Received 21 December 2021; revised 13 March 2022; accepted 8 April 2022. Date of publication 11 May 2022; date of current version 9 June 2022. The review of this paper was arranged by Associate Editor Solmaz S. Kia.

Digital Object Identifier 10.1109/OJCSYS.2022.3171787

Data-Driven Distributed and Localized Model Predictive Control

CARMEN AMO ALONSO [1] (Student Member, IEEE), FENGJUN YANG [2], AND NIKOLAI MATNI [3] (Member, IEEE)

[1]Department of Computing and Mathematical Sciences, California Institute of Technology, Pasadena, CA 91106 USA
[2]Department of Computer and Information Science, University of Pennsylvania, Philadelphia, PA 19104 USA
[3]Department of Electrical and Systems Engineering, University of Pennsylvania, Philadelphia, PA 19104 USA

CORRESPONDING AUTHOR: CARMEN AMO ALONSO (e-mail: camoalon@caltech.edu).

Carmen Amo Alonso and Fengjun Yang contributed equally to this work.

The work of Nikolai Matni was supported in part by NSF award CPS-2038873, in part by CAREER award ECCS-2045834, and in part by a Google Research Scholar award. The work of Carmen Amo Alonso was supported by a Caltech/Amazon AI4Science fellowship. The work of Fengjun Yang was supported in part by NSF CAREER award ECCS-2045834.

ABSTRACT Motivated by large-scale but computationally constrained settings, e.g., the Internet of Things, we present a novel data-driven distributed control algorithm that is synthesized directly from trajectory data. Our method, data-driven Distributed and Localized Model Predictive Control (D^{3}LMPC), builds upon the data-driven System Level Synthesis (SLS) framework, which allows one to parameterize *closed-loop* system responses directly from collected open-loop trajectories. The resulting model-predictive controller can be implemented with distributed computation and only local information sharing. By imposing locality constraints on the system response, we show that the amount of data needed for our synthesis problem is independent of the size of the global system. Moreover, we show that our algorithm enjoys theoretical guarantees for recursive feasibility and asymptotic stability. Finally, we also demonstrate the optimality and scalability of our algorithm in a simulation experiment.

INDEX TERMS Data-driven optimization, decentralized/distributed control, large-scale systems, optimal control.

I. INTRODUCTION

Contemporary large-scale distributed systems such as the Internet of Things enjoy ubiquitous sensing and communication, but are resource constrained in terms of power consumption, memory, and computation power. If such systems are to move from passive data-collecting networks to active distributed control systems, algorithmic approaches that exploit the aforementioned advantages subject to the underlying resource constraints must be developed. Motivated by this emerging paradigm, we seek to devise a distributed control scheme that is (a) model-free, eliminating the need for expensive system identification algorithms, and (b) scalable in implementation and computation. Our hypothesis is that for such systems, collecting local trajectory data from a small subset of neighboring systems is a far more feasible approach than deriving the intricate and detailed system models needed by model-based control algorithms. In this paper, we show that such a data-driven distributed approach can scalably provide optimal performance and constraint satisfaction, along with feasibility and stability guarantees.

Prior work: Data-driven solutions to the linear quadratic regulator (LQR) optimal control problem have been studied extensively in the literature, see for example [1] and references therein. Most closely related to this paper are *direct methods* that do not require a system identification step [2]–[8]. Specifically, we highlight the results of [6], in which behavioral systems theory is applied to parameterize achievable system input/output behavior using past trajectories.[1] This idea has then given rise to several different data-enabled

[1]For a more in-depth treatment of contemporary applications of behavioral system theory to data-driven control, we refer interested readers to [9], [10] and the references therein.

Model Predictive Control (MPC) approaches [11]–[14]. However, these centralized approaches require gathering past trajectories of the global system, which limits their scalability and challenges their applicability to the distributed setting. While recent work has applied data-driven approaches to the distributed setting, making such approaches scalable remains a challenge, as they still require collecting trajectories of the global system. In [15] stabilizing distributed MPC controllers are synthesized from past trajectories for linear systems with unknown dynamics using only local communication. However, the amount of data that is required by the algorithm scales proportionally with the size of the network. In [16], it assumed that "auxiliary" links among subsystems exist, which are then exploited to solve the LQR problem with unknown dynamics and local communication; this assumption however makes its extension to an online approach (e.g., MPC) computionally unappealing. Moreover, while there exist works that guarantee stability in centralized data-driven MPC settings [9], it is unclear how such techniques can be adapted to provide theoretical guarantees in the distributed setting. A complementary line of work relies on data-driven approaches to formulating terminal costs and constraints in order to provide theoretical guarantees (recursive feasibility and asymptotic stability) in distributed MPC approaches where providing guarantees with conventional techniques is in general a hard problem and usually results in conservatism [17], [18]. However, in these cases knowledge of the system dynamics is assumed. It remains as an open question how to develop a *scalable* distributed MPC approach where the system model is unknown and only *local* measurements are available for each subsystem. It is important that such an approach enjoys the same theoretical guarantees of recursive feasibility and asymptotic stability as standard MPC approaches.

Contributions: We address this gap and present a data-driven version of the model-based Distributed Localized MPC (DLMPC) algorithm for linear time-invariant systems in a noise-free setting. The contributions of this paper are two-fold. First, we provide for the first time the necessary theoretical results that allow one to incorporate local communication constraints into the data-driven SLS framework. We rely on recent results on data-driven System Level Synthesis (SLS) [14], which show that optimization problems over system-responses can be posed using only libraries of past system trajectories without explicitly identifying a system model. We extend these results to the localized and distributed setting, where subsystems can only collect and communicate information within a local neighborhood. We note that [14] has been extended to the input-output parameterization (IOP) in [19]. However, IOP does not allow for localized constraints and thus would require longer data trajectories when applied to our problem. Second, we apply these results to the DLMPC problem, where we show that the model-based DLMPC problem can equivalently be posed using only *local* libraries of past system trajectories without explicitly identifying a system model. We then exploit this structure, together with the the separability properties of the objective function and constraints, and provide a distributed synthesis algorithm based on the Alternating Direction Method of the Multipliers (ADMM) [20] where only local information sharing is required. Hence, in the resulting implementation, each sub-controller solves a low-dimensional optimization problem defined over a local neighborhood, requiring only local data sharing and no system model. Moreover, the length of the data trajectory needed scales with the size of local neighborhoods rather than the size of the global system. We note that this result holds not only for decoupled systems, which can be seen as a special case of [21], but also for general linear network systems. Since this problem is analogous to the model-based DLMPC problem [22], [23], our approach directly inherits its guarantees for convergence, recursive feasibility and asymptotic stability [24]. Through numerical experiments, we validate these results and confirm that the complexity of the subproblems solved at each subsystem does not scale with to the full size of the system.

Paper structure: In Section II we present the problem formulation. In Section III, we introduce the necessary preliminaries on the Data-Driven SLS framework. Section IV expands on these results and provides a Data-Driven formulation of SLS that allows for locality constraints to be imposed such that only local information exchange is needed between subsystems. In Section V we apply these results to the Distributed and Localized MPC problem, and provide a distributed algorithm via ADMM that allows the MPC problem to be solved with only local information. In Section VI, we present a numerical experiment and we end in Section VII with conclusions and future directions.

Notation: Lower-case and upper-case letters such as x and A denote vectors and matrices respectively, although lower-case letters might also be used for scalars or functions (the distinction will be apparent from the context). Bracketed indices denote time-step of the real system, i.e., the system input is $u(t)$ at time t, not to be confused with x_t which denotes the predicted state x at time t. Superscripted variables in curly brackets, e.g. $x^{\{k\}}$, correspond to the value of x at the k^{th} iteration of a given algorithm. Calligraphic letters such as $\mathcal{S}$ denote sets, and lowercase script letters such as $\mathfrak{c}$ denotes subsets of $\mathbb{Z}^+$. Square bracket notation i.e., $[x]_i$ denotes the components of x corresponding to subsystem i, and $[x]_{\mathcal{S}}$ denotes the components of x corresponding to every subsystem $j \in \mathcal{S}$. Boldface lower and upper case letters such as $\mathbf{x}$ and $\mathbf{K}$ denote finite horizon signals and block lower triangular (causal) operators, respectively:

$$\mathbf{x} = \begin{bmatrix} x_0 \\ x_1 \\ \vdots \\ x_T \end{bmatrix}, \ \mathbf{K} = \begin{bmatrix} K_0[0] & & & \\ K_1[1] & K_1[0] & & \\ \vdots & \ddots & \ddots & \\ K_T[T] & \dots & K_T[1] & K_T[0] \end{bmatrix},$$

where each x_i is a vector in $\mathbb{R}^n$, and each $K_i[j]$ is the value of matrix K at the j^{th} time-step when computed at time i.

II. PROBLEM FORMULATION

Consider a discrete-time linear time-invariant (LTI) system with dynamics

$$x(t+1) = Ax(t) + Bu(t) \tag{1}$$

composed of N interconnected subsystems. Each subsystem i has dynamics

$$[x(t+1)]_i = \sum_{j \in \mathcal{N}_i} [A]_{ij}[x(t)]_j + [B]_{ii}[u(t)]_i, \tag{2}$$

where $[x]_i$, $[u]_i$ are partitions of the global state $x \in \mathbb{R}^n$ and control input $u \in \mathbb{R}^p$ respectively. Similarly, $[A]_{ij}$ and $[B]_{ij}$ are compatible blocks of system matrices (A, B), and the set $\mathcal{N}_i$ contains all subsystems j such that $[A]_{ij} \neq 0$.[2]

Our goal is to design a *localized* MPC controller for the system when the system model (A, B) is unknown. In this setup, each subsystem i has access to a collection of past local state and input trajectories, and only local communication is possible among subsystems. We model the interconnection topology of the system as a time-invariant unweighted directed graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$ where each subsystem i is identified with a vertex $v_i \in \mathcal{V}$. An edge $e_{ij} \in \mathcal{E}$ exists whenever $j \in \mathcal{N}_i$. A system is *d-localized* if the sub-controllers are restricted to exchange their measurements and control inputs with neighbors at most d hops away, as measured by the topology $\mathcal{G}(\mathcal{V}, \mathcal{E})$.

Definition 1: For a graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$, the *d-incoming set*, *d-outgoing set* and *d-external set* of subsystem i are respectively defined as

- $\mathbf{out}_i(d) := \{v_j \mid \mathbf{dist}(v_i \to v_j) \leq d \in \mathbb{N}\}$,
- $\mathbf{in}_i(d) := \{v_j \mid \mathbf{dist}(v_j \to v_i) \leq d \in \mathbb{N}\}$,
- $\mathbf{ext}_i(d) := \{v_j \mid \mathbf{dist}(v_j \to v_i) > d \in \mathbb{N}\}$,

where $\mathbf{dist}(v_j \to v_i)$ is the distance between v_j and v_i i.e., number of edges in the shortest path connecting v_j to v_i.

Hence, we can enforce a d-local information exchange constraint on the distributed MPC problem – where the size of the local neighborhood d is a *design parameter* – by imposing that each sub-controller policy of subsystem i is computed using only states and control actions collected from its d-hop incoming neighbors. Mathematically, we restrict the control action $[u_t]_i$ to be a function of the form

$$[u_t]_i = \gamma_t^i\left([x_{0:t}]_{j \in \mathbf{in}_i(d)}, [u_{0:t-1}]_{j \in \mathbf{in}_i(d)}\right), \tag{3}$$

for all $t = 0, \ldots, T$ and $i = 1, \ldots, N$, where γ_t^i is a measurable function of its arguments.

With this in mind, we formulate the data-driven localized MPC problem. At each time-step τ, a constrained optimal control problem with finite horizon T is solved, where the current state is used as the initial condition.

$$\begin{aligned}
\min_{x_t, u_t, \gamma_t} \quad & \sum_{t=0}^{T-1} f_t(x_t, u_t) + f_T(x_T) \\
\text{s.t.} \quad & x_0 = x(\tau),\ x(t+1) = Ax(t) + Bu(t), \\
& x_T \in \mathcal{X}_T,\ x_t \in \mathcal{X}_t,\ u_t \in \mathcal{U}_t,\ t \in [0, T-1], \\
& [u_t]_i = \gamma_t^i\left([x_{0:t}]_{j \in \mathbf{in}_i(d)}, [u_{0:t-1}]_{j \in \mathbf{in}_i(d)}\right)\ \forall i,
\end{aligned} \tag{4}$$

where the matrices A and B are unknown, $f_t(\cdot, \cdot)$ and $f_T(\cdot)$ are closed, proper, and convex cost functions, and $\mathcal{X}_t$ and $\mathcal{U}_t$ are closed and convex sets containing the origin. Further, in order to achieve local synthesis and implementation at each subsystem, we work under suitable structural assumptions between the cost function and state and input constraints.

Assumption 1: In formulation (4) the objective function f_t is such that $f_t(x, u) = \sum f_t^i([x]_i, [u]_i)$, and the constraint sets are such that $x \in \mathcal{X}_t = \mathcal{X}_t^1 \times \ldots \times \mathcal{X}_t^N$, where $x \in \mathcal{X}_t$ if and only if $[x]_i \in \mathcal{X}_t^i$ for all i and $t \in \{0, \ldots, T\}$, and idem for $\mathcal{U}_t$.

In what follows we show problem (4) admits a distributed solution and implementation requiring only local data and no explicit estimate of the system model.

III. DATA-DRIVEN SYSTEM LEVEL SYNTHESIS

In this section, we introduce an abridged summary of some of the preliminary work on SLS [25], [26] and its extension to a data-driven formulation [14] based on the behavioral framework of Willems [27]. In following sections, we build on these concepts to provide the necessary results to provide a tractable reformulation of problem (4).

A. SYSTEM LEVEL SYNTHESIS PARAMETRIZATION

The following is adapted from Section2 of [26]. Consider the dynamics of the LTI system (1) evolving over a finite horizon T and subject to an additive disturbance $w(t)$ at each time-step t. We can compactly express the dynamics as

$$\mathbf{x} = Z(\hat{A}\mathbf{x} + \hat{B}\mathbf{u}) + \mathbf{w}, \tag{5}$$

where $\mathbf{x}$, $\mathbf{u}$ and $\mathbf{w}$ are the finite horizon signals corresponding to state, control input, and disturbance respectively. By convention, we define the disturbance to contain the initial condition, i.e., $\mathbf{w} = [x_0^\top\ w_0^\top\ \ldots\ w_{T-1}^\top]^\top$.[3] Here, Z is the block-downshift matrix,[4] $\hat{A} := \text{blkdiag}(A, \ldots, A)$, and $\hat{B} := \text{blkdiag}(B, \ldots, B, 0)$.

Let the control input of this system be a causal linear time-varying state-feedback controller, i.e., $\mathbf{u} = \mathbf{K}\mathbf{x}$ for controller $\mathbf{K}$ block-lower triangular. Then the closed-loop response of system (5) is given by:

$$\mathbf{x} = (I - Z(\hat{A} + \hat{B}\mathbf{K}))^{-1}\mathbf{w} =: \mathbf{\Phi}_x\mathbf{w} \tag{6a}$$

$$\mathbf{u} = \mathbf{K}(I - Z(\hat{A} + \hat{B}\mathbf{K}))^{-1}\mathbf{w} =: \mathbf{\Phi}_u\mathbf{w}. \tag{6b}$$

The SLS approach relies on the so-called *system responses* $\mathbf{\Phi}_x$ and $\mathbf{\Phi}_u$ to parametrize the set of achievable closed-loop behaviors of the system by virtue of the following theorem:

Theorem 1: (Theorem 2.1 of [26]) For the system (1) evolving under the state-feedback policy $\mathbf{u} = \mathbf{K}\mathbf{x}$ with $\mathbf{K}$ a block-lower-triangular matrix, the following are true

[2] As in [15], we assume block-diagonal B to simplify presentation. Our results can be extended to B matrices satisfying locality as in Section III-B.

[3] We introduce here the general SLS framework and specialize the notation to the noiseless setting described in (1) later in this section.

[4] Identity matrices along its first block sub-diagonal and zeros elsewhere.

1) The affine subspace

$$Z_{AB}\mathbf{\Phi} := \begin{bmatrix} I - Z\hat{A} & -Z\hat{B} \end{bmatrix} \begin{bmatrix} \mathbf{\Phi}_x \\ \mathbf{\Phi}_u \end{bmatrix} = I \tag{7}$$

with block-lower-triangular $\{\mathbf{\Phi}_x, \mathbf{\Phi}_u\}$ parameterizes all possible system responses (6).
2) For any block lower-triangular matrices $\{\mathbf{\Phi}_x, \mathbf{\Phi}_u\}$ satisfying (7), the controller $\mathbf{K} = \mathbf{\Phi}_u\mathbf{\Phi}_x^{-1}$ achieves the desired response (6) from $\mathbf{w} \mapsto (\mathbf{x}, \mathbf{u})$.

Theorem 1 allows one to reformulate an optimal control problem over state and input variables $(\mathbf{x}, \mathbf{u})$ as an equivalent one over system responses $(\mathbf{\Phi}_x, \mathbf{\Phi}_u)$. Additional details are provided in Section II of [26].

B. LOCALITY CONSTRAINTS IN SYSTEM LEVEL SYNTHESIS

The SLS framework allows one to take into account the structure of a LTI system (1) composed of subsystems (2) [25]. In particular, the information constraints defined in (3) can be enforced via *locality constraints* on the system responses (6). To see this, note that according to Theorem 1, the controller $\mathbf{K} = \mathbf{\Phi}_u\mathbf{\Phi}_x^{-1}$ achieves the desired closed-loop behavior characterized by $\mathbf{\Phi}_x$ and $\mathbf{\Phi}_u$. Such a controller can be implemented as:

$$\mathbf{u} = \mathbf{\Phi}_u\hat{\mathbf{w}}, \quad \mathbf{x} = (\mathbf{\Phi}_x - I)\hat{\mathbf{w}}, \quad \hat{\mathbf{w}} = \mathbf{x} - \hat{\mathbf{x}}, \tag{8}$$

where $\hat{\mathbf{x}}$ can be interpreted as a nominal state trajectory, and $\hat{\mathbf{w}} = Z\mathbf{w}$ is a delayed reconstruction of the disturbance. Given this controller implementation, any structure imposed on the maps $(\mathbf{\Phi}_x, \mathbf{\Phi}_u)$ is directly translated to structure on controller implementation (8). Hence, one can transparently impose information exchange constraints as sparsity structure on the responses $(\mathbf{\Phi}_x, \mathbf{\Phi}_u)$.

Definition 2: Let $[\mathbf{\Phi}_x]_{ij}$ be the submatrix of $\mathbf{\Phi}_x$ describing the map from disturbance $[w]_j$ to the state $[x]_i$ of subsystem i. The map $\mathbf{\Phi}_x$ is *d-localized* if for every subsystem j, $[\mathbf{\Phi}_x]_{ij} = 0 \,\forall\, i \notin \mathbf{out}_j(d)$. The definition for *d-localized* $\mathbf{\Phi}_u$ is analogous but with disturbance to control action $[u]_i$.

By simply enforcing the system responses $\mathbf{\Phi}_x$ and $\mathbf{\Phi}_u$ to have a d-localized and $(d+1)$-localized sparsity pattern,[5] only a local subset $[\hat{\mathbf{w}}]_{\mathbf{in}_i(d)}$ of $\hat{\mathbf{w}}$ is necessary for subsystem i to construct for $[\mathbf{x}]_i$ and $[\mathbf{u}]_i$. Therefore, the information exchange needed between subsystems during the controller synthesis is limited to d-hop neighbors, as defined by constraint (3). Moreover, this constraint can be imposed on the system responses as an affine subspace.

Definition 3: The subspace $\mathcal{L}_d := \{(\mathbf{\Phi}_x, \mathbf{\Phi}_u) : [\mathbf{\Phi}_x]_{ij} = 0 \;\forall i \notin \mathbf{out}_j(d),\; [\mathbf{\Phi}_u]_{ij} = 0 \;\forall i \notin \mathbf{out}_j(d+1)\}$ enforces d-locality constraints. A system (A, B) is d-localizable if the intersection of $\mathcal{L}_d$ with the affine space of achievable system responses (7) is non-empty.

As introduced in [22], when local model information i.e. $[A]_{ij}$, $[B]_i$ is available, locality constraints allow the MPC subroutine (4) to be reformulated into a tractable problem that admits a distributed solution. Because the setting considered in this paper has no driving noise, only the first block columns of $\mathbf{\Phi}_x$ and $\mathbf{\Phi}_u$ are necessary to characterize system behavior. For this reason, in what follows we abuse notation and use $\mathbf{\Phi}_x$ and $\mathbf{\Phi}_u$ to denote only the first block column of the original block-lower triangular matrices. Therefore, (6) reduces to $\mathbf{x} = \mathbf{\Phi}_x x_0$ and $\mathbf{u} = \mathbf{\Phi}_u x_0$.

Hence, in a *model-based* setting the MPC subroutine (4) can be reformulated as

$$\begin{aligned} \underset{\mathbf{\Phi}}{\text{minimize}} \quad & f(\mathbf{\Phi}x_0) \\ \text{s.t.} \quad & x_0 = x(t),\; \mathbf{\Phi}x_0 \in \mathcal{P}, Z_{AB}\mathbf{\Phi} = I,\; \mathbf{\Phi} \in \mathcal{L}_d, \end{aligned} \tag{9}$$

where f is a convex function compatible with all the f_t in formulation (4), and $\mathcal{P}$ is defined so that $\mathbf{\Phi}x_0 \in \mathcal{P}$ if and only if $x(t) \in \mathcal{X}_t$ and $u(t) \in \mathcal{U}_t$ for all $t = 0, \ldots, T$. For additional details on this formulation readers are referred to [23] and references therein.

Remark 1: Although $\mathcal{L}_d$ is always a convex subspace, not all systems are d-localizable. For systems that are not d-localizable, constraint $\mathbf{\Phi} \in \mathcal{L}_d$ would lead to an infeasible subroutine (9). The locality diameter d can be viewed as a design parameter (independent of T) that is application dependent and captures how "far" in the interconnection topology a disturbance striking a subsystem is allowed to spread. For the remainder of the paper, we assume that there exists a $d \ll n$ such that the system (A, B) to be controlled is d-localizable. For more context on locality constraints, see [25], [28].

C. DATA-DRIVEN SYSTEM LEVEL SYNTHESIS

This subsection is adapted from Section II of [14].

Behavioral system theory [6], [9], [10], [29] offers a natural way of studying the behavior of a dynamical system in terms of its input/output signals. In particular, Willem's Fundamental Lemma [27] offers a parametrization of state and input trajectories based on past trajectories as long as the data matrix satisfies a notion of persistence of excitation.

Definition 4: A finite-horizon signal $\mathbf{x}$ with horizon T is *persistently exciting* (PE) of order L if the Hankel matrix

$$H_L(\mathbf{x}) := \begin{bmatrix} x(0) & x(1) & \ldots & x(T-L) \\ x(1) & x(2) & \ldots & x(T-L+1) \\ \vdots & \vdots & \ddots & \vdots \\ x(L-1) & x(L) & \ldots & x(T-1) \end{bmatrix}$$

has full rank.

Lemma 1: (Willem's Fundamental Lemma [29]) Consider the LTI system (1) with controllable (A, B) matrices, and assume that there is no driving noise. Let $\{\tilde{\mathbf{x}}, \tilde{\mathbf{u}}\}$ be the state and input signals generated by the system over a horizon T. If $\tilde{\mathbf{u}}$ is PE of order $n + L$, then the signals $\mathbf{x}$ and $\mathbf{u}$ are valid trajectories of length L of the system (1) if and only if

$$\begin{bmatrix} \mathbf{x} \\ \mathbf{u} \end{bmatrix} = H_L(\tilde{\mathbf{x}}, \tilde{\mathbf{u}})\, g \text{ for some } g \in \mathbb{R}^{T-L+1}, \tag{10}$$

where $H_L(\tilde{\mathbf{x}}, \tilde{\mathbf{u}}) := [\, H_L(\tilde{\mathbf{x}})^\mathsf{T} \; H_L(\tilde{\mathbf{u}})^\mathsf{T} \,]^\mathsf{T}$.

[5] We impose $\mathbf{\Phi}_u$ to be $(d+1)$-localized because the "boundary" controllers at distance $d+1$ must take action in order to localize the effects of a disturbance within the region of size d, (see [26] for more details).

A natural connection can be established between the data-driven parametrization (10) and the SLS parametrization (6). In particular, the achievability constraint (7) can be replaced by a data-driven representation by applying Willems' Fundamental Lemma [29] to the columns of the system responses. Given a system response, we denote the set of columns corresponding to subsystem i as $\mathbf{\Phi}^i$, i.e.,

$$\mathbf{\Phi} = \left[\, \mathbf{\Phi}^1 \ \mathbf{\Phi}^2 \ \ldots \ \mathbf{\Phi}^N \,\right].$$

The key insight is that, by definition of the system responses (6), $\mathbf{\Phi}^i_x$ and $\mathbf{\Phi}^i_u$ are the impulse response of $\mathbf{x}$ and $\mathbf{u}$ to $[x_0]_i$, which are themselves, valid system trajectories that can be characterized using Willems' Fundamental Lemma. This can be seen from the following decomposition of the trajectories

$$\begin{bmatrix} \mathbf{x} \\ \mathbf{u} \end{bmatrix} = \mathbf{\Phi} x_0 = \sum_{i=1}^{N} \mathbf{\Phi}^i [x_0]_i.$$

Lemma 2: (Lemma 1 of [14]) Given the assumptions of Lemma 1, the set of feasible solutions to constraint (7) over a time horizon $t = 0, 1, ..., L-1$ can be equivalently characterized as:

$$\{\mathbf{\Phi} : Z_{AB}\mathbf{\Phi} = I\} = \{H_L(\tilde{\mathbf{x}}, \tilde{\mathbf{u}})\mathbf{G} : \mathbf{G} \text{ s.t. } H_1(\tilde{\mathbf{x}})\mathbf{G} = I\}. \quad (11)$$

The following Corollary follows naturally from Lemma 2, and will be useful later to provide locality results in this data-driven parametrization.

Corollary 1: The following is true:

$$\{\mathbf{\Phi}^i : Z_{AB}\mathbf{\Phi}^i = I^i\} = \{H_L(\tilde{\mathbf{x}}, \tilde{\mathbf{u}})\mathbf{G}^i : \mathbf{G}^i \text{ s.t. } H_1(\tilde{\mathbf{x}})\mathbf{G}^i = I^i\},$$

where I^i is the i-th block column of the identity matrix.

Proof: This follows directly from Lemma 2 by noting that the constraints can be separated column-wise. ∎

While this connection between SLS and the behavioral formulation does not offer an immediate benefit, we will build on it in the following sections to equip the data-driven parametrization (11) with locality constraints so as to provide a reformulation of the localized MPC subroutine (4).

IV. LOCALIZED DATA-DRIVEN SYSTEM LEVEL SYNTHESIS

In this section we present the results that allow us to recast the constraints in (9) in a localized data-driven parametrization. We first extend Lemma 2 to provide a naive way to parametrize localized system responses from global trajectories (Lemma 3). We then show a key property of localized system responses in Lemma 4, which is essential in proving Theorem 2: all localized system responses can be characterized using only locally collected trajectories satisfying PE conditions scaling with the size d of the localized region, rather than the global system size.

A. LOCALITY CONSTRAINTS IN DATA-DRIVEN SYSTEM LEVEL SYNTHESIS

We start by rewriting the locality constraints using the data-driven parameterization (11).

Lemma 3: Consider the LTI system (1) with controllable (A, B) matrices, where each subsystem (2) is subject to locality constraints (3). Assume that there is no driving noise. Given the state and input trajectories $\{\tilde{\mathbf{x}}, \tilde{\mathbf{u}}\}$ generated by the system over a horizon T with $\mathbf{u}$ PE of order at least $n + L$, the following parametrization over $\mathbf{G}$ characterizes all possible d-localized system responses over a time span of $L - 1$:

$$\begin{aligned} &H_L(\tilde{\mathbf{x}}, \tilde{\mathbf{u}})\mathbf{G}, \text{ for all } \mathbf{G} \text{ s.t. } H_1(\tilde{\mathbf{x}})\mathbf{G} = I, \\ &\qquad H_L([\tilde{\mathbf{x}}]_i)\mathbf{G}^j = 0 \ \forall j \notin \mathbf{in}_i(d), \\ &\qquad H_L([\tilde{\mathbf{u}}]_i)\mathbf{G}^k = 0 \ \forall k \notin \mathbf{in}_i(d+1), \end{aligned} \quad (12)$$

for all $i = 1, \ldots, N$.

Proof: We aim to show that

$$\{\mathbf{\Phi} : Z_{AB}\mathbf{\Phi} = I, \mathbf{\Phi} \in \mathcal{L}_d\} = \{H_L(\tilde{\mathbf{x}}, \tilde{\mathbf{u}})\mathbf{G} : \mathbf{G} \text{ s.t. (12)}\}.$$

($\subseteq$) First, suppose that $\mathbf{\Phi} \in \mathcal{L}_d$ satisfies that $Z_{AB}\mathbf{\Phi} = I$. From Lemma 2, we immediately have that there exists a matrix $\mathbf{G}$ s.t. $\mathbf{\Phi} = H_L(\tilde{\mathbf{x}}, \tilde{\mathbf{u}})\mathbf{G}$. Thus, we need only verify that this $\mathbf{G}$ satisfies the linear constraint in (12). This follows directly from the assumption that $\mathbf{\Phi} \in \mathcal{L}_d$, which states that

$$\begin{aligned} H_L([\tilde{\mathbf{x}}]_i)\mathbf{G}^j &= [\mathbf{\Phi}_x]_{ij} = 0 \ \forall j \notin \mathbf{in}_i(d), \\ H_L([\tilde{\mathbf{u}}]_i)\mathbf{G}^k &= [\mathbf{\Phi}_u]_{ik} = 0 \ \forall k \notin \mathbf{in}_i(d+1). \end{aligned}$$

Hence, $\mathbf{\Phi} \in$ RHS, proving this direction.

($\supseteq$) Now suppose that there exists a $\mathbf{G}$ that satisfies the constraints on the RHS and let $\mathbf{\Phi} = H_L(\tilde{\mathbf{x}}, \tilde{\mathbf{u}})\mathbf{G}$. Since $H_1(\tilde{\mathbf{x}})\mathbf{G} = I$, from Lemma 2, we have that $\mathbf{\Phi}$ is achievable. From the other two constraints, we have that $\mathbf{\Phi} \in \mathcal{L}_d$, proving this direction and hence the lemma. ∎

It is important to note that even though Lemma 3 allows one to capture the locality constraint (3) by simply translating the locality constraints over $\mathbf{\Phi}$ to constraints over $\mathbf{G}$, it cannot be implemented with only local information exchange. In order to satisfy the constraints (12), each subsystem has to have access to global state and input trajectories and construct a global Hankel matrix. The PE condition of Lemma 1 implies that the length of the trajectory that needs to be collected grows with the dimension of the global system state. In what follows we show how constraint (12) can further be relaxed to only require local information without introducing any additional conservatism.

B. LOCALIZED DATA-DRIVEN SYSTEM LEVEL SYNTHESIS

In this subsection we show that constraint (12) can be enforced (i) with local communication between neighbors i.e. no constraints are imposed outside each subsystem d-neighborhood, and (ii) the amount of data needed i.e., trajectory length, only scales with the size of the d-localized neighborhood, and not the global system. We start with (i) in the following lemma.

Definition 5: Given a subsystem i satisfying the local dynamics (2), we define its *augmented d-localized subsystem* as the system composed by the states $[x]_{\mathbf{in}_i(d+1)}$ and augmented control actions $[\bar{u}]_i := ([u]^{\mathsf{T}}_{\mathbf{in}_i(d+2)} \ [x]^{\mathsf{T}}_j)^{\mathsf{T}}$, $\forall j$ *s.t.* $\mathbf{dist}(j \rightarrow$

$i) = d+2$. That is, the system given by

$$\begin{aligned}[x(t+1)]_{\mathbf{in}_i(d+1)} = [A]_{\mathbf{in}_i(d+1)}[x(t)]_{\mathbf{in}_i(d+1)} \\ + [\bar{B}]_{\mathbf{in}_i(d+1)}[\bar{u}(t)]_i, \end{aligned} \tag{13}$$

with $\bar{B} := [\,[B]_{\mathbf{in}_i(d+2)}\ [A]_{ij}\,]\ \forall\ j$ s.t. $dist(j \to i) = d+2$.

Notice that by treating the state of the boundary subsystems as additional control inputs, we can view the augmented d-localized system as a standalone LTI system.

Lemma 4: For $i = 1, \ldots, N$, let $\mathbf{\Psi}^i$ be an achievable system response for the augmented d-localized subsystem (13) of subsystem i. Further assume that each $\mathbf{\Psi}^i$ satisfies constraints (14):

$$[\mathbf{\Psi}_x^i]_j = 0,\ \forall j \text{ s.t. } d+1 \le \mathbf{dist}(j \to i) \le d+2, \tag{14a}$$

$$[\mathbf{\Psi}_u^i]_j = 0,\ \forall j \text{ s.t. } \mathbf{dist}(j \to i) = d+2 \tag{14b}$$

for all i. Then, the system response $\mathbf{\Phi}$ defined by (15) is achievable for system (1) and d-localized.

$$[\mathbf{\Phi}]_{ij} := \begin{cases} [\mathbf{\Psi}^i]_j, & \forall j \in \mathbf{in}_i(d+1) \\ 0, & \text{otherwise} \end{cases} \tag{15}$$

for all $i = 1, \ldots, N$ is also achievable and d-localized.

Proof: First, from the fact that $\mathbf{\Psi}^i$ is achievable for all $i = 1, \ldots, N$, we have that $\Phi_x[0] = I$ by construction. Thus, to show that $\mathbf{\Phi}$ is achievable, it suffices to show that

$$\Phi_x[t+1] = A\Phi_x[t] + B\Phi_u[t], \quad \forall\, 0 \le t \le T-1.$$

We show this block-column-wise: we show that the block columns Φ_x^i and Φ_u^i associated with subsystem i satisfy

$$\Phi_x^i[t+1] = A\Phi_x^i[t] + B\Phi_u^i[t], \quad \forall\, 0 \le t \le T-1. \tag{16}$$

We note that the argument below for subsystem i applies identically to all subsystems. For the i-th subsystem, we further partition the rows of its corresponding block-columns into four subsets as follows:

$$\Phi_x^i = \left[[\Phi_x^i]^\mathsf{T}_{\mathbf{in}_i(d)}\ [\Phi_x^i]^\mathsf{T}_{\mathbf{b}_i(d+1)}\ [\Phi_x^i]^\mathsf{T}_{\mathbf{b}_i(d+2)}\ [\Phi_x^i]^\mathsf{T}_{\mathbf{ext}_i(d+2)} \right]^\mathsf{T},$$

where $\mathbf{b}_i(k+1) = \mathbf{in}_i(k+1) \setminus \mathbf{in}_i(k)$ correspond to subsystems that are $k+1$ hops away from subsystem i. Identical notation holds for the partition of Φ_u^i.

Using this partition, we have the following for Φ_x^i and $\Phi_u^i[t]$ given their definition in terms of $\mathbf{\Psi}$:

$$\Phi_x^i[t] = \begin{bmatrix} [\Psi_x^i[t]]_{\mathbf{in}_i(d)} \\ 0 \\ 0 \\ 0 \end{bmatrix}, \quad \Phi_u^i[t] = \begin{bmatrix} [\Psi_u^i[t]]_{\mathbf{in}_i(d)} \\ [\Psi_u^i[t]]_{\mathbf{b}_i(d+1)} \\ 0 \\ 0 \end{bmatrix}. \tag{17}$$

We also partition A and B accordingly, where

$$A = \begin{bmatrix} A^{\mathbf{in}(d)}_{\mathbf{in}(d)} & A^{\mathbf{in}(d)}_{\mathbf{b}_i(d+1)} & 0 & 0 \\ A^{\mathbf{b}_i(d+1)}_{\mathbf{in}(d)} & A^{\mathbf{b}_i(d+1)}_{\mathbf{b}_i(d+1)} & A^{\mathbf{b}_i(d+1)}_{\mathbf{b}_i(d+2)} & 0 \\ 0 & A^{\mathbf{b}_i(d+2)}_{\mathbf{b}_i(d+1)} & A^{\mathbf{b}_i(d+2)}_{\mathbf{b}_i(d+2)} & A^{\mathbf{b}_i(d+2)}_{\mathbf{ext}_i(d+2)} \\ 0 & 0 & A^{\mathbf{ext}_i(d+2)}_{\mathbf{b}_i(d+2)} & A^{\mathbf{ext}_i(d+2)}_{\mathbf{ext}_i(d+2)} \end{bmatrix},$$

$$B = \begin{bmatrix} B^{\mathbf{in}(d)}_{\mathbf{in}(d)} & 0 & 0 & 0 \\ 0 & B^{\mathbf{b}_i(d+1)}_{\mathbf{b}_i(d+1)} & 0 & 0 \\ 0 & 0 & B^{\mathbf{b}_i(d+2)}_{\mathbf{b}_i(d+2)} & 0 \\ 0 & 0 & 0 & B^{\mathbf{ext}_i(d+2)}_{\mathbf{ext}_i(d+2)} \end{bmatrix}. \tag{18}$$

Here, the superscript represents an index on the block row and the subscript represents an index on the block column. The sparsity pattern follows from the definition of augmented d-localized subsystems and the subsystem dynamics (2).

We now show that (16) holds for each Φ^i. First,

$$\begin{aligned} &[\Phi_x^i[t+1]]_{\mathbf{in}_i(d)} \\ &\quad = [A\Phi_x^i[t] + B\Phi_u^i[t]]_{\mathbf{in}_i(d)} \\ &\quad = A^{\mathbf{in}_i(d)}_{\mathbf{in}_i(d)}[\Psi_x^i[t]]_{\mathbf{in}_i(d)} + B^{\mathbf{in}_i(d)}_{\mathbf{in}_i(d)}[\Psi_u^i[t]]_{\mathbf{in}_i(d)} \\ &\qquad + B^{\mathbf{in}_i(d)}_{\mathbf{b}_i(d+1)}[\Psi_u^i[t]]_{\mathbf{b}_i(d+1)} \\ &\quad = [\Psi_x^i[t+1]]_{\mathbf{in}_i(d)}, \end{aligned}$$

where the second equality comes from the sparsity patterns of A, B ((18)), and $\mathbf{\Phi}^i$ ((17)), and the third equality from the achievability of $\mathbf{\Psi}^i$. Similarly, to show that the boundary subsystems satisfy the dynamics, we note that

$$\begin{aligned} &[\Phi_x^i[t+1]]_{\mathbf{b}_i(d+1)} \\ &\quad = A^{\mathbf{b}_i(d+1)}_{\mathbf{in}_i(d)}[\Psi_x^{(i)}(t)]_{\mathbf{in}_i(d)} + B^{\mathbf{b}_i(d+1)}_{\mathbf{b}_i(d+1)}[\Psi_u^{(i)}(t)]_{\mathbf{b}_i(d+1)} \\ &\quad = [\Psi_x^{(i)}(t+1)]_{\mathbf{b}_i(d+1)} \\ &\quad = 0. \end{aligned}$$

Lastly, from the sparsity pattern (17), (18)), we have that

$$[\Phi_x^{(i)}(t+1)]_{\mathbf{ext}_i(d)} = [A\Phi_x^i[t] + B\Phi_u^i[t]]_{\mathbf{ext}_i(d)} = 0,$$

concluding the proof for the achievability of $\mathbf{\Phi}$. We end by noting that $\mathbf{\Phi}$ is d-localized by construction. ■

In light of this result, locality constraints as in Definition 2 i.e., $[\mathbf{\Phi}_x]_{ij} = 0\ \forall\ i \notin \mathbf{out}_j(d)$, do not need to be imposed on every subsystem $i \notin \mathbf{out}_j(d)$. Instead, it suffices to impose this constraint only on subsystems i at a distance $d+2$ of subsystem j. Intuitively, this can be seen as a constraint on the propagation of a signal: if $[w]_j$ has no effect on subsystem i at distance $d+1$ because $[\mathbf{\Phi}_x]_{ij} = 0$, then the propagation of that signal is stopped and localized within that neighborhood. This idea allows to reformulate constraint (12) so that it can be imposed with only local communications.

However, despite the fact that locality constraints can now be achieved with local information exchanges, the amount of data that needs to be collected scales with the global size of the network n because we require that the control trajectory be at least PE of order at least $n + L$. In the following theorem, we build upon the previous results and show how this requirement can also be reduced to only depend on the size of a d-localized neighborhood.

Theorem 2: Consider the LTI system (1) composed of subsystems (2), each with controllable $([A]_{\mathbf{in}_i(d+2)}, [B]_{\mathbf{in}_i(d+2)})$

matrices for the augmented d-localized subsystem i. Assume that there is no driving noise and that the local control trajectory at the d-localized subsystem $[\tilde{\mathbf{u}}]_{\mathbf{in}_i(d+1)}$ is PE of order at least $n_{\mathbf{in}_i(d)} + L$, where $n_{\mathbf{in}_i(d)}$ is the dimension of $[\tilde{\mathbf{x}}]_{\mathbf{in}_i(d)}$. Then, $\mathbf{\Phi}$ is an achievable d-localized system response for each subsystem (2) if and only if it can be written as

$$[\mathbf{\Phi}^i]_{\mathbf{in}_i(d)} = H_L([\tilde{\mathbf{x}}]_{\mathbf{in}_i(d+1)}, [\tilde{\mathbf{u}}]_{\mathbf{in}_i(d+1)})\mathbf{G}^i, \tag{19a}$$

$$[\mathbf{\Phi}^i]_{\mathbf{ext}_i(d+1)} = 0, \tag{19b}$$

where $\mathbf{G}^i$ satisfies

$$H_1([\tilde{\mathbf{x}}]_{\mathbf{in}_i(d+1)})\mathbf{G}^i = I^i, \tag{20a}$$

$$H_L([\tilde{\mathbf{x}}]_j)\mathbf{G}^i = 0 \ \forall i, j \text{ s.t. } d+1 \leq \mathbf{dist}(j \to i) \leq d+2, \tag{20b}$$

$$H_L([\tilde{\mathbf{u}}]_j)\mathbf{G}^i = 0 \ \forall i, j \text{ s.t. } \mathbf{dist}(j \to i) = d+2. \tag{20c}$$

Proof: ($\Rightarrow$) We first show that any d-localized system response $\mathbf{\Phi}$ can be parameterized by a set of matrices $\{\mathbf{G}^i\}_{i=1}^N$. First, since $\mathbf{\Phi}$ is d-localized, each d-localized subsystem impulse response $[\mathbf{\Phi}^i]_{\mathbf{in}_i(d+1)}$ is achievable on the augmented d-localized subsystem i. Thus, it follows from Corollary 1 that there exists $\mathbf{G}^i$ satisfying constraint (20a) such that

$$[\mathbf{\Phi}^i]_{\mathbf{in}_i(d)} = H_L([\tilde{\mathbf{x}}]_{\mathbf{in}_i(d)}, [\tilde{\mathbf{u}}]_{\mathbf{in}_i(d+1)})\mathbf{G}^i.$$

Since $\mathbf{\Phi}$ is d-localized, we have that $\mathbf{G}^i$ satisfies both constraints (20b) and (20c), concluding this direction.
($\Leftarrow$) Now we show that if each $\mathbf{G}^i$ satisfies the constraint (20) for all $i = 1, \ldots, N$, then the resulting $\mathbf{\Phi}$ is achievable and localized. Consider the augmented d-localized subsystem i and define

$$\mathbf{\Psi}^i = H_L([\tilde{\mathbf{x}}]_{\mathbf{in}_i(d)}, [\tilde{\mathbf{u}}]_{\mathbf{in}_i(d+1)})\mathbf{G}^i.$$

From Corollary 1, we have that $\mathbf{\Psi}^i$ is an achievable impulse response on the augmented d-localized subsystem i. Moreover, by construction it satisfies the sparsity condition in Lemma 4. Thus, constructing $\mathbf{\Phi}$ using $\mathbf{\Psi}$ as in (15) we conclude that $\mathbf{\Phi}$ is achievable and d-localized. ■

Corollary 2: Consider a function $g : \mathbf{\Phi} \to \mathbb{R}$ such that

$$g(\mathbf{\Phi}) = \sum_{i=1}^{N} g^i([\mathbf{\Phi}^i]_{\mathbf{in}_i(d+1)}).$$

Then, solving the optimization problem

$$\min \ g(\mathbf{\Phi}) \ s.t. \ Z_{AB}\mathbf{\Phi} = I, \ \ \mathbf{\Phi} \in \mathcal{L}_d$$

is equivalent to solving

$$\min \sum_{i=1}^{N} g^i \left(H_L\left([\tilde{\mathbf{x}}]_{\mathbf{in}_i(d)}, [\tilde{\mathbf{u}}]_{\mathbf{in}_i(d+1)}\right)\mathbf{G}^i\right)$$

$$s.t. \ \mathbf{G}^i \text{ satisfies (20) for all } i = 1 \ldots, N,$$

and then constructing $\mathbf{\Phi}$ as per (19).

This result provides a data-driven approach in which locality constraints, as in (3), can be seamlessly considered and imposed by means of an affine subspace where only local information exchanges are necessary. Moreover, the amount of data needed to parametrize the behavior of the system does not scale with the size of the network but rather with d, the size of the localized region, which is usually much smaller than n. To the best of our knowledge, this is the first such result. As we show next, this will prove key in extending data-driven SLS to the distributed setting.

V. DISTRIBUTED AND LOCALIZED DATA-DRIVEN MPC

In this section we make use of the results on localized data-driven SLS from previous sections and apply them to reformulate the MPC subproblem (4). We provide a distributed and localized algorithmic solution that does not scale with the size of the network. Lastly, we comment on the theoretical guarantees of this data-driven DLMPC (D^{3}LMPC) approach in terms of convergence, recursive feasibility and asymptotic stability.

A. SYSTEM LEVEL SYNTHESIS BASED DATA-DRIVEN MPC

In light of Theorem 2, we can write the MPC subproblem (4) in terms of the variable $\mathbf{G}$ and localized Hankel matrices $H_L([\tilde{\mathbf{x}}]_{\mathbf{in}_i(d+2)}, [\tilde{\mathbf{u}}]_{\mathbf{in}_i(d+2)})$. To do this, we proceed as in reformulation (9) and rely on the equivalence between standard and data-driven SLS parametrizations, i.e., $\mathbf{\Phi} = H_L(\tilde{\mathbf{x}}, \tilde{\mathbf{u}})\mathbf{G} \Leftrightarrow Z_{AB}\mathbf{\Phi} = I$. We make use of Lemma 2 to recast the locality constraints (3) into local affine constraints. Hence, we rewrite problem (4) as

$$\begin{aligned} \underset{\mathbf{\Phi}, \{\mathbf{G}^i\}_{i=1}^N}{\text{minimize}} \quad & f(\mathbf{\Phi}x_0) \\ \text{s.t.} \quad & x_0 = x(t), \ \mathbf{\Phi}_x x_0 \in \mathcal{X}, \ \mathbf{\Phi}_u x_0 \in \mathcal{U}, \\ & [\mathbf{\Phi}^i]_{\mathbf{in}_i(d)} = H_L([\tilde{\mathbf{x}}]_{\mathbf{in}_i(d)}, [\tilde{\mathbf{u}}]_{\mathbf{in}_i(d+1)})\mathbf{G}^i, \\ & \mathbf{G}^i \text{ satisfies (20) } \forall i = 1, \ldots, N. \end{aligned} \tag{21}$$

By introducing duplicate decision variables $\mathbf{\Phi}$ and $\mathbf{G}$, and rewriting the achievability and localization constraints in terms of the variable $\mathbf{G}$ by means of (20), the problem now enjoys a partially separable structure [22]. In what follows, we make such structure explicit and take advantage of it to distribute the problem across different subsystems via ADMM.

B. A DISTRIBUTED SUBPROBLEM SOLUTION VIA ADMM

To rewrite (21) and take advantage of the separability features in Assumption 1, we introduce the concept of row-wise separability and column-wise separability:

Definition 6: Given the partition $\{\mathfrak{r}_1, \ldots, \mathfrak{r}_k\}$, a functional/set is *row-wise separable* if:
- For a functional, $g(\mathbf{\Phi}) = \sum_{i=1}^k g_i(\mathbf{\Phi}(\mathfrak{r}_i, :))$ for some functionals g_i for $i = 1, \ldots, k$.
- For a set, $\mathbf{\Phi} \in \mathcal{P}$ if and only if $\mathbf{\Phi}(\mathfrak{r}_i, :) \in \mathcal{P}_i \ \forall i$ for some sets $\mathcal{P}_i$ for $i = 1, \ldots, k$, where $\mathcal{P} := \otimes_{i=0}^N \mathcal{P}_i$.

An analogous definition exists for *column-wise separable* functionals and sets, where the partition $\{\mathfrak{c}_1, \ldots, \mathfrak{c}_k\}$ entails the columns of $\mathbf{\Phi}$, i.e., $\mathbf{\Phi}(:, \mathfrak{c}_i)$.

By Assumption 1 the objective function and the safety/saturation constraints in (21) are row-separable in terms of $\mathbf{\Phi}$. At the same time, the achievability and locality constraints (19), (20) are column-separable in terms of $\mathbf{G}$. Hence, the data-driven MPC subroutine becomes:

$$\begin{aligned} \underset{\mathbf{\Phi},\mathbf{\Psi},\{\mathbf{G}^i\}_{i=1}^N}{\text{minimize}} \quad & \sum_{i=1}^{N} f^i([\mathbf{\Phi}]_i[x_0]_{\mathbf{in}_i(d)}) \\ \text{s.t.} \quad & x_0 = x(t),\ \mathbf{G}^i \text{ satisfies (20)}, \\ & [\mathbf{\Phi}_x]_i[x_0]_{\mathbf{in}_i(d)} \in \mathcal{X}^i,\ [\mathbf{\Phi}_u]_i[x_0]_{\mathbf{in}_i(d)} \in \mathcal{U}^i, \\ & [\mathbf{\Psi}^i]_{\mathbf{in}_i(d)} = H_L([\tilde{\mathbf{x}}]_{\mathbf{in}_i(d)}, [\tilde{\mathbf{u}}]_{\mathbf{in}_i(d+1)})\mathbf{G}^i \\ & \qquad\qquad\qquad\qquad \forall i = 1, \ldots, N, \\ & \mathbf{\Phi} = \mathbf{\Psi}. \end{aligned} \tag{22}$$

Notice that the objective function and the constraints decompose across the d-localized neighborhoods of each subsystem i. Given this structure, we can make use of ADMM to decompose this problem into row-wise local subproblems in terms of $[\mathbf{\Phi}]_i$ and column-wise local subproblems in terms of $\mathbf{G}^i$ and $\mathbf{\Psi}^i$, both of which can also be parallelized across the subsystems. The ADMM subroutine iteratively updates the variables as

$$[\mathbf{\Phi}]_i^{\{k+1\}} = \left\{ \begin{array}{ll} \underset{[\mathbf{\Phi}]_i}{\text{argmin}} & f^i([\mathbf{\Phi}]_i[x_0]_{\mathbf{in}_i(d)}) \\ & \qquad + \frac{\rho}{2}\left\| g^i(\mathbf{\Phi}, \mathbf{\Psi}^{\{k\}}, \mathbf{\Lambda}^{\{k\}}) \right\|_F^2 \\ s.t. & [\mathbf{\Phi}_x]_i[x_0]_{\mathbf{in}_i(d)} \in \mathcal{X}^i, \\ & [\mathbf{\Phi}_u]_i[x_0]_{\mathbf{in}_i(d)} \in \mathcal{U}^i, \quad x_0 = x(t). \end{array} \right\} \tag{23a}$$

$$[\mathbf{\Psi}^i]_{\mathbf{in}_i(d)}^{\{k+1\}} = \left\{ \begin{array}{ll} \underset{[\mathbf{\Psi}^i]_{\mathbf{in}_i(d)}, \mathbf{G}^i}{\text{argmin}} & \| g^{\mathbf{in}_i(d)}\left([\mathbf{\Phi}^i]^{\{k+1\}}, [\mathbf{\Psi}^i], [\mathbf{\Lambda}^i]^{\{k\}}\right) \|_F^2 \\ s.t. & [\mathbf{\Psi}^i]_{\mathbf{in}_i(d)} = [H_L(\tilde{\mathbf{x}}, \tilde{\mathbf{u}})]_{\mathbf{in}_i(d)}\mathbf{G}^i, \\ & \mathbf{G}^i \text{ satisfies (20).} \end{array} \right\} \tag{23b}$$

$$[\mathbf{\Lambda}]_i^{\{k+1\}} = g^i(\mathbf{\Phi}^{\{k+1\}}, \mathbf{\Psi}^{\{k+1\}}, \mathbf{\Lambda}^{\{k\}}) \tag{23c}$$

where we define

$$g^*(\mathbf{\Phi}, \mathbf{\Psi}, \mathbf{\Lambda}) := [\mathbf{\Phi}]_* - [\mathbf{\Psi}]_* + [\mathbf{\Lambda}]_*$$

where $*$ denotes a (collection of) subsystem(s), and

$$[H_L(\tilde{\mathbf{x}}, \tilde{\mathbf{u}})]_{\mathbf{in}_i(d)} := H_L([\tilde{\mathbf{x}}]_{\mathbf{in}_i(d)}, [\tilde{\mathbf{u}}]_{\mathbf{in}_i(d+1)}).$$

We note that to solve this subroutine, and in particular optimization (23b), each subsystem only needs to collect trajectory of states and control actions of subsystems that are at most $d+2$ hops away. This subroutine thus constitutes a distributed and localized solution. We also emphasize that the trajectory length only needs to scale with the d-localized system size instead of the global system size. The algorithm is given in Algorithm 1.

Algorithm 1: Subsystem's i Implementation of D^{3}LMPC.

Input: tolerance parameters $\epsilon_p, \epsilon_d > 0$, Hankel matrices $[H_L(\tilde{\mathbf{x}}, \tilde{\mathbf{u}})]_{\mathbf{in}_i(d+1)}$ constructed from arbitrarily generated PE trajectories $\tilde{\mathbf{x}}_{\mathbf{in}_i(d+1)}, \tilde{\mathbf{u}}_{\mathbf{in}_i(d+1)}$.

1: Measure local state $[x_0]_i$, $k \leftarrow 0$.
2: Share measurement $[x_0]_i$ with $\mathbf{out}_i(d)$ and receive $[x_0]_j$ from $j \in \mathbf{in}_i(d)$.
3: Solve optimization problem (23a).
4: Share $[\mathbf{\Phi}]_i^{\{k+1\}}$ with $\mathbf{out}_i(d)$. Receive the corresponding $[\mathbf{\Phi}]_j^{\{k+1\}}$ from $\mathbf{in}_i(d)$ and construct $[\mathbf{\Phi}^i]_{\mathbf{in}_i(d)}^{\{k+1\}}$.
5: Perform update (23b).
6: Share $[\mathbf{\Psi}^i]_{\mathbf{in}_i(d)}^{\{k+1\}}$ with $\mathbf{out}_i(d)$. Receive the corresponding $[\mathbf{\Psi}^j]_{\mathbf{in}_i(d)}^{\{k+1\}}$ from $j \in \mathbf{in}_i(d)$ and construct $[\mathbf{\Psi}]_i^{\{k+1\}}$.
7: Perform update (23c).
8: **if** $\|[\mathbf{\Psi}]_i^{\{k+1\}} - [\mathbf{\Phi}]_i^{\{k+1\}}\|_F \leq \epsilon_p$ and $\|[\mathbf{\Phi}]_i^{\{k+1\}} - [\mathbf{\Phi}]_i^{\{k\}}\|_F \leq \epsilon_d$**:**
   Apply computed control action: $[u_0]_i = \mathcal{H}_L([\tilde{\mathbf{u}}]_{\mathbf{in}_i(d)})\mathbf{G}^i[x_0]_{\mathbf{in}_i(d)}$, and return to step 1.
   else:
   Set $k \leftarrow +1$ and return to step 3.

C. THEORETICAL GUARANTEES

It is worth noting that this reformulation is equivalent to the closed-loop DLMPC also introduced in [23], with the achievability constraint $Z_{AB}\mathbf{\Phi} = I$ replaced by the data-driven parametrization in terms of $\mathbf{G}$ and the Hankel matrix H_L. For this reason, guarantees derived for the DLMPC formulation [24] directly apply to problem (21) when the constraint sets $\mathcal{X}$ and $\mathcal{U}$ are polytopes.

1) CONVERGENCE

Algorithm 1 relies on ADMM to separate both row, and column-wise computations, in terms of $\mathbf{\Phi}$ and $\mathbf{G}$ respectively. Each of these is then distributed into the subsystems in the network, and a communication protocol is established to ensure the ADMM steps are properly followed. Hence, one can guarantee the convergence of the data-driven version of DLMPC in the same way that convergence of model-based DLMPC is shown: by leveraging the convergence result of ADMM in [19]. For additional details see Lemma 2 in [24].

2) RECURSIVE FEASIBILITY

Recursive feasibility for formulation (9) is guaranteed by means of a localized maximally invariant terminal set $\mathcal{X}_T$. This set can be computed in a distributed manner and with local information only as described in [24]. In particular, a closed-loop map $\mathbf{\Phi}$ for the unconstrained localized closed-loop system has to be computed. In the model-based SLS problem with quadratic cost, a solution exists for the infinite-horizon case [30], which can be done in a distributed manner and with local information only. When no model is available, the same problem can be solved using the localized data-driven SLS approach introduced in Section IV with a finite-horizon approximation, which also allows for a distributed synthesis with only local data. The length of the time horizon chosen to solve the localized data-driven SLS problem might slightly impact the conservativeness of the terminal set, but since the conservatism in the FIR approach decays exponentially with the horizon length, this harm in performance is not expected to be substantial. Once $\mathbf{\Phi}$ for the unconstrained localized closed-loop system has been computed, Algorithm 2 in [24] can be used to synthesize this terminal set in a distributed and localized manner. Therefore, a terminal set that guarantees recursive feasibility for D^{3}LMPC can be computed in a distributed manner offline using only local information and without the need for a system model.

3) ASYMPTOTIC STABILITY

Similar to recursive feasibility, asymptotic stability for the D^{3}LMPC problem (21) is directly inherited from the asymptotic stability guarantee for model-based DLMPC. In particular, adding a terminal cost based on the terminal set previously described is a sufficient condition to guarantee asymptotic stability of the DLMPC problem [24]. Moreover, such cost can be incorporated in the D^{3}LMPC formulation in the same way as in the model-based DLMPC problem, and the structure of the resulting problem is analogous. This terminal cost introduces coupling among subsystems, but the coupling can be dealt with by solving step 3 of Algorithm 1 via local consensus. Notice that since this step is written in terms of $\mathbf{\Phi}$, Algorithm 3 in [24] can be used to handle this coupling. For additional details the reader is referred to [24].

VI. SIMULATION EXPERIMENTS

We demonstrate through experiments that the D^{3}LMPC controller using only local data performs as well as a model-based DLMPC controller. We also show that for our algorithm, both runtime and the dimension of the data needed scale well with the size of the network. All the code needed to reproduce the experiments can be found at https://github.com/unstable-zeros/dl-mpc-sls.

A. SETUP

We evaluate the performance and scalability of our algorithm on a system composed of a chain of subsystems, i.e., that $\mathcal{E} = \{(i, i+1), (i+1, i),\ i = 1, \ldots, N-1\}$. Each subsystem i has a 2-dimensional state $[\theta_i, \omega_i]^\intercal$, where θ_i and ω_i are respectively the phase angle deviation and frequency deviation of the subsystem. We assume that each subsystem takes a scalar control action u_i. We consider the subsystem dynamics as in [23], obtained from discrete-time linearized swing-equations:

$$\begin{bmatrix} \theta(t+1) \\ \omega(t+1) \end{bmatrix}_i = \sum_{j \in \mathbf{in}_i(1)} [A]_{ij} \begin{bmatrix} \theta(t) \\ \omega(t) \end{bmatrix}_i + [B]_i [u]_i, \tag{24}$$

where

$$[A]_{ii} = \begin{bmatrix} 1 & \Delta t \\ -\frac{k_i}{m_i}\Delta t & 1 - \frac{d_i}{m_i}\Delta t \end{bmatrix}, \quad [A]_{ij} = \begin{bmatrix} 0 & 0 \\ \frac{k_{ij}}{m_i}\Delta t & 0 \end{bmatrix},$$

and $B_{ii} = [0\ 1]^\top$ for all i. The parameters m_i, d_i, k_{ij} are sampled uniformly at random from the intervals $[0, 2], [0.5, 1], [1, 1.5]$, respectively and $k_i := \sum_{j \in \mathbf{in}_i(1)} k_{ij}$. Discretization time is set to be $\Delta t = 0.2$. The goal of the controller is to minimize the LQR cost with $Q = I, R = I$.

To show the optimality of our approach in Section VI-B, we consider a base system with 64 subsystems. To demonstrate the scalability of our method, we consider systems of varying sizes for our experiment in Section VI-C. Unless mentioned otherwise, for all the experiments, we consider a locality region of size $d = 2$, use a planning horizon of $T = 5$ steps and simulate the system forward for 30 steps.

B. OPTIMAL PERFORMANCE

We evaluate the performance of D^{3}LMPC (Algorithm 1) on the system described in Section VI-A. First, we show that the trajectory given by the D^{3}LMPC algorithm matches the trajectory generated by a model-based DLMPC solved with a centralized solver [31], [32]. Notice that the model-based DLMPC solves the optimization problem (9) with perfect knowledge of the system dynamics, while the D^{3}LMPC algorithm (Algorithm 1) only has access to local past trajectories. Due to space constraints, we only show the state trajectory of the first subsystem (Fig. 1). We observe that the trajectory generated by our controller matches the trajectory of the optimal controller with the same locality region size. Further, the optimal cost for both schemes is the same up to numerical precision. This confirms that the D^{3}LMPC algorithm (Algorithm 1) can synthesize optimal controllers using only local trajectory data and no knowledge of the system dynamics.

We further highlight the relevance of locality region size on the optimality of the solution. The size of the locality region d can be seen as a design parameter in Algorithm 1 that allows one to tradeoff between computation complexity and performance of the controller. In Fig. 2, we show how the optimal cost varies with the size of the locality region on the same system. As the size of the locality region grows, the optimal cost decreases. This matches the intuition that by allowing each subsystem to influence more subsystems, and as more information is made available to each subsystem, controllers of better quality can be synthesized. We note that the performance improvement by increasing the locality region size is the most significant when the locality region is small. In Fig. 3 we simultaneously show how much the trajectory length needs to grow with the size of the locality region to satisfy the persistence of excitation condition for applying Willem's Fundamental Lemma. We note that the growth in the necessary trajectory length not only means longer trajectory needs to be collected, but also means that the size of the optimization problem grows, thus incurring higher computation complexity

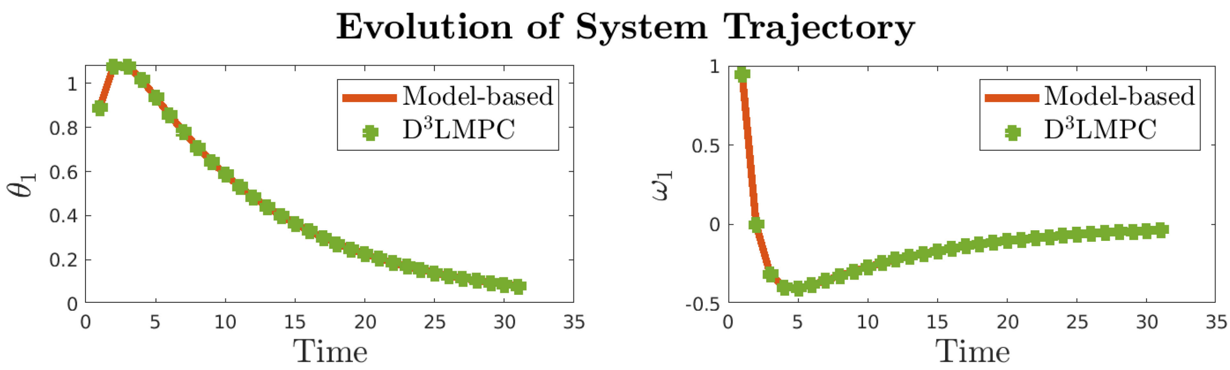


FIGURE 1. The trajectory generated by the model-based DLMPC algorithm (solid orange line) vs. the trajectory generated by D^{3}LMPC (green circles). We observe that the two coincide.

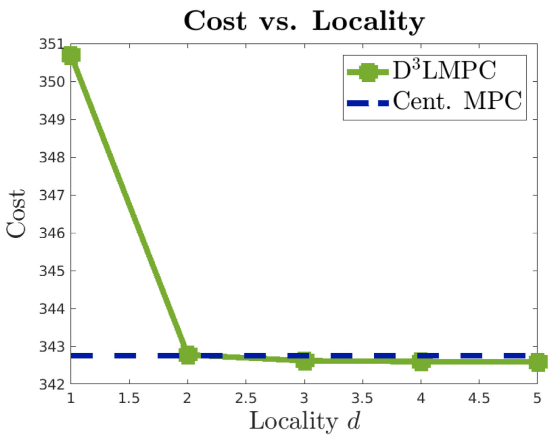


FIGURE 2. (Green) The cost achieved by the optimal controller versus the size of the locality region for the system response. (Blue) The cost achieved by a centralized MPC controller, i.e., that it has no locality constraints. We observe that cost for the distributed controllers decreases as the size of the locality region grows and approaches the cost of the centralized controller.

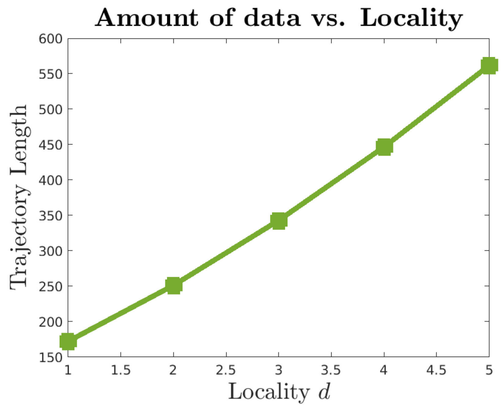


FIGURE 3. The growth of necessary length of collected trajectory versus the size of the locality region of system response. The trajectory length grows with the size of the locality region.

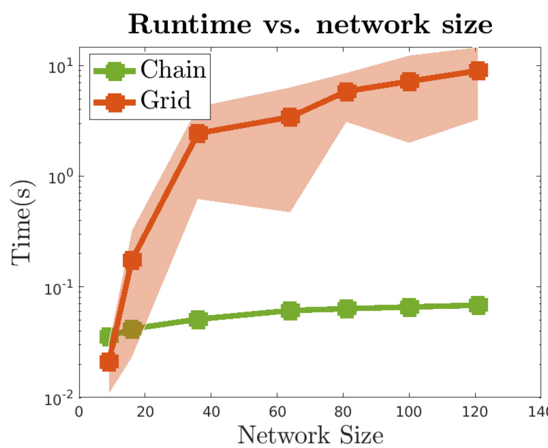


FIGURE 4. The average per-step per-subsystem runtime of the MPC algorithm. The solid line shows the average runtime over 10 randomly generated systems, and the circles represent the runtime for each of the 10 randomly generated instances for each system size.

for each optimization step. Hence, the choice of an optimal d heavily depends on the specific application considered.

C. SCALABILITY

First, we show that the runtime of our method scales well with the size of the global system. For this experiment only, we also apply our algorithm to sparsely connect square grids where the subsystems follow the same dynamics (24) as the chain system. The grid topology is generated randomly, with each subsystem having a $p = 0.6$ chance to be connected with each of its neighboring subsystems[6] We consider systems composed of 9, 16, 36, 64, 81, 100, and 121 subsystems. For each system size, we randomly generate 10 different systems of each type and report their respective average computation time per MPC step.[7] The result is shown in Fig. 4. For the chain systems, we note that the runtime only increased $2\times$ while the size of the system has increased more than $12\times$. The runtime scales less well for the more densely-connected grid topology. This of course would be true for any distributed

[6]We also perform rejection sampling to ensure that the resulting topology is strongly connected.

[7]Runtime is measured after one warmstart iteration. The optimization problems were solved using the Gurobi [32] optimizer on a personal desktop computer with an 8-core Intel i7 processor.

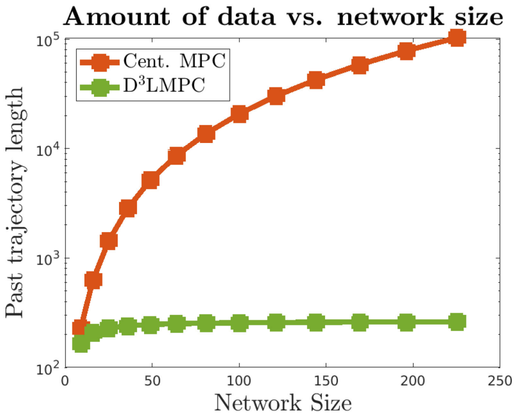


FIGURE 5. Length of necessary trajectory length versus network size. Note that this is plotted on a semilog axis. Our distributed approach requires much shorter trajectory over a centralized data-driven approach.

MPC algorithm, data-driven or not, due to the denser interconnections between local subsystems. Nevertheless, in both cases, we show that the growth of the runtime flattens as the size of the network grows, suggesting that our method scales well on sparsely connected systems. This trend has previously been observed with ADMM schemes for MPC [33]. Next, we show that the length of the trajectory that needs to be collected for D^{3}LMPC grows much more slowly than that for a centralized data-driven method that does not exploit the locality structure of the problem (equivalent to solving the SLS problem with constraints (12) instead of (20)). The result is shown in Fig. 5. We note that our method requires less data (length of the trajectory) than a centralized data-driven approach, with the saving bigger for larger systems.[8]

VII. CONCLUSION

In this paper we define and analyze a data-driven Distributed and Localized Model Predictive Control (D^{3}LMPC) scheme. This approach can synthesize optimal localized control policies using only local communication and requires no knowledge of the system model. We base our results on the data-driven SLS approach [14], and extend this framework to allow for locality constraints. We then use these results to provide an alternative data-driven synthesis for the DLMPC algorithm [22] by exploiting the separability of the problem via ADMM. The resulting algorithm enjoys the same scalability properties as model-based DLMPC [23] and only need trajectory data that scales with the size of the d-localized neighborhood. Moreover, recursive feasibility and stability guarantees that exist for model-based DLMPC [24] directly apply to this framework.

The work presented here is, to the best of our knowledge, the first fully distributed and localized data-driven MPC approach that achieves globally optimal performance with local information collection and communication among subsystems, and that requires collecting trajectory data of length independent of the global system size. In future work, we will look to relax the assumptions made in this paper regarding noiseless data/systems, and localizability. To do so, we will integrate the results of this paper with robust SLS-based methods [23], [34], [35] and the centralized robust data-driven SLS results of [14], which will allow for all of the same computational benefits to be enjoyed in the case where systems are only approximately localizable and/or noisy. Another interesting direction for future work is to apply recent extensions of the Fundamental Lemma [21] to relax the controllability condition and further reduce the length of the data trajectories needed for the D^{3}LMPC algorithm.

[8] We report that mean data trajectory length over all subsystems. Thus, the reported trajectory length for D^{3}LMPC increases at first as the boundary subsystems require less data.

REFERENCES

[1] N. Matni, A. Proutiere, A. Rantzer, and S. Tu, "From self-tuning regulators to reinforcement learning and back again," in *Proc. IEEE 58th Conf. Decis. Control*, 2019, pp. 3724–3740.

[2] H. Hjalmarsson, M. Gevers, S. Gunnarsson, and O. Lequin, "Iterative feedback tuning: Theory and applications," *IEEE Control Syst. Mag.*, vol. 18, no. 4, pp. 26–41, Aug. 1998.

[3] M. Fazel, R. Ge, S. Kakade, and M. Mesbahi, "Global convergence of policy gradient methods for the linear quadratic regulator," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 1467–1476.

[4] H. Mohammadi, M. Soltanolkotabi, and M. R. Jovanović, "On the linear convergence of random search for discrete-time LQR," *IEEE Control Syst. Lett.*, vol. 5, no. 3, pp. 989–994, Jul. 2021.

[5] S. Bradtke, B. Ydstie, and A. Barto, "Adaptive linear quadratic control using policy iteration," in *Proc. Amer. Control Conf.*, 1994, pp. 3475–3479.

[6] C. De Persis and P. Tesi, "Formulas for data-driven control: Stabilization, optimality, and robustness," *IEEE Trans. Autom. Control*, vol. 65, no. 3, pp. 909–924, Mar. 2020.

[7] J. Jiao, H. J. van Waarde, H. L. Trentelman, M. K. Camlibel, and S. Hirche, "Data-driven output synchronization of heterogeneous leader-follower multi-agent systems," in *Proc. 60th IEEE Conf. Decis. Control*, 2021, pp. 466–471, doi: 10.1109/CDC45484.2021.9683073.

[8] T. R. Steentjes, M. Lazar, and P. M. Van den Hof, "$\mathcal{H}_\infty$ performance analysis and distributed controller synthesis for interconnected linear systems from noisy input-state data," in *Proc. 60th IEEE Conf. Decis. Control*, 2021, pp. 3717–3722.

[9] I. Markovsky and F. Dörfler, "Behavioral systems theory in data-driven analysis, signal processing, and control," *Annu. Rev. Control*, vol. 52, pp. 42–64, 2021.

[10] F. Dörfler, P. Tesi, and C. D. Persis, "On the certainty-equivalence approach to direct data-driven LQR design," 2021, *arXiv:2109.06643*.

[11] F. Dörfler, J. Coulson, and I. Markovsky, "Bridging direct & indirect data-driven control formulations via regularizations and relaxations," *IEEE Trans. Autom. Control*, 2022, doi: 10.1109/TAC.2022.3148374.

[12] J. Coulson, J. Lygeros, and F. Dorfler, "Distributionally robust chance constrained data-enabled predictive control," *IEEE Trans. Autom. Control*, 2021, doi: 10.1109/TAC.2021.3097706.

[13] J. Berberich, J. Köhler, M. A. Müller, and F. Allgöwer, "Data-driven tracking MPC for changing setpoints," *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 6923–6930, 2020.

[14] A. Xue and N. Matni, "Data-driven system level synthesis," in *Proc. Mach. Learn. Res.*, vol. 144, pp. 1–12, 2021.

[15] A. Allibhoy and J. Cortés, "Data-based receding horizon control of linear network systems," *IEEE Control Syst. Lett.*, vol. 5, no. 4, pp. 1207–1212, Oct. 2021.

[16] S. Alemzadeh, S. Talebi, and M. Mesbahi, "D3PI: Data-driven distributed policy iteration for homogeneous interconnected systems," 2021, *arXiv:2103.11572*.

[17] S. Muntwiler, K. P. Wabersich, L. Hewing, and M. N. Zeilinger, "Data-driven distributed stochastic model predictive control with closed-loop chance constraint satisfaction," in *Proc. Eur. Control Conf.*, 2021, pp. 210–215, doi: 10.23919/ECC54610.2021.9655214.

[18] Y. R. Sturz, E. L. Zhu, U. Rosolia, K. H. Johansson, and F. Borrelli, “Distributed learning model predictive control for linear systems,” in *Proc. 59th IEEE Conf. Decis. Control*, 2020, pp. 4366–4373.

[19] L. Furieri, B. Guo, A. Martin, and G. Ferrari-Trecate, “Near-optimal design of safe output feedback controllers from noisy data,” 2021, *arXiv:2105.10280*.

[20] S. Boyd, N. Parikh, E. Chu, B. Peleato, 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.

[21] Y. Yu, S. Talebi, H. J. van Waarde, U. Topcu, M. Mesbahi, and B. Açıkmeşe, “On controllability and persistency of excitation in data-driven control: Extensions of Willems’ fundamental lemma,” in *Proc. 60th IEEE Conf. Decis. Control* 2021, pp. 6485–6490, doi: 10.1109/CDC45484.2021.9682952.

[22] C. A. Alonso and N. Matni, “Distributed and localized closed-loop model predictive control via system level synthesis,” in *Proc. 59th IEEE Conf. Decis. Control*, 2020, pp. 5598–5605.

[23] C. A. Alonso, S. Li, N. Matni, and J. Anderson, “Distributed and localized model predictive control. Part I: Synthesis and implementation,” 2021, *arXiv:2110.07010*.

[24] C. A. Alonso, S. Li, N. Matni, and J. Anderson, “Distributed and localized model predictive control. Part II: Theoretical guarantees,” 2021, *arXiv:2203.00780*.

[25] Y. Wang, N. Matni, and J. C. Doyle, “Separable and localized system-level synthesis for large-scale systems,” *IEEE Trans. Autom. Control*, vol. 63, no. 12, pp. 4234–4249, Dec. 2018.

[26] J. Anderson, J. C. Doyle, S. H. Low, and N. Matni, “System level synthesis,” *Annu. Rev. Control*, vol. 47, pp. 364–393, 2019.

[27] J. C. Willems, P. Rapisarda, I. Markovsky, and B. L. De Moor, “A note on persistency of excitation,” *Syst. Control Lett.*, vol. 54, no. 4, pp. 325–329, 2005.

[28] Y.-S. Wang, N. Matni, and J. C. Doyle, “A system-level approach to controller synthesis,” *IEEE Trans. Autom. Control*, vol. 64, no. 10, pp. 4079–4093, Oct. 2019.

[29] J. Willems and J. Polderman, *Introduction to Mathematical Systems Theory: A Behavioral Approach* (Texts in Applied Mathematics). New York, NY, USA: Springer, 1997.

[30] J. Yu, Y.-S. Wang, and J. Anderson, “Localized and distributed $\mathcal{H}_2$ state feedback control,” *Amer. Control Conf.*2021, pp. 2732–2738, doi: 10.23919/ACC50511.2021.9483301.

[31] M. Grant and S. Boyd, “CVX: Matlab software for disciplined convex programming, version 2.1,” 2014. [Online]. Available: http://cvxr.com/cvx

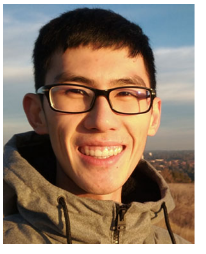
[32] L. Gurobi Optimization, “Gurobi optimizer reference manual,” 2021. [Online]. Available: http://www.gurobi.com

[33] C. Conte, T. Summers, M. N. Zeilinger, M. Morari, and C. N. Jones, “Computational aspects of distributed optimization in model predictive control,” in *Proc. 51st IEEE Conf. Decis. Control*, 2012, pp. 6819–6824.

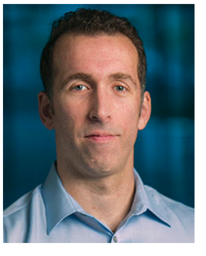
[34] N. Matni, Y.-S. Wang, and J. Anderson, “Scalable system level synthesis for virtually localizable systems,” in *Proc. IEEE 56th Annu. Conf. Decis. Control*, 2017, pp. 3473–3480.

[35] S. Chen, N. Matni, M. Morari, and V. M. Preciado, “System level synthesis-based robust model predictive control through convex inner approximation,” 2021, *arXiv:2111.05509*.

CARMEN AMO ALONSO (Student Member, IEEE) received the B.Sc. degree in aerospace engineering from the Polytechnic University of Madrid, Madrid, Spain, in 2016, and the M.Sc. degree in space engineering in 2017 from the California Institute of Technology, Pasadena, CA, USA, where she is currently working toward the Ph.D. degree in control and dynamical systems. She is an AI4Science Amazon/Caltech Fellow. Her research interests include distributed optimal control, convex optimization, and parallel programming.



FENGJUN YANG received the B.A. degree in computer science from Swarthmore College, Swarthmore, PA, USA, and the M.Sc. degree in aerospace engineering from Stanford University, Stanford, CA, USA. He is currently working toward the Ph.D. degree with the Department of Computer and Information Science, University of Pennsylvania, Philadelphia, PA, USA. His research interests include distributed control, robotics, and multiagent systems.



NIKOLAI MATNI (Member, IEEE) received the Ph.D. degree in control and dynamical systems from Caltech in June 2016. He is currently an Assistant Professor with the Department of Electrical and Systems Engineering, University of Pennsylvania, Philadelphia, PA, USA. Prior to joining Penn, he was a Postdoctoral Scholar with the Department of Electrical Engineering and Computer Sciences, University of California Berkeley, Berkeley, CA, USA. He has also held a position as a Postdoctoral Scholar with the Department of Computing and Mathematical Sciences, California Institute of Technology, Pasadena, CA, USA. He was the recipient of the NSF CAREER Award in 2021, Google Research Scholar Award in 2021, 2021 IEEE CSS George S. Axelby Award, IEEE ACC 2017 Best Student Paper Award (as co-advisor), and IEEE CDC 2013 Best Student Paper Award.