

# Data-Based Receding Horizon Control of Linear Network Systems

Ahmed Allibhoy<sup>ID</sup> and Jorge Cortés<sup>ID</sup>, *Fellow, IEEE*

**Abstract**—We propose a distributed data-based predictive control scheme to stabilize a network system described by linear dynamics. Agents cooperate to predict the future system evolution without knowledge of the dynamics, relying instead on learning a data-based representation from a single sample trajectory. We employ this representation to reformulate the finite-horizon Linear Quadratic Regulator problem as a network optimization with separable objective functions and locally expressible constraints. We show that the controller resulting from approximately solving this problem using a distributed optimization algorithm in a receding horizon manner is stabilizing. We validate our results through numerical simulations.

**Index Terms**—Data-based control, network systems, predictive control of linear systems.

## I. INTRODUCTION

WITH the growing complexity of engineering systems, data-based methods in control theory are becoming increasingly popular, particularly for systems where it is too difficult to develop models from first principles and parameter identification is impractical or too costly. An important class of such systems are network systems, which arise in many applications such as neuroscience, power systems, traffic management, and robotics. Without a system model, agents must use sampled data to characterize the network behavior. However, the decentralized nature of the system means that agents only have access to information that can be measured locally, and must coordinate with one another to predict the network response and decide their control actions. These observations motivate the focus here on distributed data-based control of network systems with linear dynamics.

*Literature review:* Distributed control of network systems is a burgeoning area of research, see [1]–[3] and references therein. In general, designing optimal controllers for network systems is an NP-hard problem, but under certain conditions optimal distributed controllers for linear systems can be obtained as the solution to a convex program [4]. When these conditions do not hold, suboptimal controllers can be obtained by convex relaxations [5], [6] or convex restrictions [7] of

Manuscript received June 9, 2020; revised August 25, 2020; accepted August 29, 2020. Date of publication September 2, 2020; date of current version September 22, 2020. This work was supported by ARO-W911NF-18-1-0213. Recommended by Senior Editor M. Guay. (Corresponding author: Ahmed Allibhoy.)

The authors are with the Department of Mechanical and Aerospace Engineering, UC San Diego, San Diego, CA 92093 USA (e-mail: aallibho@ucsd.edu; cortes@ucsd.edu).

Digital Object Identifier 10.1109/LCSYS.2020.3021050

the original problem. Although these methods produce distributed controllers, the computation of the controller itself is typically done offline, in a centralized manner, and requires knowledge of the underlying system model. Reinforcement learning (RL) is an increasingly popular approach for controlling robots [8] and multi-agent systems [9]. However, RL approaches typically require a very large number of samples to perform effectively [10] and their complexity makes it difficult to get stability, safety, and robustness guarantees as is standard with other control approaches. For applications where safety assurances are required, model predictive control (MPC) is widely used since performance and safety constraints can be directly incorporated into an optimization problem that is solved online. Several distributed MPC formulations are available for multi-agent systems where the dynamics of the agents are coupled, such as [11], [12] where each agent implements a control policy minimizing its own objective while accounting for network interactions locally, or [13] where agents cooperate to minimize a system-wide objective using a network optimization algorithm. Data-based approaches to predictive control have also been proposed. System identification [14] is often leveraged to learn a parameterized model which can then be used with any of the MPC formulations previously mentioned. Methods for implementing a controller directly from sampled data without any intermediate identification also exist. The fundamental lemma from behavioral systems theory [15], which characterizes system trajectories from a single sample trajectory, has recently gained attention in the area of data-based control [16]–[18], and has been used for predictive control in the recently developed DeePC framework [19], [20]. This letter here extends the DeePC framework to network systems where each node only has partial access to the data.

*Statement of contributions:* We develop distributed data-based feedback controllers for network systems.<sup>1</sup> A

<sup>1</sup>Throughout this letter, we make use of the following notation. Given integers,  $a, b \in \mathbb{Z}$  with  $a < b$ , let  $[a, b] = \{a, a+1, \dots, b\}$ . Let  $\mathcal{G} = (\mathcal{V}, \mathcal{E})$  be an undirected graph with  $N$  nodes, where  $\mathcal{V} = [1, N]$  and  $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$ . The neighbors of  $i \in \mathcal{V}$  are  $\mathcal{N}_i = \{j : (i, j) \in \mathcal{E}\}$ . Given  $S = \{s_1, s_2, \dots, s_M\} \subseteq [1, N]$  and a vector  $x = [x_1^\top, x_2^\top, \dots, x_N^\top]^\top$ , we denote  $x_S = [x_{s_1}^\top, x_{s_2}^\top, \dots, x_{s_M}^\top]^\top$ . For  $x_i \in \mathbb{R}^{d_i}$  with  $i \in [1, K]$ , we let  $\text{col}(x_1, x_2, \dots, x_K) = [x_1^\top, x_2^\top, \dots, x_K^\top]^\top$ . For positive semidefinite  $Q \in \mathbb{R}^{n \times n}$ , we denote  $\|x\|_Q = \sqrt{x^\top Q x}$ . For  $M \in \mathbb{R}^{n \times m}$ , we denote by  $M^\dagger$  its Moore-Penrose pseudoinverse. The Hankel matrix of a signal  $w : [0, T] \rightarrow \mathbb{R}^k$  with  $t \leq T$  block rows is the  $kt \times (T-t+1)$  matrix

$$\mathcal{H}_t(w) = \begin{bmatrix} w(0) & w(1) & \dots & w(T-t) \\ w(1) & w(2) & \dots & w(T-t+1) \\ \vdots & \vdots & \ddots & \vdots \\ w(t-1) & w(t) & \dots & w(T-1) \end{bmatrix}.$$

Given two signals  $v_1 : [0, T-1] \rightarrow \mathbb{R}^{k_1}$  and  $v_2 : [0, T-1] \rightarrow \mathbb{R}^{k_2}$ , let  $v = \text{col}(v_1, v_2)$  be the signal where  $v(t) = v_1(t)$  for  $0 \leq t < T$ , and  $v(t) = v_2(t-T)$  for  $T \leq t < 2T$ .

group of agents whose state evolves according to unknown coupled linear dynamics each have access to their own state and those of their neighbors in some sample trajectory. Their collective objective is to drive the network state to the origin while minimizing a quadratic objective function without knowledge of the system dynamics. The approach we use computes the control policy online and in a distributed manner by extending the DeePC formalism to the network case. Building upon the fundamental lemma, we introduce a new distributed, data-based representation of possible network trajectories. We use this representation to pose the control synthesis as a network optimization problem, without state or input constraints, in terms of the data available to each agent. We show that this optimization problem is equivalent to the standard finite-horizon Linear Quadratic Regulation (LQR) problem and introduce a primal-dual method along with a suboptimality certificate to allow agents to cooperatively find an approximate solution. Finally, we show that the controller that results from implementing the distributed solver in a receding horizon manner is stabilizing.

## II. PRELIMINARIES

We briefly recall here basic concepts on the identifiability of Linear Time-Invariant (LTI) systems from data. Given  $t, T_d \in \mathbb{Z}_{\geq 0}$  with  $t < T_d$ , a signal  $w : [0, T_d - 1] \rightarrow \mathbb{R}^k$  is *persistently exciting of order  $t$*  if  $\text{rowrank } \mathcal{H}_t(w) = kt$ . Informally, this means that any arbitrary signal  $v : [0, t - 1] \rightarrow \mathbb{R}^k$  can be described as a linear combination of windows of width  $t$  in the signal  $w$ . A necessary condition for persistence of excitation is  $T_d \geq (k + 1)t - 1$ .

**Lemma 1 (Fundamental Lemma [15]):** Consider the LTI system  $x(t + 1) = Ax(t) + Bu(t)$ , with  $(A, B)$  controllable. Let  $u^d : [0, T_d - 1] \rightarrow \mathbb{R}^m$ ,  $x^d : [0, T_d - 1] \rightarrow \mathbb{R}^n$  be sequences such that  $w^d = \text{col}(u^d, x^d)$  is a trajectory of the system and  $u^d$  is persistently exciting of order  $n + \tau$ . Then for any pair  $u : [0, \tau - 1] \rightarrow \mathbb{R}^m$ ,  $x : [0, \tau - 1] \rightarrow \mathbb{R}^n$ ,  $w = \text{col}(u, x)$  is a trajectory of the system if and only if there exists  $g \in \mathbb{R}^{T_d - \tau - 1}$  such that  $\mathcal{H}_\tau(w^d)g = w$ .

Lemma 1 is stated here in state-space form, even though the result was originally presented in the language of behavioral systems theory. The result states that all trajectories of a controllable LTI system can be characterized by a single trajectory if the corresponding input is persistently exciting of sufficiently high order, obviating the need for a model or parameter estimation when designing a controller.

## III. PROBLEM FORMULATION

Consider a network system described by an undirected graph  $\mathcal{G} = (\mathcal{V}, \mathcal{E})$  with  $N$  nodes. Each node corresponds to an agent with sensing, communication, and computation capabilities. Each edge corresponds to both a physical coupling and a communication link between the corresponding agents. A subset of the nodes  $S \subset \mathcal{V}$ , with  $|S| = M$ , also have actuation capabilities via inputs  $u_i \in \mathbb{R}^{m_i}$ . The system dynamics are then

$$x_i(t + 1) = \begin{cases} A_{ii}x_i(t) + \sum_{j \in \mathcal{N}_i} A_{ij}x_j(t) + B_i u_i & i \in S, \\ A_{ii}x_i(t) + \sum_{j \in \mathcal{N}_i} A_{ij}x_j(t) & i \notin S, \end{cases} \quad (1)$$

where  $x_i \in \mathbb{R}^{n_i}$ ,  $A_{ij} \in \mathbb{R}^{n_i \times n_j}$  and  $B_i \in \mathbb{R}^{n_i \times m_i}$ . Let  $n = \sum_{i=1}^N n_i$  and  $m = \sum_{i=1}^M m_i$  and define  $x = \text{col}(x_1, x_2, \dots, x_N) \in \mathbb{R}^n$  and  $u = \text{col}(u_1, u_2, \dots, u_M) \in \mathbb{R}^m$ . Let  $A \in \mathbb{R}^{n \times n}$  and  $B \in \mathbb{R}^{n \times m}$

be matrices so that (1) takes the compact form  $x(t + 1) = Ax(t) + Bu(t)$ .

To each node  $i \in \mathcal{V}$ , we associate an objective of the form  $J_i(x_i, u_i) = \sum_{t=0}^{T-1} \|x_i(t)\|_{Q_i} + \|u_i(t)\|_{R_i}$  when  $i \in S$  and  $J_i(x_i) = \sum_{t=0}^{T-1} \|x_i(t)\|_{Q_i}$  otherwise. Here, each  $Q_i \in \mathbb{R}^{n_i \times n_i}$  is positive semidefinite, each  $R_i \in \mathbb{R}^{m_i \times m_i}$  is positive definite,  $\text{col}(u, x)$  is a system trajectory, and  $T$  is the time horizon of trajectories being considered.

Each node wants to drive its state  $x_i$  to the origin while minimizing  $J_i$  and satisfying the constraints. The resulting network objective function is the sum of the objective functions across the nodes. Letting  $Q = \text{blkdiag}(Q_1, Q_2, \dots, Q_N) \in \mathbb{R}^{n \times n}$  and  $R = \text{blkdiag}(R_{s_1}, R_{s_2}, \dots, R_{s_M}) \in \mathbb{R}^{m \times m}$ , this objective can be written as

$$J(x, u) = \sum_{i \in S} J_i(x_i, u) + \sum_{i \in \mathcal{V} \setminus S} J_i(x_i)$$

so that  $J(x, u) = \sum_{t=0}^{T-1} \|x(t)\|_Q + \|u(t)\|_R$ . If the system starts from  $x(0) = x^0 \in \mathbb{R}^n$ , the agents' goal can be formulated as the network optimization problem:

$$\begin{aligned} & \underset{u, x}{\text{minimize}} && \sum_{t=0}^{T-1} \|x(t)\|_Q + \|u(t)\|_R \\ & \text{subject to} && x(t + 1) = Ax(t) + Bu(t), \text{ for } t \in [0, T - 1] \\ & && x(0) = x^0, x(T) = 0. \end{aligned} \quad (2)$$

Note that the agents' decisions on their control inputs are coupled through the constraints. Since  $R \succ 0$ , if (2) is feasible, its optimal state and input trajectories are unique.

A key aspect of this letter is that we consider scenarios where the system matrices  $A$  and  $B$  are unknown to the network. Instead, we assume that, for a set of given input sequences  $\{u_i^d : [0, T_d - 1] \rightarrow \mathbb{R}^{m_i}\}_{i \in S}$ , the corresponding state trajectories  $\{x_i^d : [0, T_d - 1] \rightarrow \mathbb{R}^{n_i}\}_{i \in \mathcal{V}}$  are available, and each node  $i \in \mathcal{V}$  has access to its own state trajectory as well as those of its neighbors. Actuated nodes  $i \in S$  also have access to their own input  $u_i^d$ , but this is unknown to its neighbors  $\mathcal{N}_i$ . Our aim is to synthesize a control policy that can be implemented by each node in a distributed way with data available to it. The resulting controller should stabilize the system to the origin while minimizing  $J(x, u)$ .

## IV. DATA-BASED REPRESENTATION FOR OPTIMIZATION

Here, we introduce a data-based representation of system trajectories that is employed to pose a network optimization problem equivalent to (2). Throughout this section, we let  $x^d : [0, T_d - 1] \rightarrow \mathbb{R}^n$ ,  $u^d : [0, T_d - 1] \rightarrow \mathbb{R}^m$  be sequences such that  $w^d(t) = \text{col}(u^d(t), x^d(t))$  is a trajectory of (1). Let

$$w_i^d(t) = \begin{cases} \text{col}(u_i^d(t), x_{\mathcal{N}_i}^d(t), x_i^d(t)) & \text{if } i \in S \\ \text{col}(x_{\mathcal{N}_i}^d(t), x_i^d(t)) & \text{if } i \notin S, \end{cases}$$

for each  $i \in \mathcal{V}$  and  $0 \leq t < T_d - 1$ . Then  $w_i^d$  is the data available to each node. Let  $u : [0, \tau] \rightarrow \mathbb{R}^m$ ,  $x : [0, \tau] \rightarrow \mathbb{R}^n$  be arbitrary sequences where  $T_d \geq (n + m)\tau - 1$ . Define  $w(t) = \text{col}(x(t), u(t))$  and

$$w_i(t) = \begin{cases} \text{col}(u_i(t), x_{\mathcal{N}_i}(t), x_i(t)) & \text{if } i \in S \\ \text{col}(x_{\mathcal{N}_i}(t), x_i(t)) & \text{if } i \notin S. \end{cases}$$

Let  $k_i = n_i\tau + \sum_{j \in \mathcal{N}_i} n_j\tau + m_i\tau$  for  $i \in S$ , and  $k_i = n_i\tau + \sum_{j \in \mathcal{N}_i} n_j\tau$  otherwise. We define  $E_i \in \mathbb{R}^{k_i \times (m+n)\tau}$  to be the matrix consisting of all ones and zeros such that  $E_i w^d = w_i^d$  and  $E_i w = w_i$ .

#### A. Data-Based Representation of Network Trajectories

Lemma 1 states conditions under which the behavior of the system can be described completely by the Hankel matrix of the sampled data. Here we extend Lemma 1 to the setting of a network system to build a data-based representation of network trajectories using the Hankel matrices of the data available to each agent,  $\mathcal{H}_\tau(w_i^d)$ . We show that under certain conditions the image of  $\mathcal{H}_\tau(w_i^d)$  is the set of all possible trajectories of node  $i$ .

*Proposition 1 (Sufficiency of Data-Based Image Representation):* If for each  $i \in \mathcal{V}$  there exists  $g_i \in \mathbb{R}^{T_d - \tau + 1}$  with  $\mathcal{H}_\tau(w_i^d)g_i = w_i$ , then  $w$  is a trajectory of (1).

*Proof:* Writing  $g_i = (g_i(0), g_i(1), \dots, g_i(T_d - \tau))^T$  so for all  $0 \leq t < \tau - 1$ ,  $w_i(t) = \sum_{k=0}^{T_d - \tau + 1} g_i(k)w_i^d(t+k)$ , it follows that  $x_i(t+1) = \sum_{k=0}^{T_d - \tau} g_i(k)x_i^d(t+k+1)$  so for  $i \notin S$ ,

$$\begin{aligned} x_i(t+1) &= \sum_{k=0}^{T_d - \tau} g_i(k) \left( A_{ii}x_i^d(t+k) + \sum_{j \in \mathcal{N}_i} A_{ij}x_j^d(t+k) \right) \\ &= A_{ii}x_i(t) + \sum_{j \in \mathcal{N}_i} A_{ij}x_j(t). \end{aligned}$$

By a similar computation, we can show that for each  $i \in S$ ,

$$x_i(t+1) = A_{ii}x_i(t) + \sum_{j \in \mathcal{N}_i} A_{ij}x_j(t) + B_i u_i(t),$$

which is consistent with (1).  $\blacksquare$

Next we identify conditions for the converse of the above result to hold, i.e., when the Hankel matrices of all the agents characterize all possible network trajectories.

*Proposition 2 (Necessity of Data-Based Image Representation):* If  $(A, B)$  is controllable,  $w^d$  is a trajectory of (1) and either

- 1)  $u^d$  is persistently exciting of order  $n + \tau$ ;
- 2)  $\text{col}(u_i^d(t), x_{\mathcal{N}_i}^d(t))$  is persistently exciting of order  $n_i + \tau$  for each  $i \in S$ , and  $x_{\mathcal{N}_i}^d$  is persistently exciting of order  $n_i + \tau$  for each  $i \in \mathcal{V} \setminus S$ ;

then for all  $i \in \mathcal{V}$  there exists  $g_i \in \mathbb{R}^{T_d - \tau + 1}$  such that  $\mathcal{H}_\tau(w_i^d)g_i = w_i$ .

*Proof:* In the case of 1) we simply apply Lemma 1 to obtain  $g \in \mathbb{R}^{T_d - \tau + 1}$  where  $\mathcal{H}_\tau(w^d)g = w$ , and note that for all  $i \in \mathcal{V}$ ,  $w_i = E_i w = E_i \mathcal{H}_\tau(w^d)g = \mathcal{H}_\tau(w_i^d)g$ , so the result follows by letting  $g_i = g$ . For case 2), we think of  $x_j$  for  $j \in \mathcal{N}_i$  as an input to node  $i$ . Letting  $k = |\mathcal{N}_i|$ , where  $\mathcal{N}_i = \{j_1, j_2, \dots, j_k\}$ , and defining

$$\tilde{B}_i = \begin{cases} \begin{bmatrix} A_{ij_1} & A_{ij_2} & \cdots & A_{ij_k} & B_i \end{bmatrix} & i \in S \\ \begin{bmatrix} A_{ij_1} & A_{ij_2} & \cdots & A_{ij_k} \end{bmatrix} & i \notin S, \end{cases}$$

we have

$$x_i(t+1) = \begin{cases} A_{ii}x_i(t) + \tilde{B}_i \text{col}(x_{j_1}, x_{j_2}, \dots, x_{j_k}, u_i) & i \in S \\ A_{ii}x_i(t) + \tilde{B}_i \text{col}(x_{j_1}, x_{j_2}, \dots, x_{j_k}) & i \notin S \end{cases}$$

Let  $x_i^0 \in \mathbb{R}^{n_i}$  be arbitrary, and  $x^0 \in \mathbb{R}^n$  such that the  $i$ th block component is  $x_i^0$ . Since  $(A, B)$  is controllable there exists an input  $\tilde{u} : [0, n] \rightarrow \mathbb{R}^m$  such the corresponding state trajectory  $\tilde{x} : [0, n] \rightarrow \mathbb{R}^m$  with  $\tilde{x}(0) = x^0$  has  $\tilde{x}(n) = 0$ . Note that if  $i \in S$ , then  $\tilde{x}_i$  is the state trajectory corresponding

to the input  $\text{col}(\tilde{u}_i, \tilde{x}_{\mathcal{N}_i})$ , and  $\tilde{x}_i(0) = x_i^0$  and  $\tilde{x}_i(n) = 0$ . Likewise, if  $i \notin S$ ,  $\tilde{x}_i$  is the state trajectory corresponding to the input  $\tilde{x}_{\mathcal{N}_i}$ , and  $\tilde{x}_i(0) = x_i^0$  and  $\tilde{x}_i(n) = 0$ . Hence  $(A_{ii}, \tilde{B}_i)$  is controllable for all  $i \in \mathcal{V}$  and the result follows from Lemma 1.  $\blacksquare$

*Remark 1 (Feasibility of Identifiability Conditions):* Proposition 2 gives conditions on when the data is rich enough to characterize all possible trajectories of the system. Condition (i) gives conditions on the input sequence,  $u^d$ , which guarantee *a priori* the identifiability of the system from data. This condition is generically true in the sense that the set of sequences  $u^d$  which are not persistently exciting of order  $n + \tau$  (even though for all  $i \in S$ ,  $u_i^d$  is) have zero Lebesgue measure. In general, it is difficult to verify condition (i) in a distributed manner. On the other hand, it is straightforward to verify condition (ii) using only information available to the individual agents. However this verification must be done in an ad hoc manner, after the input has been applied to the system. While the condition is sufficient for identifiability, there are systems where for all inputs  $u^d$ , the resulting trajectory  $w^d$  will never satisfy it.

#### B. Equivalent Network Optimization Problem

Here, we build on the data-based image representation of network trajectories in a distributed fashion to pose a network optimization problem that can be solved with the data available to each agent, which is equivalent to an LQR problem with a time horizon of  $T$ . Each node can use this representation along with  $T_{\text{ini}} > 0$  past states and inputs to predict future trajectories assuming that the hypotheses of Proposition 2 are satisfied. Formally, let  $\tau = T_{\text{ini}} + T + 1$  and let  $u^{\text{ini}} : [0, T_{\text{ini}} - 1] \rightarrow \mathbb{R}^m$  and  $x^{\text{ini}} : [0, T_{\text{ini}} - 1] \rightarrow \mathbb{R}^n$  be sequences such that  $\text{col}(u^{\text{ini}}, x^{\text{ini}})$  is a  $T_{\text{ini}}$  long trajectory of the system. In the network optimization we introduce below, we optimize over system trajectories  $\text{col}(u, x)$  of length  $\tau$ , constrained so the first  $T_{\text{ini}}$  samples of  $u$  and  $x$  are  $u^{\text{ini}}$  and  $x^{\text{ini}}$  resp. This plays a similar role to the initial condition constraint in (2).

For each node  $i \in \mathcal{V}$ , define

$$H_i = \begin{cases} \begin{bmatrix} \mathcal{H}_\tau(u_i^d) \\ \mathcal{H}_\tau(x_{\mathcal{N}_i}^d) \\ \mathcal{H}_\tau(x_i^d) \end{bmatrix} & \text{if } i \in S \\ \begin{bmatrix} \mathcal{H}_\tau(x_{\mathcal{N}_i}^d) \\ \mathcal{H}_\tau(x_i^d) \end{bmatrix} & \text{if } i \notin S. \end{cases} \quad (3)$$

Consider the following problem

$$\begin{aligned} & \underset{g_i, u_i, x_i}{\text{minimize}} && \sum_{i \in S} J_i(x_i, u_i) + \sum_{i \in \mathcal{V} \setminus S} J_i(x_i) \\ & \text{subject to} && H_i g_i = \text{col}(u_i^{\text{ini}}, u_i, x_{\mathcal{N}_i}^{\text{ini}}, x_{\mathcal{N}_i}^{\text{ini}}, x_i^{\text{ini}}, x_i), \quad i \in S \\ & && H_i g_i = \text{col}(x_{\mathcal{N}_i}^{\text{ini}}, x_{\mathcal{N}_i}^{\text{ini}}, x_i^{\text{ini}}, x_i), \quad i \notin S \\ & && x_i(T) = 0 \quad i \in \mathcal{V}. \end{aligned} \quad (4)$$

Although (4) does not necessarily have a unique optimizer, any optimizer  $(g^*, u^*, x^*)$  of (4) is such that  $u^*$  and  $x^*$  are the optimal input and state sequences of (2), as formalized next.

*Proposition 3 (Equivalent Network Optimization):* Consider the system (1) and sample trajectory  $w^d$  satisfying the hypotheses of Proposition 2 and let  $x^0 = Ax^{\text{ini}}(T_{\text{ini}} - 1) + Bu^{\text{ini}}(T_{\text{ini}} - 1)$ . Then the following hold:

- 1) If problem (2) is feasible, then it has a unique optimizer;
- 2) Problem (4) is feasible if and only if (2) is feasible;

**Algorithm 1** Distributed Data-Based Predictive Control

- 
- 1: **Input:** Sample trajectory  $w^d$ , performance indices  $(Q_i)_{i=1}^N$ ,  $(R_i)_{i=1}^N$
  - 2: Initialize  $H_i$  as in equation (3), let  $u^{\text{ini}}$  and  $x^{\text{ini}}$  be the  $T_{\text{ini}}$  most recent states and inputs respectively, and set  $t = 0$ .
  - 3: **while**  $\|x\| > 0$  **do**
  - 4:   Use a distributed optimization algorithm to obtain an approximate solution to (4),  $\hat{z} = \text{col}(\hat{g}, \hat{u}, \hat{x})$ , such that  $\|\hat{u} - u^*\| \leq \delta \min\{1, \|x^{\text{ini}}(T_{\text{ini}} - 1)\|\}$ .
  - 5:   Apply the input  $\hat{u}(0)$ .
  - 6:   Set  $t$  to  $t + 1$  and  $u^{\text{ini}}$  and  $x^{\text{ini}}$  to the  $T_{\text{ini}}$  most recent inputs and states respectively.
- 

- 3) If (4) is feasible,  $(u^{1,*}, x^{1,*})$  is the optimizer of (2) and  $(g^*, u^{2,*}, x^{2,*})$  is an optimizer of (4), then  $u^{1,*} = u^{2,*}$  and  $x^{1,*} = x^{2,*}$ .

We omit the proof for space reasons, but note that it is the analogue of [19, Th. 5.1 and Corollary 5.2] for the case of network systems once one invokes Propositions 1 and 2 above. Unlike the original network optimization problem (2), for which agents lack knowledge of the system matrices  $A$ ,  $B$ , the network optimization problem (4) can be solved in a distributed way with the information available to them. The structure of the problem (aggregate objective functions plus locally expressible constraints) makes it amenable to a variety of distributed optimization algorithms, see [21], [22]. In Section V-B below, we employ a primal-dual dynamic to find asymptotically a solution of (4) in a distributed way.

*Remark 2 (Scalability of Network Optimization):* As the number of nodes in the network increases so does the state dimension, hence more data is required in order to maintain persistency of excitation. A necessary condition is  $T_d \geq (n+m+1)(T_{\text{ini}}+T) - 1$ . Assuming that  $T \sim O(n)$ ,  $T_{\text{ini}} \sim O(1)$ , we have  $T_d \sim O(n^2 + mn)$ . The decision variable for each node is  $z_i = \text{col}(g_i, u_i, x_i)$  when  $i \in S$  and  $z_i = \text{col}(g_i, x_i)$  otherwise. The size of  $z_i$  is  $O(n^2 + mn)$ . However, using the distributed optimization algorithm of Section V-B, agent  $i$  only needs to send messages of size  $O(k_i)$  to agent  $j$ .

## V. DISTRIBUTED DATA-BASED PREDICTIVE CONTROL

Here we introduce a distributed data-based predictive control scheme to stabilize the system (1) to the origin, as described in Section III. To do this, we solve the network optimization problem (4) in a receding horizon manner with  $u^{\text{ini}}$  and  $x^{\text{ini}}$  updated every time step based on the systems current state. The control scheme is summarized in Algorithm 1.

The rest of the section proceeds by first showing that the controller resulting from Algorithm 1 is stabilizing even when the network optimization (4) is solved only approximately; and then introducing a particular distributed solver for (4) along with a suboptimality certificate to check, in a distributed manner, the stopping condition in Step 4 of Algorithm 1.

### A. Stability Analysis of Closed-Loop System

In the rest of this letter, we rely on the following assumption.

*Assumption 1:* Consider the system (1),

- 1) the collected data satisfies the hypotheses of Proposition 2;

- 2) the optimization problem (4) is feasible for all  $(u^{\text{ini}}, x^{\text{ini}})$  which is a valid  $T_{\text{ini}}$ -sample long system trajectory.

Since the system is controllable (see 1)), a sufficient condition for guaranteeing the feasibility in 2) is  $T \geq n$ . In fact, in such case, there exists a trajectory  $(u, x)$  such that  $x(T) = 0$ . It follows that  $(u, x)$  is feasible for (2), so by Proposition 3, there exists some  $g$  such that  $(g, u, x)$  is feasible for (4).

Under Assumption 1, the closed-loop system with the controller corresponding to a receding horizon implementation of (2) is globally exponentially stable, see [23, Th. 12.2]. Unlike [19], we do not assume we have access to the exact solution of (4) since distributed optimization algorithms typically only converge asymptotically to the true optimizer and must be terminated in finite time. Here we show that Algorithm 1 still stabilizes the system when the tolerance  $\delta$  is sufficiently small.

*Theorem 1 (Distributed, Data-Based Predictive Control is Stabilizing):* For  $\delta > 0$ , let  $\phi_\delta : \mathbb{R}^n \rightarrow \mathbb{R}^m$  be the feedback control corresponding to Algorithm 1. Under Assumption 1, there exists  $\delta^* > 0$  such that for all  $\delta < \delta^*$ , the origin is globally asymptotically stable with respect to the closed-loop dynamics  $x(t+1) = Ax(t) + B\phi_\delta(x(t))$ .

*Proof:* Let  $\phi_{\text{mpc}} : \mathbb{R}^n \rightarrow \mathbb{R}^m$  be the feedback corresponding to a receding horizon implementation of (2). Consider the system  $x(t+1) = f(x(t), v(t))$ , where  $f(x, v) = A + B\phi_{\text{mpc}}(x) + Bv$ . Let  $J^*(x^0) = J(x^*, u^*)$ , where  $(u^*, x^*)$  is an optimizer of (2). Because (4) is nondegenerate,  $J^*$  is continuously differentiable, see [23, Th. 6.9], and the system is input-to-state stable (ISS) with  $J^*$  being a ISS-Lyapunov function satisfying  $J^*(f(x, v)) - J^*(x) \leq -\alpha\|x\|^2 + \sigma\|v\|^2$  for constants  $\alpha, \sigma > 0$ , see [24, Th. 1] (albeit the result is stated there for systems where  $A$  is Schur stable, the same reasoning is valid when there are no state or input constraints and  $A$  is unstable). Because the system  $x(t+1) = f(x(t), 0)$  without disturbances is exponentially stable [25], it follows by [26, proof of Lemma 3.5] that the gain function is linear, so there exist  $\gamma > 0$  and a class  $\mathcal{KL}$  function  $\beta$  such that

$$\|x(t)\| \leq \beta(\|x(0)\|, t) + \gamma \sup_{0 \leq \tau \leq t} \|v(\tau)\|,$$

for all  $t \in \mathbb{Z}_{\geq 0}$ . Let  $x : \mathbb{Z}_{\geq 0} \rightarrow \mathbb{R}^n$  be a trajectory of the closed-loop dynamics of (1) with the controller described by Algorithm 1 where  $\delta < \delta^* = \gamma^{-1}$  and define  $v(t) = \phi_\delta(x(t)) - \phi_{\text{mpc}}(x(t))$ . It follows that  $x(t+1) = f(x(t), v(t))$ . Note that  $\phi_{\text{mpc}}(x(t)) = u^*(0; u^{\text{ini}}, x^{\text{ini}})$ , where  $u^{\text{ini}} = \phi_\delta(x(t-1))$  and  $x^{\text{ini}} = x(t-1)$  so it follows that  $\|v(t)\| = \|\hat{u}(0; u^{\text{ini}}, x^{\text{ini}}) - u^*(0; u^{\text{ini}}, x^{\text{ini}})\| \leq \delta\|x(t-1)\|$ . We claim that for all  $k \in \mathbb{N}$ , there exists  $T_k \in \mathbb{N}$  such that  $\|x(t)\| \leq (k+1)(\gamma\delta)^k$  whenever  $t \geq T_k$ . The case when  $k = 1$  follows by observing that  $\|x(t)\| \leq \beta(\|x(0)\|, t) + \gamma\delta$  and there exists  $T_1$  such that  $\beta(\|x(0)\|, t) < \gamma\delta$  for all  $t \geq T_1$ . If the claim holds for some  $k$ , then for all  $t \geq T_k + 1$ ,

$$\begin{aligned} \|x(t)\| &\leq \beta((k+1)(\gamma\delta)^k, t) + \gamma \sup_{T_k+1 < \tau \leq t} \|v(\tau)\| \\ &\leq \beta((k+1)(\gamma\delta)^k, t) + (k+1)(\gamma\delta)^{k+1}, \end{aligned}$$

so by choosing  $T_{k+1}$  such that  $\beta((k+1)(\gamma\delta)^k, t) < (\gamma\delta)^{k+1}$  for all  $t \geq T_{k+1}$ , then  $\|x(t)\| < (k+2)(\gamma\delta)^{k+1}$  for all  $t > T_{k+1}$  and the claim follows by induction. Hence  $\limsup_{t \rightarrow \infty} \|x(t)\| \leq \limsup_{k \rightarrow \infty} (k+1)(\gamma\delta)^k = 0$ . To show global Lyapunov stability, let  $\eta > 0$  be arbitrary, and suppose that  $\|x(0)\|$  is chosen

so that  $\beta(\|x(0)\|, 0) < (1 - \gamma\delta)\eta$  and  $\|x(0)\| < \eta$ . Then for all  $t > 0$ ,

$$\begin{aligned} \|x(t)\| &\leq (1 - \gamma\delta)\eta + \gamma \sup_{0 \leq \tau \leq t} \|v(\tau)\| \\ &\leq (1 - \gamma\delta)\eta + \gamma\delta\|x(t-1)\|. \end{aligned}$$

If  $\|x(t-1)\| < \eta$ , then  $\|x(t)\| < \eta$ . It follows by induction on  $t$  that  $\|x(t)\| < \eta$  for all  $\eta$ . ■

### B. Primal-Dual Solver for Network Optimization

In this section we introduce a method for solving the optimization problem (4) in a distributed way. We let

$$z_i = \begin{cases} \text{col}(g_i, u_i, x_i) & i \in S, \\ \text{col}(g_i, x_i) & i \notin S, \end{cases}$$

and  $z = \text{col}(z_1, \dots, z_N)$ . Note  $z_i \in \mathbb{R}^{d_i}$ , where  $d_i = T - (T_{\text{ini}} + T) + 1 + n_i + m_i$  for  $i \in S$  and  $d_i = T - (T_{\text{ini}} + T) + 1 + n_i$  otherwise. Problem (4) can be written as

$$\begin{aligned} &\text{minimize} \quad \sum_{z_i \in \mathbb{R}^{d_i}} \|z_i\|_{\mathcal{Q}_i}^2 \\ &\text{subject to} \quad \mathcal{A}_i z_{\mathcal{N}_i} = b_i. \end{aligned} \quad (5)$$

for suitable  $\mathcal{Q}_i \in \mathbb{R}^{d_i \times d_i}$ ,  $\mathcal{Q}_i \succeq 0$ ,  $\mathcal{A}_i \in \mathbb{R}^{c_i \times d_i}$ , and  $b_i \in \mathbb{R}^{c_i}$ , with  $i, c_i \in \mathbb{Z}$ . The Lagrangian of (5) is

$$\mathcal{L}(z, \lambda) = \sum_{i \in \mathcal{V}} \|z_i\|_{\mathcal{Q}_i}^2 + \lambda_i^\top (\mathcal{A}_i z_{\mathcal{N}_i} - b_i).$$

If  $\lambda^*$  is an optimizer of the dual problem, then the pair  $(z^*, \lambda^*)$  is a (min-max) saddle point of  $\mathcal{L}$ , meaning that  $\mathcal{L}(z^*, \lambda) \leq \mathcal{L}(z^*, \lambda^*) \leq \mathcal{L}(z, \lambda^*)$  for all  $z \in \mathbb{R}^d$  and  $\lambda \in \mathbb{R}^c$ . The saddle-point property of the Lagrangian suggests that the primal-dual flow, which descends along the gradient of the primal variable and ascends along the gradient of the dual variable,

$$\begin{aligned} \begin{bmatrix} \dot{z}_i \\ \dot{\lambda}_i \end{bmatrix} &= \begin{bmatrix} -\nabla_{z_i} \mathcal{L}(z, \lambda, \mu) \\ \nabla_{\lambda_i} \mathcal{L}(z, \lambda, \mu) \end{bmatrix} \\ &= \begin{bmatrix} -2\mathcal{Q}_i z_i - F_{ii} \mathcal{A}_i^\top \lambda_i - \sum_{j \in \mathcal{N}_i} F_{ij} \mathcal{A}_j^\top \lambda_j \\ \mathcal{A}_i z_{\mathcal{N}_i} + b_i \end{bmatrix}, \end{aligned} \quad (6)$$

can be used to compute the optimizer. Here,  $F_{ij} \in \mathbb{R}^{(d_i + \sum_{j \in \mathcal{N}_i} d_j) \times d_i}$  is the matrix such that  $F_{ij} z_{\mathcal{N}_j} = z_i$ . By [22, Corollary 4.5], the flow converges asymptotically to a saddle point of  $\mathcal{L}$ . This procedure is fully distributed, since the flow equations in (6) can be computed with the information available to each agent or its direct neighbors. In particular, if  $j \in \mathcal{N}_i$ , then the message agent  $j$  shares with agent  $i$  consists of  $\text{col}(x_j, \lambda_j) \in \mathbb{R}^{n_j T + k_j}$ , which is  $O(k_i)$  (see Remark 2).

We conclude by providing a certificate that can be used to verify the stopping condition of Step 4 in Algorithm 1.

*Proposition 4 (Suboptimality Certificate):* Let  $u^*$  and  $x^*$  denote the optimal input and state trajectories of (2),  $y = \text{col}(z, \lambda)$ ,  $\mathcal{Q} = \text{diag}(\mathcal{Q}_1, \mathcal{Q}_2, \dots, \mathcal{Q}_N)$ ,  $\mathcal{A} = [F_1^\top \mathcal{A}_1^\top, F_2^\top \mathcal{A}_2^\top, \dots, F_N^\top \mathcal{A}_N^\top]^\top$ ,  $b = \text{col}(b_1, b_2, \dots, b_N)$ , and

$$M = \begin{bmatrix} -2\mathcal{Q}^\top & -\mathcal{A}^\top \\ \mathcal{A} & 0 \end{bmatrix} \quad q = \begin{bmatrix} 0 \\ b \end{bmatrix}.$$

Under Assumption 1, and with the flow given by (6), if  $\|\text{col}(\dot{z}_i, \dot{\lambda}_i)\| < \rho$  for all  $i \in \mathcal{V}$ , where  $\rho = \frac{\delta^2}{N\|M^\dagger\|^2}$ , then  $\|u - u^*\| < \delta$ .

*Proof:* The set of saddle points of  $\mathcal{L}$  is  $\mathcal{S} = \{y \mid My + q = 0\}$ . Since the optimal input and state trajectories are unique, all

saddle points share the property that, for each  $i \in S$ , the  $(u_i, x_i)$  components of their  $z_i$  equal  $(u_i^*, x_i^*)$ . Given an arbitrary  $y$ , we have  $\|u - u^*\| \leq \|y - w\|$  for all  $w \in \mathcal{S}$ , and hence  $\|u - u^*\| \leq \inf_{w \in \mathcal{S}} \|y - w\|$ . The set of saddle points can also be described as  $\mathcal{S} = \{\hat{y}\} + \ker M$ , for any  $\hat{y} \in \mathcal{S}$ . Therefore,  $\inf_{w \in \mathcal{S}} \|y - w\| = \inf_{v \in \ker M} \|y - \hat{y} - v\|$ . Since  $I - M^\dagger M$  is the orthogonal projection onto  $\ker M$ ,

$$\begin{aligned} \inf_{v \in \ker M} \|y - \hat{y} - v\| &= \|(y - \hat{y}) - (I - M^\dagger M)(y - \hat{y})\| \\ &= \|M^\dagger(My - M\hat{y})\| = \|M^\dagger(My + q)\| \leq \|M^\dagger\| \|\hat{y}\|, \end{aligned}$$

and therefore,

$$\|u - u^*\|^2 \leq \|M^\dagger\|^2 \sum_{i \in \mathcal{V}} \|\dot{y}_i\|^2 < \|M^\dagger\|^2 \sum_{i \in \mathcal{V}} \frac{\delta^2}{N\|M^\dagger\|^2} = \delta^2. \quad \blacksquare$$

The suboptimality certificate can be checked in a fully distributed manner using information locally available to each agent provided that  $\|M^\dagger\|$  is known. Because  $M$  depends only on the objective  $\mathcal{Q}$  and constraints  $\mathcal{A}$ , which in turn comes from the sample trajectory  $w^d$ , it can be computed offline. Finally, it is possible for each agent to compute a bound on  $\|M^\dagger\|$  using the fact that, for  $M = [M_1^\top, M_2^\top, \dots, M_N^\top]^\top$ , one has  $\|M^\dagger\| \leq \|M_i^\dagger\|$  for all  $i \in \mathcal{V}$ . It follows that for all  $i \in \mathcal{V}$ ,

$$\|M^\dagger\| \leq \left\| \begin{bmatrix} 2\mathcal{Q}_i & F_{ii} \mathcal{A}_i^\top & F_{ij_1} \mathcal{A}_{j_1}^\top & \dots & F_{ij_{|\mathcal{N}_i|}} \mathcal{A}_{j_{|\mathcal{N}_i|}}^\top \end{bmatrix}^\dagger \right\|,$$

for  $\{j_1, j_2, \dots, j_{|\mathcal{N}_i|}\} = \mathcal{N}_i$ , so each agent can compute a bound on  $\|M^\dagger\|$  using data available to itself and its neighbors.

### C. Numerical Simulations

We simulate the proposed distributed data-based predictive controller on a Newman-Watts-Strogatz network [27] and a star network. In each case,  $A$  and  $B$  are chosen at random so that  $(A, B)$  is controllable. The input sequence is chosen as  $u^d(t) = Kx^d(t) + w(t)$ , where  $K$  is a matrix so that  $A + BK$  is marginally stable (the data does not need to be generated from a stable system, but this is done to avoid numerical issues), and  $w(t)$  is a Gaussian white noise process. We use Proposition 2 to ensure that the data is informative enough for data-driven control. In both cases, condition 1 is satisfied. Condition 2 fails for the Newman-Watts-Strogatz network, but is satisfied by the star network (see Remark 1). We integrate the primal-dual flow using the stopping condition in Proposition 4 is used to terminate the flow. Fig. 1 shows the closed-loop state trajectories and the number of iterations on each time step with different values of  $\rho$ . For the Newman-Watts-Strogatz network, see Fig. 1(a), the time horizon is  $T = n$ . For the star network, see Fig. 1(b), the time horizon is  $T = 5 < n$ , but the optimization at each time step is still feasible. In both cases, the distributed data-based predictive controller better approximates the exact MPC for smaller values of  $\rho$  at the cost of more iterations per time step.

## VI. CONCLUSION AND FUTURE WORK

We have introduced a distributed data-based predictive controller for stabilizing network linear dynamics described by unknown system matrices. Instead of building a dynamic model, agents learn a non-parametric representation based on a single trajectory and use it to implement a controller as the

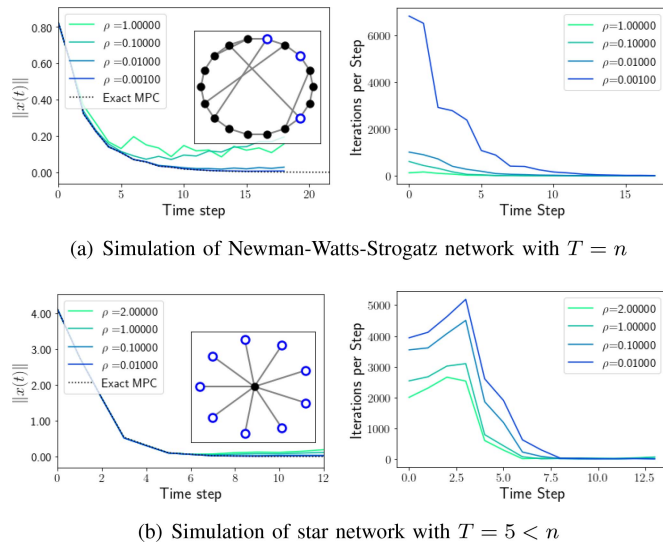


Fig. 1. State trajectories of Data-based Receding Horizon Controller on various network topologies.

solution of a network optimization problem solved in a receding horizon manner and in a distributed way. Future work will explicitly quantify the tolerance  $\delta^*$  in terms of the available data and study ways to construct a terminal cost without knowledge of the underlying model to guarantee stability when the stabilizing terminal constraint is omitted. We plan to extend the results to cases where there are constraints on the state and input, characterize the robustness properties of the introduced control scheme, investigate ways of improving its scalability, and consider more general scenarios, including the presence of noise in the data, inputs not persistently exciting of sufficiently high order, and partial observations of the network state. We also plan to explore improvements to the primal-dual flow to solve the optimization problem with fewer iterations and less communication between the agents.

## REFERENCES

- [1] F. Bullo, J. Cortés, and S. Martinez, *Distributed Control of Robotic Networks* (Applied Mathematics Series). Princeton, NJ, USA: Princeton Univ. Press, 2009.
- [2] M. Mesbahi and M. Egerstedt, *Graph Theoretic Methods in Multiagent Networks* (Applied Mathematics Series). Princeton, NJ, USA: Princeton Univ. Press, 2010.
- [3] R. R. Negenborn and J. M. Maestre, "Distributed model predictive control: An overview and roadmap of future research opportunities," *IEEE Control Syst.*, vol. 34, no. 4, pp. 87–97, Aug. 2014.
- [4] M. Rotkowitz and S. Lall, "A characterization of convex problems in decentralized control," *IEEE Trans. Autom. Control*, vol. 51, no. 2, pp. 274–286, Feb. 2006.
- [5] F. Lin, M. Fardad, and M. R. Jovanović, "Design of optimal sparse feedback gains via the alternating direction method of multipliers," *IEEE Trans. Autom. Control*, vol. 58, no. 9, pp. 2426–2431, Sep. 2013.
- [6] G. Fazelnia, R. Madani, A. Kalbat, and J. Lavaei, "Convex relaxation for optimal distributed control problems," *IEEE Trans. Autom. Control*, vol. 62, no. 1, pp. 206–221, Jan. 2017.
- [7] L. Frieri, Y. Zheng, A. Papachristodoulou, and M. Kamgarpour, "On separable quadratic Lyapunov functions for convex design of distributed controllers," in *Proc. Eur. Control Conf.*, Naples, Italy, 2019, pp. 42–49.
- [8] J. Kober, J. A. Bagnell, and J. Peters, "Reinforcement learning in robotics: A survey," *Int. J. Robot. Res.*, vol. 32, no. 11, pp. 1238–1274, 2013.
- [9] L. Bu, R. Babu, and B. D. Schutter, "A comprehensive survey of multiagent reinforcement learning," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 38, no. 2, pp. 156–172, Mar. 2008.
- [10] B. Recht, "A tour of reinforcement learning: The view from continuous control," *Annu. Rev. Control Robot. Auton. Syst.*, vol. 2, pp. 253–279, May 2019.
- [11] D. Jia and B. Krogh, "Min–max feedback model predictive control for distributed control with communication," in *Proc. Amer. Control Conf.*, Anchorage, AK, USA, 2002, pp. 4507–4512.
- [12] W. B. Dunbar, "Distributed receding horizon control of dynamically coupled nonlinear systems," *IEEE Trans. Autom. Control*, vol. 52, no. 7, pp. 1249–1263, Jul. 2007.
- [13] B. T. Stewart, A. N. Venkat, J. B. Rawlings, S. J. Wright, and G. Pannocchia, "Cooperative distributed model predictive control," *Syst. Control Lett.*, vol. 59, no. 8, pp. 460–469, 2010.
- [14] L. Ljung, *System Identification: Theory for the User* (Prentice Hall Information and System Sciences Series). Upper Saddle River, NJ, USA: Prentice-Hall, 1999.
- [15] J. C. Willems, P. Rapisarda, I. Markovskiy, and B. L. M. D. Moor, "A note on persistency of excitation," *Syst. Control Lett.*, vol. 54, no. 4, pp. 325–329, 2005.
- [16] U. Park and M. Ikeda, "Stability analysis and control design of LTI discrete-time systems by the direct use of time series data," *Automatica*, vol. 45, no. 5, pp. 1265–1271, 2009.
- [17] T. M. Maupong and P. Rapisarda, "Data-driven control: A behavioral approach," *Syst. Control Lett.*, vol. 101, pp. 37–43, Mar. 2017.
- [18] C. D. Persis and P. Tesi, "Formulas for data-driven control: Stabilization, optimality and robustness," 2019. [Online]. Available: arXiv:1903.06842.
- [19] J. Coulson, J. Lygeros, and F. Dörfler, "Data-enabled predictive control: In the shallows of the DeePC," in *Proc. Eur. Control Conf.*, Naples, Italy, 2019, pp. 307–312.
- [20] J. Berberich, J. Köhler, A. M. Müller, and F. Allgöwer, "Data-driven model predictive control with stability and robustness guarantees," 2019. [Online]. Available: arXiv:1906.04679.
- [21] 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, 2011.
- [22] A. Cherukuri, B. Ghahsifard, and J. Cortés, "Saddle-point dynamics: Conditions for asymptotic stability of saddle points," *SIAM J. Control Optim.*, vol. 55, no. 1, pp. 486–511, 2017.
- [23] F. Borrelli, A. Bemporad, and M. Morari, *Predictive Control for Linear and Hybrid Systems*. Cambridge, U.K.: Cambridge Univ. Press, 2017.
- [24] A. Jadbabaie and A. S. Morse, "On the ISS property for receding horizon control of constrained linear systems," *IFAC Proc. Vol.*, vol. 35, no. 1, pp. 37–40, 2002.
- [25] D. Q. Mayne, J. B. Rawlings, C. V. Rao, and P. O. M. Scokaert, "Constrained model predictive control: Stability and optimality," *Automatica*, vol. 36, pp. 789–814, Jun. 2000.
- [26] Z.-P. Jiang and Y. Wang, "Input-to-state stability for discrete-time nonlinear systems," *Automatica*, vol. 37, no. 6, pp. 857–869, 2001.
- [27] M. E. J. Newman and D. J. Watts, "Renormalization group analysis of the small-world network model," *Phys. Lett. A*, vol. 263, nos. 4–6, pp. 341–346, 1999.