i^r$ by $\mathbf{x}_{i,t}^*$;
8 Compute and save $J_{i,t\to\infty}^r(x_{i,t}), \forall x_{i,t} \in \mathbf{x}_{i,t}^*$;
9 $r = r + 1$;
10 **end**
11 Set $\mathcal{CS}_i = \mathcal{CS}_i^r$;
12 **end**

Output: $\mathcal{CS}_i$ containing successful feasible trajectories from $x_{i,0}^{\text{des}}$ to $x_{i,F}$, $\forall i \in \mathcal{N}$

with $\mathbf{x}_{\mathcal{N}_i,t,N}$, $\mathbf{u}_{i,t,N-1}$, $\boldsymbol{\alpha}_i^{q-1}$ as in (29), and where the consensus constraints in (30) have to hold. Note that no initial successful feasible trajectories $\mathbf{x}_{i,t}^0$ need to be available. Instead, we use only the target states $x_{i,F}$ as initial feasible trajectories and therefore define $\mathcal{CS}_i^0 = x_{i,F}$.

Iteratively solving (35) and computing the DLMPC closed-loop trajectories by Algorithm 2 enlarges the domain of the DLMPC policy and converges to feasible trajectories starting at $x_{i,0} = x_{i,0}^{\text{des}}$ and ending in $x_{i,0} = x_{i,F}$, which can used as the input to Algorithm 2. These steps are summarized in the following Algorithm 3

Remark 4: Algorithm 3 can also be used to compute a larger domain of the DLMPC policy. If no specific initial states $x_{i,0}^{\text{des}}$ are given, instead of the cost function $\|x_{i,0}^q - x_{i,0}^{\text{des}}\|_2^2$ in (35), a different cost function can be used, for example to compute the initial states $x_{i,0}$ for all subsystems $i \in \mathcal{N}$ as the points furthest in the direction of interest at the borders of the convex safe sets $\mathcal{CS}_i^q$.

VI. Numerical Experiments

In this section, we present numerical examples to demonstrate the methods of the DLMPC scheme in Algorithm 2 and the data generation in Algorithm 3.

We consider a system of three dynamically coupled subsystems with coupled state constraints. The subsystems have two states each, i.e., $x_i = [x_{i1},\ x_{i2}]^\top$, $\forall i \in \mathcal{N}$. The overall system state is given by $x = [x_1^\top,\ x_2^\top,\ x_3^\top]^\top \in \mathbb{R}^6$, and the input vector by $u = [u_1,\ u_2,\ u_3]^\top \in \mathbb{R}^3$. The system matrices of the global system are given by

$$A = \begin{bmatrix} A_{11} & A_{12} & 0 \\ 0 & A_{22} & A_{23} \\ A_{31} & 0 & A_{33} \end{bmatrix}, \quad B = \text{diag}(B_{11}, B_{22}, B_{33}), \tag{36}$$

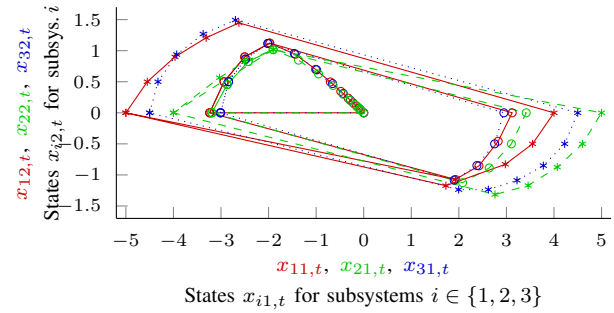

Fig. 1: Domain enlargement for desired initial states $x_{i,0}^{\text{des},1}$, $x_{i,0}^{\text{des},2}$: $\mathcal{CS}_i^q$ of subsystems $i = 1$ (solid red ——), $i = 2$ (dashed green - - -), $i = 3$ (dotted blue .....) over iterations $q = 1, 2$ ($\circ$), $q = 3, 4$ ($*$), starting with $\mathcal{CS}_i^0 = [0,\ 0]^\top$.

with

$$A_{11} = \begin{bmatrix} 1 & 0.5 \\ 0 & 1.1 \end{bmatrix}, \quad A_{22} = \begin{bmatrix} 1.05 & 0.6 \\ 0 & 1 \end{bmatrix}, \quad A_{33} = \begin{bmatrix} 1 & 0.55 \\ 0 & 1.05 \end{bmatrix},$$
$$A_{12} = A_{23} = A_{31} = -\begin{bmatrix} 0.1 & 0.2 \\ 0 & 0.3 \end{bmatrix}, \quad B_{11} = B_{22} = B_{33} = [0, 1]^\top, \tag{37}$$

$$\begin{aligned} -0.9 &\leq x_{11} - x_{21} \leq 0.9, \\ -0.9 &\leq x_{21} - x_{31} \leq 0.9, \\ -3 &\leq u_i \leq 3 \\ -5 &\leq x_{ik} \leq 5, \quad \forall i \in \mathcal{N}, \quad k \in \{1, 2\}. \end{aligned} \tag{38}$$

A. Data Generation

First, we generate the feasible trajectories required as inputs to Algorithm 2, by making use of Algorithm 3. We choose the following desired initial states

$$\begin{aligned} x_{1,0}^{\text{des},1} &= [-5, 0]^\top, & x_{1,0}^{\text{des},2} &= [4, 0]^\top, \\ x_{2,0}^{\text{des},1} &= [-4.5, 0]^\top, & x_{2,0}^{\text{des},2} &= [4.5, 0]^\top, \\ x_{3,0}^{\text{des},1} &= [-4, 0]^\top, & x_{3,0}^{\text{des},2} &= [5, 0]^\top. \end{aligned} \tag{39}$$

We iteratively compute the inital states $x_{i,0}$ as those states closest to $x_{i,0}^{\text{des},1}$ and $x_{i,0}^{\text{des},2}$, in an alternating way, thus enlarging the domain of the DLMPC policy in both the negative and positive x_{i1} directions within the feasible region of the state space.

Remark 5: Note that if no specific initial states $x_{i,0}^{\text{des},1}$ and $x_{i,0}^{\text{des},2}$ are defined, a similar result of domain enlargement is achieved by changing the cost function in (35) to $x_{i,0}^q$ and $-x_{i,0}^q$, respectively.

Figure 1 shows the enlargement of the convex hulls of the safe sets, $\mathcal{CS}_i^q$, of the three subsystems over iterations $q = 0$ to 4 of Algorithm 3. At iteration 4, the given initial states $x_{i,0}^{\text{des},1}$ and $x_{i,0}^{\text{des},2}$ have been reached, i.e., closed-loop trajectories from $x_{i,0}^{\text{des},1}$ to $x_{i,F}$ and from $x_{i,0}^{\text{des},2}$ to $x_{i,F}$ under the DLMPC law have been generated.

B. Iterative Control Task

We consider now the control task to steer the coupled subsystems in (36) from $x_{i,0}^{\text{des},1}$ as in (39) to $x_{i,F} = [0,\ 0]^\top$. The cost function is given as in (12) with $Q_i = I_{n_i}$ and

Iteration	0	1	2	3	4
Sys.	295.63	216.96	216.41	216.31	216.28
Subsys. 1	112.47	87.97	88.07	88.22	88.31
Subsys. 2	113.05	76.52	76.10	75.90	75.78
Subsys. 3	70.11	52.47	52.23	52.19	52.19

5	6	7	8	9	10
216.26	216.26	216.25	216.25	216.25	216.25*
88.36	88.38	88.40	88.41	88.42	88.42
75.72	75.68	75.66	75.64	75.63	75.63
52.19	52.19	52.19	52.20	52.20	52.20

TABLE I: Iteration costs *converged to global optimal solution (computed for the centralized system with $N = 200$).

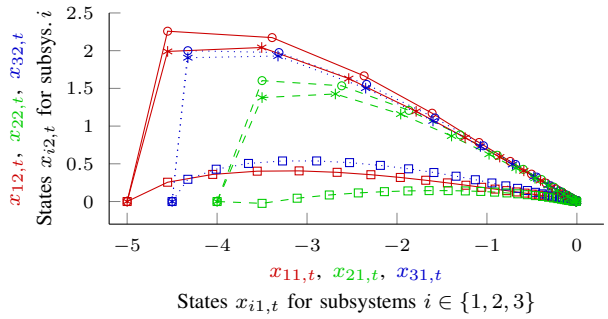

Fig. 2: Iterative regulation task for 3 dynamically coupled subsystems. Trajectores of subystems $i = 1$ (solid red ——), $i = 2$ (dashed green - - -), $i = 3$ (dotted blue .....) shown for iterations $q = 0$ ($\square$), $q = 1$ ($\circ$), $q = 10$ ($*$).

$R_i = I_{m_i}$. We generate the first feasible trajectory by a distributed MPC with time horizon $N = 15$, without terminal sets and constraints, and with $Q_i = 0.1 I_{n_i}$ and $R_i = I_{m_i}$. In real, possibly safety-critical, applications a feasible trajectory could be obtained by manual control of the subsystems, or with the data generation in Algorithm 3, as illustrated in Section VI-A. Table I shows the performance improvement over iterations of Algorithm 2 with a time horizon of $N = 4$. Figure 2 shows the resulting closed-loop trajectories. It is interesting to note that while the iteration costs of subsystems 2 and 3 are decreasing over the iterations, the one of subsystem 1 is increasing. As noted before, Theorem 1 guarantees that the sum, i.e., the iteration cost of the overall system is guaranteed to be non-increasing.

VII. Conclusion

A distributed learning model predictive control scheme was presented, which exploits data in order to construct the terminal cost and constraints of the DMPC problem without imposing the distributed structure of the system. The required computation is done online in a distributed way. It was shown how the scheme can be used to safely explore the state-space and generate the required data or exploit data from iterative control tasks. In addition to recursive feasibility and asymptotic stability, performance improvement over iterations and convergence to the global centralized optimal solution under mild conditions are guaranteed.

References

[1] Y. R. Stürz, A. Eichler, and R. S. Smith, "Scalable controller synthesis for heterogeneous interconnected systems applicable to an overlapping control framework," *Eur. Control Conf.*, vol. 1, pp. 2561–2568, 2018.

[2] S. Riverso and G. Ferrari-Trecate, "Tube-based distributed control of linear constrained systems," pp. 1–10, 2011.

[3] E. Camponogara, D. Jia, B. H. Krogh, and S. Talukdar, "Distributed model predictive control," *Control Syst. Mag.*, vol. 22, no. 1, pp. 44–52, 2002.

[4] P. Giselsson and A. Rantzer, "On feasibility, stability and performance in distributed model predictive control," *IEEE Trans. Automat. Contr.*, vol. 59, no. 4, pp. 1031–1036, 2014.

[5] C. Conte, C. N. Jones, M. Morari, and M. N. Zeilinger, "Distributed synthesis and stability of cooperative distributed model predictive control for linear systems," *Automatica*, vol. 69, pp. 117–125, 2016.

[6] G. Darivianakis, A. Eichler, and J. Lygeros, "Distributed model predictive control for linear systems with adaptive terminal sets," *IEEE Trans. Automat. Contr.*, vol. 65, no. 3, pp. 1044–1056, 2020.

[7] A. Aboudonia, A. Eichler, and J. Lygeros, "Distributed model predictive control with asymmetric adaptive terminal sets for the regulation of large-scale systems," *arXiv:2005.04077*, 2020.

[8] D. P. Bertsekas and J. N. Tsitsiklis, *Parallel and distributed computation: Numerical methods*. Prentice Hall, Inc., 1989.

[9] F. Farokhi, I. Shames, and K. H. Johansson, *Distributed MPC via dual decomposition and alternative direction method of multipliers*. Dordrecht: Springer Netherlands, 2014, pp. 115–131.

[10] P. A. Trodden and J. M. Maestre, "Distributed predictive control with minimization of mutual disturbances," *Automatica*, vol. 77, pp. 31–43, 2017.

[11] S. Lucia, M. Kögel, and R. Findeisen, "Contract-based predictive control of distributed systems with plug and play capabilities," *IFAC-PapersOnLine*, vol. 48, no. 23, pp. 205–211, 2015.

[12] C. Alonso and N. Matni, "Distributed and localized model predictive control via system level synthesis," *ArXiv:1909.10074*, 2019.

[13] E. L. Zhu, Y. R. Stürz, U. Rosolia, and F. Borrelli, "Trajectory Optimization for Nonlinear Multi-Agent Systems using Decentralized Learning Model Predictive Control," *ArXiv:2004.01298*, 2020.

[14] S. Muntwiler, K. P. Wabersich, A. Carron, and M. N. Zeilinger, "Distributed Model Predictive Safety Certification for Learning-based Control," *ArXiv:1911.01832*, 2019.

[15] H. Hu, K. Gatsis, M. Morari, and G. J. Pappas, "Non-Cooperative Distributed MPC with Iterative Learning," *IFAC-PapersOnLine*, 2020.

[16] A. Hock and A. P. Schoellig, "Distributed iterative learning control for multi-agent systems: Theoretic developments and application to formation flying," *Auton. Rob.*, vol. 43, no. 8, pp. 1989–2010, 2019.

[17] U. Rosolia and F. Borrelli, "Learning model predictive control for iterative tasks: A computationally efficient approach for linear system," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 3142–3147, 2017.

[18] U. Rosolia, X. Zhang, and F. Borrelli, "Robust learning model predictive control for linear systems," 2019.

[19] U. Rosolia and F. Borrelli, "Learning model predictive control for iterative tasks. A data-driven control framework." *IEEE Trans. Automat. Contr.*, vol. 63, no. 7, pp. 1883–1896, 2018.

[20] F. Borrelli, *Constrained optimal control of linear and hybrid systems*. Berlin: Springer, 2003.

[21] C. N. Jones and M. Morari, "Polytopic approximation of explicit model predictive controllers," *IEEE Trans. Automat. Contr.*, vol. 55, no. 11, pp. 2542–2553, 2010.

[22] S. Boyd, N. Parikh, E. Chu, P. Borja, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Found. Trends Mach. Learn.*, vol. 3, no. 1, pp. 1–122, 2010.

[23] Y. R. Stürz, A. Eichler, and R. S. Smith, "Distributed control design for heterogeneous interconnected systems," *arXiv preprint arXiv:2004.04876*, 2020.

[24] G. Banjac, F. Rey, P. Goulart, and J. Lygeros, "Decentralized resource allocation via dual consensus ADMM," *Proc. Am. Control Conf.*, vol. 2019-July, pp. 2789–2794, 2019.