Data-Based Receding Horizon Control of Linear Network Systems

Ahmed Allibhoy Jorge Cortés

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.

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 e.g., [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 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], [14] 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 [15] 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 [16], which characterizes system trajectories from a single sample trajectory, has recently gained attention in the area of data-based control [17]–[19], and has been used for predictive control in the recently developed DeePC framework [20, 21]. Our work 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[4]. A group of agents whose state evolves according to unknown coupled linear dynamics have each 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. Building upon the fundamental lemma, we introduce a distributed, data-based representation of possible network trajectories. We use this representation to pose the control synthesis as a network optimization problem 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 \( t, T \in \mathbb{Z}_{\geq 0} \) with \( t < T \), a signal \( w : [0, T - 1] \rightarrow \mathbb{R}^k \) is persistently exciting of order \( T \) if \( \text{rrow} \mathcal{H}_T(w) = kt \). Informally, this means that any arbitrary signal \( w : [0, T - 1] \rightarrow \mathbb{R}^k \) can be described as a linear combination of windows of width \( t \) in the signal \( w \). A necessary condition for persistence of excitation is \( T \geq (k + 1)t - 1 \).

Lemma II.1. (Fundamental Lemma [16]): Consider the LTI system \( x(t + 1) = Ax(t) + Bu(t) \), with \( (A, B) \) controllable. Let \( u^d : [0, T - 1] \rightarrow \mathbb{R}^m \) be sequences such that \( u^d = \text{col}(u^d, x^d) \) is a trajectory of the system and \( u^d \) is persistently exciting of order \( T \). Then for any pair \( u : [0, T - 1] \rightarrow \mathbb{R}^m \), \( x : [0, T - 1] \rightarrow \mathbb{R}^n \), \( w = \text{col}(u, x) \) is a trajectory of the system if and only if there exists \( g \in \mathbb{R}^{T + 1} \) such that \( \mathcal{H}_T(u^d)g = w \).

Lemma II.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 parameterized 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 \( G = (\mathcal{V}, \mathcal{E}) \) 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 \( u_{si} \in \mathbb{R}^{m_{si}} \), where \( s_i \) is the index of the input corresponding to node \( i \). The system dynamics are then

\[
\begin{cases}
    x_i(t+1) = A_i x_i(t) + \sum_{j \in \mathcal{N}_i} A_{ij} x_j(t) + B_{si} u_{si}, & i \in S, \\
    x_i(t+1) = \sum_{j \in \mathcal{N}_i} A_{ij} x_j(t), & i \notin S,
\end{cases}
\]

where \( x_i \in \mathbb{R}^{n_i} \), \( A_{ij} \in \mathbb{R}^{n_i \times n_j} \) and \( B_{si} \in \mathbb{R}^{n_i \times m_{si}} \). Let \( n = \sum_{i=1}^{N} n_i \) and \( m = \sum_{i=1}^{M} m_i \) and define \( x = \text{col}(x_1, x_2, \ldots, x_N) \in \mathbb{R}^n \) and \( u = \text{col}(u_1, u_2, \ldots, u_M) \in \mathbb{R}^m \). Let \( A \in \mathbb{R}^{n \times n} \) and \( B \in \mathbb{R}^{n \times m} \) be matrices so that \( \text{col}(u, x) \) takes the compact form \( x(t+1) = Ax(t) + Bu(t) \).

To each node \( i \in \mathcal{V} \), we associate an objective function \( J_i(x_i, u_{si}) = \sum_{t=0}^{T_{in}-1} \|x_i(t)\|_{Q_i}^2 + \sum_{t=0}^{T_{in}-1} \|u_{si}(t)\|_{R_i} \) when \( i \in S \) and \( J_i(x_i) = \sum_{t=0}^{T_{in}-1} \|x_i(t)\|_{Q_i}^2 + \sum_{t=0}^{T_{in}-1} \|x_i(t)\|_{Q_i} \) otherwise. Here, \( Q_i, Q_i \in \mathbb{R}^{n_i \times n_i} \) are positive semidefinite, \( R_i \in \mathbb{R}^{m \times m_i} \) is positive definite, \( \text{col}(u, x) \) is a system trajectory, and \( T_{in} \) is the time horizon of trajectories being considered.

Each node wants to drive its state \( x_i \) to the origin while minimizing \( J_i \). The resulting network objective function is then the sum of the objective functions across the nodes. Letting \( Q = \text{blkdiag}(Q_1, Q_2, \ldots, Q_N) \in \mathbb{R}^{n \times n} \), \( Q_f = \text{blkdiag}(Q_{f_1}, Q_{f_2}, \ldots, Q_{f_N}) \in \mathbb{R}^{n \times n} \), and \( R = \text{blkdiag}(R_{s_1}, R_{s_2}, \ldots, R_{s_N}) \in \mathbb{R}^{m \times m} \), this objective can be written as

\[
J(x, u) = \sum_{i \in S} J_i(x_i, u_{si}) + \sum_{i \notin S} J_i(x_i) = \|x(T_{in})\|_{Q_f} + \sum_{t=0}^{T_{in}-1} \|x(t)\|_Q + \|u(t)\|_R.
\]

If the system starts from the initial condition \( x(0) = x^0 \in \mathbb{R}^n \), the agents’ goal can be formulated as the network optimization problem:

\[
\begin{align*}
\text{minimize}_{u, x} & \quad \|x(T_{in})\|_Q + \sum_{t=0}^{T_{in}-1} \|x(t)\|_Q + \|u(t)\|_R \\
\text{subject to} & \quad x(t+1) = Ax(t) + Bu(t), \quad t \in [0, T_{in}], \\
& \quad x(0) = x^0.
\end{align*}
\]

Note that the agents’ decisions on their control inputs are coupled through the constraints.

A key aspect of this paper 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 \( \{u_{si}^d : [0, T - 1] \rightarrow \mathbb{R}^{m_{si}}\}_{i \in S} \), the corresponding state trajectories \( \{x_{si}^d : [0, T - 1] \rightarrow \mathbb{R}^{n_i}\}_{i \in V} \) 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 \( u_{si}^d \), but this is unknown to its neighbors \( N_i \). 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 \( x^d : [0, T - 1] \rightarrow \mathbb{R}^n \), \( u^d : [0, T - 1] \rightarrow \mathbb{R}^m \) be sequences such that \( u^d(t) = \text{col}(u^d(t), x^d(t)) \) is a trajectory of (1). Let

\[
w_i^d(t) = \begin{cases}
\text{col}(u_{si}^d(t), x_{N_i}^d(t), x_{si}^d(t)) & \text{if } i \in S, \\
\text{col}(x_{N_i}^d(t), x_{si}^d(t)) & \text{if } i \notin S,
\end{cases}
\]

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

\[
w_i(t) = \begin{cases}
\text{col}(u_{si}(t), x_{N_i}(t), x_{si}(t)) & \text{if } i \in S, \\
\text{col}(x_{N_i}(t), x_{si}(t)) & \text{if } i \notin S,
\end{cases}
\]

for each \( i \in \mathcal{V} \) and \( 0 \leq t < T - 1 \). Then \( w_i \) is the data available to each node. Let \( k_i = n_i \tau + \sum_{j \in N_i} m_j \tau + m_i \tau \) for \( i \in S \), and \( k_i = n_i \tau + \sum_{j \in N_i} m_j \tau \) 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 \(E_iw^d = w^d_i\) and \(E_iw = w_i\).

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

Lemma [II.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 [II.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, \(H_r(w^d_i)\). We show that under certain conditions the image of \(H_r(w^d_i)\) is the set of all possible trajectories of node \(i\).

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

**Proof.** Writing \(g_i = (g_i(0), g_i(1), \ldots, g_i(T-\tau))\) so for all \(0 \leq t \leq \tau - 1\), \(w_i(t) = \sum_{k=0}^{T-\tau-1} g_i(k)w^d_i(t+k)\), it follows that \(x_i(t+1) = \sum_{k=0}^{T-\tau} g_i(k)(A_i x^d_i(t+k) + \sum_{j \in N_i} A_{ij} x^d_j(t+k)) \) for \(i \notin S\),

\[
x_i(t+1) = \sum_{k=0}^{T-\tau} g_i(k)(A_i x^d_i(t+k) + \sum_{j \in N_i} A_{ij} x^d_j(t+k))
= A_i x_i(t) + \sum_{j \in N_i} A_{ij} x_j(t).
\]

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

\[
x_i(t+1) = A_i x_i(t) + \sum_{j \in N_i} A_{ij} x_j(t) + B_i u_i(t),
\]

which is consistent with [I].

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 IV.2. (Necessity of Data-Based Image Representation):** If \((A, B)\) is controllable, \(w^d\) is a trajectory of \(\mathcal{H}\) and either

(i) \(w^d\) is persistently exciting of order \(n + \tau\); 
(ii) \(\text{col}(x^d_{Ni}(t), x^d_{N_i}(t))\) is persistently exciting of order \(n_i + \tau\) for each \(i \in S\), and \(x^d_{Ni}\) 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 \(g_i \in \mathbb{R}^{T-\tau+1}\) such that \(H_r(w^d_i)g_i = w_i\).

**Proof.** In the case of \(\mathcal{H}\) we simply apply Lemma [II.1] to obtain \(g \in \mathbb{R}^{T-\tau+1}\) where \(H_r(w^d)g = w\), and note that for all \(i \in \mathcal{V}\), \(w_i = E_iw = E_iH_r(w^d)g = H_r(w^d_i)g\), so the result follows by letting \(g_i = g\).

For case \(\mathcal{H}\) we think of \(x_j\) for \(j \in N_i\) as an input to node \(i\). Letting \(k = |N_i|\), where \(N_i = \{1, 2, \ldots, k\}\), and defining

\[
\tilde{B}_i = \begin{bmatrix} A_{i1j} & A_{i2j} & \cdots & A_{ijk} & B_{ij} \end{bmatrix} \quad i \in S, 
\begin{bmatrix} A_{i1j} & A_{i2j} & \cdots & A_{ijk} \end{bmatrix} \quad i \notin S
\]

we have

\[
x_i(t+1) = \begin{cases} A_i x_i(t) + \tilde{B}_i \text{col}(x_{i1}, x_{i2}, \ldots, x_{ijk}, u_i) & i \in S, \\ A_i x_i(t) + \tilde{B}_i \text{col}(x_{i1}, x_{i2}, \ldots, x_{ijk}) & i \notin S. \end{cases}
\]

Let \(x^d_i \in \mathbb{R}^n\) be arbitrary, and \(x^0 \in \mathbb{R}^n\) such that the \(i\)th block component is \(x^0_i\). Since \((A, B)\) is controllable there exists an input \(\tilde{u} : [0, n] \to \mathbb{R}^m\) such the corresponding state trajectory \(\tilde{x} : [0, n] \to \mathbb{R}^m\) with \(\tilde{x}(0) = x^0\) has \(\tilde{x}(n) = 0\). Note that if \(i \in S\), then \(x_i\) is the state trajectory corresponding to the input \(\text{col}(u_{i1}, x_{Ni})\), and \(x_i(0) = x^0_i\) and \(x_i(n) = 0\). Likewise, if \(i \notin S\), \(x_i\) is the state trajectory corresponding to the input \(x_{Ni}\), and \(x_i(0) = x^0_i\) and \(x_i(n) = 0\). Hence \((A_i, B_i)\) is controllable for all \(i \in \mathcal{V}\) and the result follows from Lemma [II.1].

**Remark IV.3. (Feasibility of Identifiability Conditions):** Proposition [IV.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 \(i \in S\), \(u^d_i\) have zero Lebesgue measure. In general, it is difficult to verify condition (ii) 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 the LQR problem with a time horizon of \(T_{\text{ini}}\). 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 [IV.2] are satisfied. Formally, let \(\tau = T_{\text{ini}} + T_{\text{egr}} + 1\) and let \(u_{\text{ini}} : [0, T_{\text{ini}} - 1] \to \mathbb{R}^m\) and \(x_{\text{ini}} : [0, T_{\text{ini}} - 1] \to \mathbb{R}^n\) 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 that 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

\[
H_i = \begin{bmatrix} H_r(w^d_i) & H_r(x^d_{Ni}) \\ H_r(x^d_i) & H_r(x^d_{Ni}) \end{bmatrix} \quad \text{if } i \in S \text{ and } H_i = \begin{bmatrix} H_r(x^d_i) \end{bmatrix} \quad \text{if } i \notin S.
\]

Consider the following problem

\[
\text{minimize } \sum_{i \in S} J_1(x_i, u_i) + \sum_{i \in \mathcal{V} \setminus S} J_1(x_i) \quad \text{(4)}
\]

subject to

\[
H_i g_i = \text{col}(u_{\text{ini}}, x_{\text{ini}}, x_{Ni}, x_{i}, x_i), \quad i \in S, 
H_i g_i = \text{col}(x_{Ni}, x_N, x_i), \quad i \notin S.
\]

We now proceed with the statement and proof of our main result, that (2) and (4) are equivalent.
Theorem IV.4. (Equivalent Network Optimization): Consider the system (1) and sample trajectory $w^d$ satisfying the hypotheses of Proposition IV.2 and let $x^0 = Ax^{ini}(T_{ini} - 1) + Bu^{ini}(T_{ini} - 1)$. If $(u^{1,*}, x^{2,*})$ and $(g^*, u^{s,*}, x^{2,*})$ are the unique optimizers of (2) and (4), respectively, then $u^{1,*} = u^{2,*}$ and $x^{1,*} = x^{2,*}$.

Proof. Let

$$u^{2,*}(t) = \begin{cases} \text{col}(u^{ini}(t), x^{ini}(t)) & 0 \leq t < T_{ini} \\ \text{col}(u^{2,*}(t - T_{ini}), x^{2,*}(t - T_{ini})) & T_{ini} \leq t < \tau \end{cases}$$

and note that by rearranging the rows of $H_i$, it can be seen that $H_i(u^*_i, g^*_i) = u^{2,*}$ and $H_i(x^{2,*})$ is the state trajectory corresponding to the input $u^{2,*}$ and therefore

$$x^{2,*}(0) = Ax^{ini}(T_{ini} - 1) + Bu^{ini}(T_{ini} - 1) = x^0$$

and $x^{2,*}(t + 1) = Ax^{2,*}(t) + Bu^{2,*}(t)$ for $0 \leq t < T_{agr}$. Hence $u^{2,*}$ is feasible for (2). Likewise, if $u_i^{ini}(t), u^{s,*}$ is the state trajectory corresponding to the input $u^{ini}(t), u^{s,*}$ and $H_i(u^{ini}(t), g^*_i) = u^{s,*}$, where

$$u^{s,*}(t) = \begin{cases} \text{col}(u^{ini}(t), x^{ini}(t)) & 0 \leq t < T_{ini} \\ \text{col}(u^{s,*}(t - T_{ini}), x^{s,*}(t - T_{ini})) & T_{ini} \leq t < \tau \end{cases}$$

so that

$$H_i g_i = \begin{cases} \text{col}(u^{ini}, u^{s,*}, x^{ini}, x^{s,*}, x^{ini}, x^{s,*}) & i \in S \\ \text{col}(x^{ini}, x^{s,*}, x^{ini}, x^{s,*}) & i \notin S \end{cases}$$

so $(g^*, u^{s,*}, x^{s,*})$ is feasible for (4). The objective in (2) and (4) is the same so $u^{s,*} = u^{s,*}$ and $x^{s,*} = x^{s,*}$, as desired.

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 e.g., [22, 23]. In Section V-B below, we employ a primal-dual dynamic to find asymptotically a solution of (4) in a distributed way.

Remark IV.5. (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 $T \geq (n + m + 1)(T_{ini} + T_{agr}) - 1$. Assuming that $T_{agr}, T_{ini} \sim O(1)$, we have $T \sim O(n + mn)$. 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 $O(n + mn)$. However, using the distributed optimization algorithm of Section V-B agents only need to share peer-to-peer messages of size $O(k_i)$.

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 $u^{ini}$ and $x^{ini}$ 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

Under the hypotheses of Theorem IV.4 it is known, cf. [24, Theorem 12.2], that the controller corresponding to a receding horizon implementation of (4) exponentially stabilizes the system. However, 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. We assume without loss of generality $T_{ini} = 1$ (the result is valid for arbitrary $T_{ini}$ but the notation gets more involved).

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

Proof. Let $\phi_{\delta}(x) : \mathbb{R}^n \to \mathbb{R}^m$ be the feedback corresponding to a receding horizon implementation of (4). Consider $x(t + 1) = f(x(t), v(t))$, where $f(x, v) = A + B\phi_{\delta}(x) + Bv$. Note that the origin is exponentially stable with respect to dynamics of the unforced system $x(t + 1) = f(x(t), 0)$. Moreover, $\phi_{\delta}$ is continuous and piecewise affine on $\mathbb{R}^n$, so $f$ is globally Lipschitz in $x$ and $v$. By [25, Lemma 4.6], the system is input-to-state stable (ISS) with respect to $v$ and there is a constant $\gamma > 0$ and a class $\mathcal{K}_\infty$ function $\beta$ such that for all $t \in \mathbb{Z}_{\geq 0}$,

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

Let $x : \mathbb{Z}_{\geq 0} \to \mathbb{R}^n$ be a trajectory of the closed-loop dynamics of (1) with the controller described by Algorithm 1 where $\delta < \delta^* = \gamma^{-1}$ and define $v(t) = \phi_{\delta}(x(t)) - \phi_{\delta}(x(t))$. It follows that $x(t + 1) = f(x(t), v(t))$. Note that $\phi_{\delta}(x(t)) = $ ...
$u^*(0; u_{i0}, x_{i0})$, where $u_{i0} = \phi_i(x(t-1))$ and $x_{i0} = x(t-1)$ so it follows that
\[
\|v(t)\| = \|\hat{u}(0; u_{i0}, x_{i0}) - u^*(0; u_{i0}, x_{i0})\| \leq \delta \|x(t-1)\|.
\]
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 $t \geq T_k + 1$:
\[
\|x(t)\| \leq \beta((k+1)(\gamma \delta)^k, t) + \gamma \sup_{T_k+1 < \tau \leq t} \|v(\tau)\|
\]
\[
\leq \beta((k+1)(\gamma \delta)^k, t) + (k+1)(\gamma \delta)^k+1,
\]
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
\[
\limsup_{t \to \infty} \|x(t)\| \leq \limsup_{k \to \infty}(k+1)(\gamma \delta)^k = 0.
\]
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 $t > 0$,
\[
\|x(t)\| \leq (1 - \gamma \delta)\eta + \gamma \sup_{0 \leq \tau \leq t} \|v(\tau)\|
\]
\[
\leq (1 - \gamma \delta)\eta + \gamma \delta \|x(t-1)\|.
\]
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 use the shorthand notation introduced in Remark IV.5, note $z_i \in \mathbb{R}^{d_i}$, where $d_i = T - (T_{m} + T_{k}) + 1 + n_i + m_i$ for $i \in S$ and $d_i = T - (T_{m} + T_{k}) + 1 + n_i$ otherwise. Problem (4) can be written as
\[
\begin{align*}
\text{minimize} & \quad \sum_{i \in V} z_i^T Q_i z_i \\
\text{subject to} & \quad A_i z_{N_i} = b_i,
\end{align*}
\]
for suitable $Q_i \in \mathbb{R}^{d_i \times d_i}$, $Q_i > 0$, $A_i \in \mathbb{R}^{c_i \times d_i}$, and $b_i \in \mathbb{R}^{c_i}$, with $c_i \in \mathbb{Z}$. Under Slater’s condition [26] (which holds since the controllability of the system along with the persistance of excitation make the problem strictly feasible), there is a unique optimizer $z^\star$. The Lagrangian of (5) is
\[
\mathcal{L}(z, \lambda) = \sum_{i \in V} z_i^T Q_i z_i + \lambda_i^T (A_i z_{N_i} - b_i).
\]
If $\lambda^\star$ is an optimizer of the dual problem, then the pair $(z^\star, \lambda^\star)$ is a (min-max) saddle point of $\mathcal{L}$, meaning that $\mathcal{L}(z^\star, \lambda^\star) \leq \mathcal{L}(z^\star, \lambda^\star^\prime) \leq \mathcal{L}(z^\star, \lambda^\star)$ 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, can be used to compute the optimizer. Here, $F_{ij} \in \mathbb{R}^{(d_i + \sum_{i \in N_j} d_i) \times d_i}$ is the matrix such that $F_{ij} z_{N_j} = z_i$. By Corollary 4.5 the flow converges asymptotically to a saddle point of $\mathcal{L}$, and since the optimizer for the primal problem is unique, this means that $z$ converges to $z^\star$. 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 $j \in N_i$, then the message agent $j$ shares with agent $i$ consists of $\text{col}(x_j, \lambda_j) \in \mathbb{R}^{n_j T_{w} + k_j}$, which is $O(k_i)$ (cf. Remark IV.5).

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

**Proposition V.2. (Suboptimality Certificate):** Let $y = \text{col}(z, \lambda)$, $Q = \text{diag}(Q_1, Q_2, \ldots, Q_N)$, $A = [F_1^T A_{11}^T, F_2^T A_{12}^T, \ldots, F_N^T A_{1N}^T]^T$, and $b = \text{col}(b_1, b_2, \ldots, b_N)$, and define
\[
M = \begin{bmatrix}
-2Q^T & -A^T \\
A & 0
\end{bmatrix} \begin{bmatrix}
q & 0 \\
0 & b
\end{bmatrix}.
\]
Under the flow given by (6), if $\|\text{col}(\dot{z}_i, \dot{\lambda}_i)\| < \rho$ for all $i \in V$, where $\rho = \frac{\sqrt{2}}{\|M\|^{1/2}}$, then $\|z - z^\star\| < \epsilon$.

**Proof.** For all dual optimizers, $\lambda^\star$, we have $M y^\star + q = 0$ and $\|M^T M (y - y^\star)\| \leq \|M^T\| \|M y + q\|$, where $y^\star = \text{col}(z^\star, \lambda^\star)$. Since $M^T M$ is the orthogonal projector onto the image of $M^T$ along ker $M$, we have
\[
\|M^T M (y - y^\star)\| \leq \min_{\hat{g} \in \text{ker } M^T} \|y - \hat{g}\|.
\]
Next, because
\[
\|z - z^\star\|^2 \leq \min_{\hat{g} \in \text{ker } M^T} \|y - \hat{g}\|^2 \leq \|M^T\| \|M y + q\|^2
\]
and $\|M y + q\|^2 = \sum_{i \in V} \|\text{col}(z_i, \lambda_i)\|^2$, the result follows.

We remark that the suboptimality certificate can be checked in a fully distributed manner using information locally available to each agent provided that $\|M^T\|$ is known. Because $M$ depends only on the objective $Q$ and constraints $A$, which in turn comes from the sample trajectory $u^d$, it can be computed offline. Finally, it is also possible for each agent to compute a bound on $\|M^T\|$ using the fact that, for $A = [A_1^T, A_2^T]^T$, one has $\|A_i\| \leq \|A_1\|$. It follows that for all $i \in V$,
\[
\|M_i\| \leq \left\| \begin{bmatrix}
-2Q_i & F_{ii}^T A_{i1} & F_{ij1} A_{j1} & \cdots & F_{ij_{N_i}} A_{j_{N_i}}
\end{bmatrix} \right\|.
\]

For (a), $A, B$ are controllable. We use $T_{\text{fr}} = 5$, $T_{\text{m}} = 1$ and $T = (m + 1)(T_{\text{m}} + T_{\text{fr}} + n)$. The input sequence is chosen as $u^d(t) = K x^d(t) + w(t)$. Here $K$ is a matrix so that $A + BK$ is marginally stable (the data does not need to be generated from
model, agents learn a non-parametric representation based on unknown system matrices. Instead of building a dynamic controller for stabilizing network linear dynamics described by high order, and partial observations of the network state.

The solution of a network optimization problem solved in a Figure 1(b).

Fig. 1: Performance of distributed data-based predictive controller. Here, $Q = R = I_n$ and actuated (resp. not actuated) nodes are in blue (resp. in black).

VI. CONCLUSIONS 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 solution of a network optimization problem solved in a receding horizon manner and in a distributed way. Future research will characterize the robustness properties of the introduced control scheme, investigate ways of improving the scalability of the proposed scheme with network size, 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.

REFERENCES

[1] F. Bullo, J. Cortés, and S. Martinez, Distributed Control of Robotic Networks. Applied Mathematics Series, Princeton University Press, 2009.

[2] M. Mesbah and M. Egerstedt, Graph Theoretic Methods in Multiagent Networks. Applied Mathematics Series, Princeton University Press, 2010.

[3] R. R. Negenborn and J. M. Maestre, “Distributed model predictive control: An overview and roadmap of future research opportunities,” IEEE Control Systems Magazine, vol. 34, no. 4, pp. 78–87, 2014.

[4] M. Rotkowitz and S. Lall, “A characterization of convex problems in decentralized control,” IEEE Transactions on Automatic Control, vol. 51, no. 2, pp. 274–286, 2006.

[5] F. Lin, M. Fardad, and M. R. Jovanovic, “Design of optimal sparse feedback gains via the alternating direction method of multipliers,” IEEE Transactions on Automatic Control, vol. 58, no. 9, pp. 2426–2431, 2013.

[6] G. Fazelzamia, R. Madani, A. Kalbat, and J. Lavaei, “Convex relaxation for optimal distributed control problems,” IEEE Transactions on Automatic Control, vol. 62, no. 1, pp. 206–221, 2017.

[7] L. Furieri, Y. Zheng, A. Papachristodoulou, and M. Kamgarpour, “On separable quadratic Lyapunov functions for convex design of distributed controllers,” in European Control Conference, (Naples, Italy), pp. 42–49, 2019.

[8] J. Kober, J. A. Bagnell, and J. Peters, “Reinforcement learning in robotics: A survey,” International Journal of Robotics Research, vol. 32, no. 11, pp. 1238–1274, 2013.

[9] L. Bu, R. Babu, B. D. Schutter, et al., “A comprehensive survey of multijagent reinforcement learning,” IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), vol. 38, no. 2, pp. 156–172, 2008.

[10] B. Recht, “A tour of reinforcement learning: The view from continuous control,” Annual Review of Control, Robotics, and Autonomous Systems, vol. 2, pp. 253–279, 2019.

[11] D. Jia and B. Krogh, “Min-max feedback model predictive control for distributed control with communication,” in American Control Conference, (Anchorage, AK), pp. 4507 – 4512 vol.6, 2002.

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

[13] A. N. Venkat, I. A. Hiskens, J. B. Rawlings, and S. J. Wright, “Distributed MPC strategies with application to power system automatic generation control,” IEEE Transactions on Control Systems Technology, vol. 16, no. 6, pp. 1192–1206, 2008.

[14] B. T. Stewart, A. N. Venkat, J. B. Rawlings, S. J. Wright, and G. Panocchia, “Cooperative distributed model predictive control,” Systems & Control Letters, vol. 59, no. 8, pp. 460–469, 2010.

[15] L. Ljung, System Identification: Theory for the User. Prentice Hall information and system sciences series, Prentice Hall, 1999.

[16] J. C. Willems, P. Rapisarda, I. Markovsky, and B. L. M. D. Moor, “A note on persistency of excitation,” Systems & Control Letters, vol. 54, no. 4, pp. 325–329, 2005.

[17] U. Park and M. Ikeda, “Stability analysis and control design of discrete-time systems by the direct use of time series data,” Automatica, vol. 45, no. 5, pp. 1265–1271, 2009.

[18] T. M. Maupong and P. Rapisarda, “Data-driven control: A behavioral approach,” Systems & Control Letters, vol. 101, pp. 37–43, 2017.

[19] C. D. Persis and P. Tesi, “Formulas for data-driven control: Stabilization, optimality and robustness,” arXiv preprint arXiv:1903.06842, 2019.

[20] J. Coulson, J. Lygeros, and F. Dörfler, “Data-enabled predictive control: in the shallows of the DeePC,” in European Control Conference, (Naples, Italy), pp. 307–312, 2019.

[21] J. Berberich, J. Köhler, A. M. Müller, and F. Allgöwer, “Data-driven model predictive control with stability and robustness guarantees,” arXiv preprint arXiv:1906.04679, 2019.

[22] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, “Distributed optimization and statistical learning via the alternating direction method of multipliers,” Foundations and Trends in Machine Learning, vol. 3, no. 1, pp. 1–122, 2011.

[23] A. Cherukuri, B. Gharesifard, and J. Cortés, “Saddle-point dynamics: conditions for asymptotic stability of saddle points,” SIAM Journal on Control and Optimization, vol. 55, no. 1, pp. 486–511, 2017.

[24] F. Borrelli, A. Bemporad, and M. Morari, Predictive Control for Linear and Hybrid Systems. Cambridge, UK: Cambridge University Press, 2017.

[25] H. K. Khalil, Nonlinear Systems. Prentice Hall, 3 ed., 2002.

[26] S. Boyd and L. Vandenberghe, Convex Optimization. Cambridge University Press, 2004.

[27] M. E. Newman and D. J. Watts, “Renormalization group analysis of the small-world network model,” Physics Letters A, vol. 263, no. 4-6, pp. 341–346, 1999.