IEEE CSS

Received 18 October 2022; revised 27 January 2023 and 21 February 2023; accepted 15 March 2023. Date of publication 20 March 2023; date of current version 4 April 2023. Recommended by Senior Editor Yue Wang.

Digital Object Identifier 10.1109/OJCSYS.2023.3259228

Distributed Data-Driven Control of Network Systems

FEDERICO CELI[1] (Student Member, IEEE), GIACOMO BAGGIO[2] (Member, IEEE), AND FABIO PASQUALETTI[1] (Member, IEEE)

[1]Department of Mechanical Engineering, University of California, Riverside, CA 92521, USA
[2]Department of Information Engineering, University of Padua, 35131 Padua, Italy

CORRESPONDING AUTHOR: FEDERICO CELI (e-mail: fceli001@ucr.edu).

This work was supported by AFOSR under Awards FA9550-19-1-0235, AFOSR-FA9550-20-1-0140, and ARO W911NF-20-2-0267.

ABSTRACT Imperfect models lead to imperfect controllers and deriving accurate models from first principles or system identification is especially challenging in networked systems. Instead, data can be used to directly compute controllers, without requiring any system identification or modeling. In this paper we propose a strategy to directly learn control actions when data from past system trajectories is distributed among multiple agents in a network. The approach we develop provably converges to a suboptimal solution in a finite number of steps, bounded by the diameter of the network, and with a sub-optimality gap that can be characterized as a function of data, and that can be made arbitrarily small. We further characterize the robustness properties of our approach and give provable guarantees on its performance when data are affected by noise or by a class of attacks.

INDEX TERMS Distributed control and optimization, learning for control, network analysis and control, optimal control.

I. INTRODUCTION

Networks and multi-agent systems have been studied extensively in the control community in the context of formation control [1], [2], consensus algorithms [3], and coordination of multi-agent systems [4], to cite a few. These results have seen application in various real-world scenarios from robotics [5], [6] to power grids [7]. Together with its opportunities, modeling the interaction between possibly heterogeneous systems through networks presents practical challenges. In fact, building accurate models of large networks is a burdensome task and modeling errors (e.g., missing or extra links, incorrect link weights) are often unavoidable [8], [9]. Network models can be built either through first principles or via system identification. In the former, a dynamical model is deduced from the physical properties of the system, which are often not fully characterized. Alternatively, system identification can be used when data are available [10], [11]. Unfortunately, this approach, too, has limitations and has returned a set of mixed results in application to modern network systems [12], [13], [14]. This is especially problematic when the end goal of system identification is to provide the basis for controller design since errors in the modeling phase can compound in the controller design phase. In this work, we propose a model-agnostic control approach for a network system. That is, we bypass the system modeling phase altogether by using data to directly learn suitable control actions.

The *direct* data-driven control literature leverages data to design controllers in a one-step solution, as opposed to the *indirect* approach, where data are used to identify a model that is then used to design a controller. Although direct data-driven controls can be dated as far back as [15], they have recently been the subject of renewed interest from the control community. A modern approach to designing direct data-driven controls includes data-driven open-loop optimal control [16], [17], [18], closed-loop and robust control [19], [20], [21], [22], and predictive and nonlinear control [23], [24], [25], [26], [27]. The problem of learning optimal controls in network systems has remained, however, relatively unexplored. To the best of our knowledge, the first result in this direction is [28], soon followed by [29] and [30]. In [28], the authors propose a modified version of the DeePC algorithm [23] to stabilize a network system through a primal-dual flow (while

minimizing a quadratic function of the inputs and the states). This strategy is based on a Model Predictive Control approach (MPC) where a behavioral systems representation built upon a recorded system trajectory is used in place of the system's model. A limit of this approach is that it assumes that input-state data from past trajectories, as well as the current state, can be freely shared among neighboring agents. In addition, the distribution of the primal-dual flow among the agents in the network requires a substantial number of messages to be shared [35]. This assumption is limiting because it presupposes that the time scales of communication and computation processes are significantly shorter than that of the controller action. An alternative approach appeared in [30], where the same problem is tackled through the system level synthesis framework (SLS) [36]. SLS parameterizes closed-loop system responses directly from collected open-loop trajectories. The result of [30] reduces to a distributed computation of a dynamic state-feedback controller via the alternating direction method of multipliers (ADMM) [37]. This approach shares much of the limitations of [28], namely the necessity of high frequency communication between nodes in the network, and the need to share input-state trajectories among the networks' nodes. Further recent results on the control of interconnected systems include [31], [32], [33], [34]. However, these works focus on finding stabilizing controllers, and do not include optimization over an objective function, robustness bounds, nor explicit data-driven formulas.

The contribution of this paper is a data-driven and distributed algorithm to learn optimal controls in a multi-agent environment. Our approach is data-driven as the control design relies exclusively on prerecorded input-state trajectories of an unknown linear system, and it is distributed as it assumes that the recorded trajectories are not available to any single agent, but are partitioned throughout the network. The goal of the agents is to compute the control action which globally minimizes a given quadratic cost function of the states and inputs, from a given initial condition. To do this, we leverage an algorithm based on iterative projections in order to distribute the computation of the control in a network of agents with partial access to data. This work builds upon and significantly expands the approach presented in [29] and departs from the cited literature in a number of ways. First, it does not require data from past recordings to be shared (at least directly) among the networks' nodes, offering an implicit layer of privacy. Moreover, it relies on a closed-form expression that provably converges to a solution, with a prescribed distance from optimality, after a finite number of iterations of the algorithm bounded by the diameter of the network. Finally, while the aforementioned papers consider the case of noiseless data, in this work we also study the robustness of the approach when the collected data are injected with noise or adversarial attacks.

The remainder of the paper is organized as follows. Section II contains the problem setup and necessary preliminary notions. In Section III we derive closed-form solutions to the problem of designing optimal data-driven controls and present our distributed algorithm. In Section IV we derive results on the robustness of the proposed approach to noisy data, and in Section V we numerically validate our results and compare them with alternative approaches. We conclude our paper in Section VI and leave the proofs to the Appendices.

II. PROBLEM SETUP

A. NOTATION

Let $\mathbb{R}$ ($\mathbb{N}$) and $\mathbb{R}^+$ ($\mathbb{N}^+$) denote the set of real (integer) and strictly positive real (integer) numbers, respectively. Given a matrix $A \in \mathbb{R}^{n\times m}$, we let $\text{Rank}(A)$, $\text{Basis}(A)$, $\text{Ker}(A)$, A^T, $\sigma_{\min}(A)$ denote the rank, a basis of the column space, the kernel, the transpose, and the smallest singular value of A, respectively. We let $\text{blkdiag}(A_1, \dots, A_n)$ be the block diagonal matrix with blocks $A_i \in \mathbb{R}^{n_i\times m_i}$. For a matrix $A \in \mathbb{R}^{n\times m}$, and for $p < n$ and $q < m$, we let $[A]_{1:p,1:q}$ be the matrix formed by the first p rows and q columns of A. For any matrix A, we denote its Moore-Penrose pseudoinverse as $A^\dagger$. We let $A \succ 0$ ($A \succeq 0$) denote a positive definite (positive semidefinite) matrix. The spectral norm of matrix A is $\|A\|$, and the Kronecker product between matrices A and B is denoted by $A \otimes B$. I_n and $0_{n,m}$ stand for the $n \times n$ identity matrix and $n \times m$ zero matrix, respectively (subscripts will be omitted when clear from the context). Given a sequence of vectors $x(t) \in \mathbb{R}^n$, with $t \in \{1, \dots, T\}$, we let $\text{col}(x(1), \dots, x(T)) = [x(1)^\top \cdots x(T)^\top]^\top \in \mathbb{R}^{nT}$. Let $\text{vec}(\cdot) : \mathbb{R}^{n\times m} \to \mathbb{R}^{nm}$ denote the vectorization operator of a matrix. Given two vectors $v, w \in \mathbb{R}^n$, we say that they are orthogonal if $v^\mathsf{T} w = 0$, and indicate this as $v \perp w$. To stress the fact that two vectors are not orthogonal we shall use $v \not\perp w$. For a random variable $x : \Omega \to \mathbb{R}$, we let $\mathbb{P}[x \in S]$ and $\mathbb{E}[x]$ be the probability that x takes on a value in a set $S \subseteq \mathbb{R}$ and the mean or expected value of x, respectively. Finally, we say that a function $f : \mathbb{R} \to \mathbb{R}$ grows sublinearly with x if $\lim_{x\to\infty} |f(x)|/x = 0$.

B. PROBLEM DEFINITION

We study the controllable linear system

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

where $x \in \mathbb{R}^n$ and $u \in \mathbb{R}^m$ are the state and the input vectors at time $t \in \mathbb{N}^+$, respectively, and with $A \in \mathbb{R}^{n\times n}$ and $B \in \mathbb{R}^{n\times m}$. We seek to compute the input sequence $\mathbf{u}_T = \text{col}(u(0), \dots, u(T-1))$ that minimizes the following finite-horizon quadratic objective function:

$$\begin{aligned} \underset{\mathbf{u}_T}{\arg\min} \quad & \mathbf{x}_T^\mathsf{T} Q \mathbf{x}_T + \mathbf{u}_T^\mathsf{T} R \mathbf{u}_T \\ \text{subject to} \quad & x(t+1) = Ax(t) + Bu(t) \\ & x(0) = x_0, \end{aligned} \tag{2}$$

where $Q \in \mathbb{R}^{n(T+1)\times n(T+1)} \succeq 0$, $R \in \mathbb{R}^{mT\times mT} \succ 0$ are the state and input weighting matrices, respectively, and $\mathbf{x}_T = \text{col}(x(0), \dots, x(T))$. When the matrices Q and R in (2) are block diagonal the minimization problem (2) is referred to as the finite-horizon discrete-time linear quadratic regulator (LQR) problem [38].

In solving the minimization problem (2) we assume that the pair (A, B) is not known. We assume, however, that a series of experiments has been performed to learn the solution of (2) directly through experimental data. In particular, we assume that N experiments of length T have been performed and that the experimental data are

$$X_{1:T} = \begin{bmatrix} \mathbf{x}_T^1 & \cdots & \mathbf{x}_T^N \end{bmatrix}, \tag{3a}$$

$$U = \begin{bmatrix} \mathbf{u}_T^1 & \cdots & \mathbf{u}_T^N \end{bmatrix}, \tag{3b}$$

$$X_0 = \begin{bmatrix} x_0^1 & \cdots & x_0^N \end{bmatrix}, \tag{3c}$$

where $X_{1:T} \in \mathbb{R}^{nT \times N}$ contains the state trajectories, $U \in \mathbb{R}^{mT \times N}$ the input sequences, and $X_0 \in \mathbb{R}^{n \times N}$ the initial states of the N experiments: $\mathbf{x}_T^j = \text{col}(x^j(1), \ldots, x^j(T))$, $\mathbf{u}_T^j = \text{col}(u^j(0), \ldots, u^j(T-1))$, and $x_0^j = x^j(0)$ of the jth experiment.

Notice that we can write the state evolution of (1) from initial condition x_0 and subject to the input sequence $\mathbf{u}_T$ as

$$\mathbf{x}_T = \underbrace{\begin{bmatrix} I \\ A \\ A^2 \\ \vdots \\ A^T \end{bmatrix}}_{O_T} x_0 + \underbrace{\begin{bmatrix} 0 & \cdots & 0 & 0 \\ B & \cdots & 0 & 0 \\ AB & \cdots & 0 & 0 \\ & \ddots & & \\ A^{T-1}B & \cdots & AB & B \end{bmatrix}}_{F_T} \mathbf{u}_T. \tag{4}$$

Thus, without measurement noise, the data matrices satisfy

$$X := \begin{bmatrix} X_0 \\ X_{1:T} \end{bmatrix} = \begin{bmatrix} O_T & F_T \end{bmatrix} \begin{bmatrix} X_0 \\ U \end{bmatrix}. \tag{5}$$

We assume that the following holds throughout the paper.

Assumption II.1 (Persistency of excitation of data): The experimental inputs and initial conditions satisfy

$$\text{Rank} \begin{bmatrix} X_0 \\ U \end{bmatrix} = n + mT.$$

□

Assumption II.1 is standard in data driven studies [20], [21] and places a lower bound on the number of experiments needed to build exact data-driven expressions, namely, $N \geq n + mT$, where T is the length of each experiment [39].

In the absence of a model, the minimization problem (2) can be solved using experimental data as shown in [20], [23], [40]. However, these works assume that the experimental data are collectively available. Instead, motivated by the increasing interest in data-driven control of multiagent and network systems, we assume that the experimental data are distributed across a set of agents, and we seek a solution to the minimization problem (2) that relies on distributed computation.

C. SETUP FOR MULTIAGENT LEARNING

Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be the graph associated with the matrix A in (1), where $\mathcal{V} = \{1, \ldots, n\}$ and $(i, j) \in \mathcal{E}$ if and only if $A_{ji} \neq 0$.

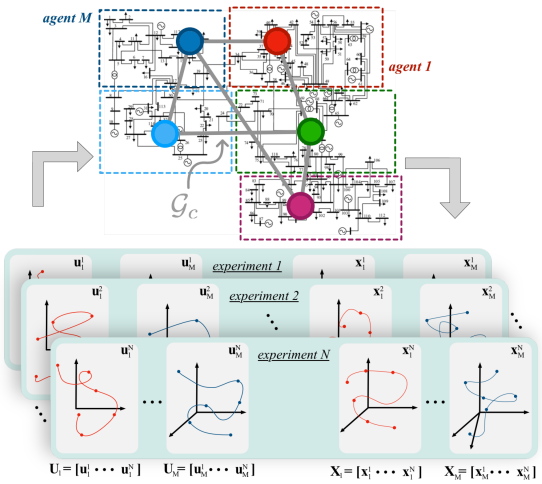

FIGURE 1. In this figure we show a visual depiction of the data collection phase. N experiments of length T are carried out over a network system. Each agent $i \in \{1, \ldots, M\}$ collects input-state trajectories of a subset of the network nodes, i.e., the initial state $x_{0,i}^j$, the state trajectory $\mathbf{x}_{T,i}^j$, and the input trajectory $\mathbf{u}_{T,i}^j$, for each experiment j.

Let $\mathcal{V}$ be partitioned as $\mathcal{V} = \mathcal{V}_1 \cup \cdots \cup \mathcal{V}_M$, with $|\mathcal{V}_i| = n_i$. Then, after reordering the nodes, the matrix A reads as

$$A = \begin{bmatrix} A_{11} & \ldots & A_{1M} \\ \vdots & \vdots & \vdots \\ A_{M1} & \ldots & A_{MM} \end{bmatrix}.$$

We assume[1] that the input matrix in (1) can be written as

$$B = \text{blkdiag}(B_1, \ldots, B_M). \tag{6}$$

Then, system (1) can equivalently be written as the interconnection of M subsystems of the form

$$x_i(t+1) = A_{ii}x_i(t) + \sum_{j=1,\, j\neq i}^{M} A_{ij}x_j(t) + B_i u_i(t), \tag{7}$$

where $x_i \in \mathbb{R}^{n_i}$ and $u_i \in \mathbb{R}^{m_i}$ are the states and inputs, respectively, of the nodes in $\mathcal{V}_i$.

We assume the presence of M agents, each one responsible for one subsystem (see also Fig. 1). In particular, agents are interconnected according to a directed communication graph $\mathcal{G}_c = (\mathcal{V}_c, \mathcal{E}_c)$, and agent i selects the input u_i to the ith subsystem using local data and information exchanged with neighboring agents $\mathcal{N}_i = \{j : (i, j) \in \mathcal{E}_c\}$. The local data available to agent i are

$$X_i = \begin{bmatrix} \mathbf{x}_{T,i}^1 & \cdots & \mathbf{x}_{T,i}^N \end{bmatrix}, \tag{8a}$$

$$U_i = \begin{bmatrix} \mathbf{u}_{T,i}^1 & \cdots & \mathbf{u}_{T,i}^N \end{bmatrix}, \tag{8b}$$

$$X_{0,i} = \begin{bmatrix} x_{0,i}^1 & \cdots & x_{0,i}^N \end{bmatrix}, \tag{8c}$$

[1] Although some of our results hold also for general input matrices, this assumption simplifies the presentation and is also common in distributed data-driven studies [28], [30].

that is, the components indexed by $\mathcal{V}_i$ of the state trajectories, control inputs, and initial conditions of the N experiments. Thus, after a possible reordering of the system states,

$$X = \begin{bmatrix} X_1 \\ \vdots \\ X_M \end{bmatrix}, \; U = \begin{bmatrix} U_1 \\ \vdots \\ U_M \end{bmatrix}, \text{ and } X_0 = \begin{bmatrix} X_{0,1} \\ \vdots \\ X_{0,M} \end{bmatrix}.$$

Agents cooperate to collectively compute the local inputs that solve the minimization problem (2). A naive solution to this problem requires the agents to share all their local experimental data with all other agents, and then employ centralized data-driven formulas for the solution of the minimization problem (2) (see [20], [23], [40] and Theorem III.1 below). Instead, we will develop algorithms that require the agents to exchange reduced information to compute, in finite time, a solution to the minimization problem (2) in a distributed manner.

III. MULTIAGENT LEARNING OF OPTIMAL CONTROLS

We start by providing a data-driven solution to the minimization problem (2) when there is only one agent. This solution is general, and appeared previously in [40]. In particular, the optimal control $\mathbf{u}_T^*$ minimizing (2) can be computed in closed form as

$$\mathbf{u}_T^* = -UK_0 S^\dagger (XK_0)^\top Q X K_U (X_0 K_U)^\dagger x_0, \tag{9}$$

with

$$S = (XK_0)^\top Q (XK_0) + (UK_0)^\top R (UK_0),$$

and $K_0 = \text{Basis}(\text{Ker}(X_0))$ and $K_U = \text{Basis}(\text{Ker}(U))$, highlighting the dependency of the optimal controls on the data. We now give an alternative formula to compute $\mathbf{u}_T^*$ in closed form. This formula requires a stronger assumption on Q, but is more compact and holds in most scenarios.

Theorem III.1 (Single-agent data-driven solution (2)): Let X_0, X and U be as in (3) and (5), and satisfying Assumption II.1. Define $P \in \mathbb{R}^{(n+mT)\times(n+mT)}$ as

$$P = \left(X \begin{bmatrix} X_0 \\ U \end{bmatrix}^\dagger \right)^\mathsf{T} Q \left(X \begin{bmatrix} X_0 \\ U \end{bmatrix}^\dagger \right) + \begin{bmatrix} 0_{n\times n} & 0 \\ 0 & R \end{bmatrix}. \tag{10}$$

If $Q^{1/2}O_T$ has full column rank the matrix P is invertible, then the solution $\mathbf{u}_T^*$ to (2) can be extracted from

$$\begin{bmatrix} x_0 \\ \mathbf{u}_T^* \end{bmatrix} = P^{-\frac{1}{2}} \left(\begin{bmatrix} I & 0 \end{bmatrix} P^{-\frac{1}{2}} \right)^\dagger x_0, \tag{11}$$

with $x(0) = x_0$, □

Theorem III.1 provides a data-driven expression of the optimal control input for the minimization problem (2) alternative to (9). The assumption that $Q^{1/2}O_T$ has full column rank is required to ensure the invertibility of P and is typically satisfied in practice. For instance, it is satisfied when $Q \succ 0$, or, in the case of block diagonal Q with diagonal blocks $Q_d \in \mathbb{R}^{n\times n}$ (standard LQR setup), when the pair $(A, Q_d^{1/2})$ is observable. This assumption on $Q^{1/2}O_T$ is made in order to keep expression (9) more compact, which helps simplifying notation in the rest of the paper. If needed, however, this assumption can be lifted and we refer the interested reader to Appendix B. Both expressions (11) and (9) require the knowledge of all the experimental data, as well as the knowledge of the cost function, which is undesirable for interconnected and possibly large systems. While a distributed data-driven solution to the minimization problem (2) could be obtained using standard techniques for distributed optimization, e.g., see [28], [30], we follow a different approach that will lead to an algorithm with finite-time convergence guarantees. To this aim, we define the following auxiliary problems:

$$\begin{aligned} \underset{\alpha,\beta}{\arg\min} \quad & \left\| \begin{bmatrix} Q^{\frac{1}{2}}X & Q^{\frac{1}{2}}X \\ 0 & R^{\frac{1}{2}}U \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \end{bmatrix} \right\|^2 \\ \text{subject to} \quad & \begin{bmatrix} X_0 & 0 \\ U & 0 \\ 0 & X_0 \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \end{bmatrix} = \begin{bmatrix} x_0 \\ 0 \\ 0 \end{bmatrix} \end{aligned} \tag{12}$$

and

$$\begin{aligned} \underset{\alpha,\beta,v,w}{\arg\min} \quad & \left\| \begin{bmatrix} \alpha \\ \beta \\ v \\ w \end{bmatrix} \right\|^2 \\ \text{subject to} \quad & \begin{bmatrix} X_0 & 0 & 0 & 0 \\ U & 0 & 0 & 0 \\ 0 & X_0 & 0 & 0 \\ Q^{\frac{1}{2}}X & Q^{\frac{1}{2}}X & \varepsilon I_X & 0 \\ 0 & R^{\frac{1}{2}}U & 0 & \varepsilon I_U \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \\ v \\ w \end{bmatrix} = \begin{bmatrix} x_0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \end{aligned} \tag{13}$$

where $\varepsilon \in \mathbb{R}^+$ is a tunable parameter, with $I_X = I_{n(T+1)}$, $I_U = I_{mT}$ and therefore $\alpha \in \mathbb{R}^N$, $\beta \in \mathbb{R}^N$, $v \in \mathbb{R}^{n(T+1)}$, $w \in \mathbb{R}^{mT}$. We now characterize the feasibility and optimality properties of (12) and (13).

Lemma III.2 (Relationship between the solution of (2) and (12)): If α^* and β^* are minimizers of problem (12), then $\mathbf{u}_T^* = U\beta^*$ is the minimizer of problem (2). □

Lemma III.3 (Relationship between the solution of (2) and (13)): The minimization problem (13) is feasible and admits a unique solution when $\varepsilon > 0$. Furthermore, if $\alpha^*(\varepsilon)$, $\beta^*(\varepsilon)$, $v^*(\varepsilon)$ and $w^*(\varepsilon)$ are minimizers of problem (13), then

$$\lim_{\varepsilon \to 0^+} U\beta^*(\varepsilon) = \mathbf{u}_T^*, \tag{14}$$

where $\varepsilon > 0$, and $\mathbf{u}_T^*$ is the minimizer of problem (2). □

Since the constraints in the minimization problem (13) can be partitioned row-wise in a way that each row depends only on the data available to a single agent (see below), a distributed algorithm can be readily obtained. Further, given the equivalence between the minimization problems (2), (12), and (13) as stated in Lemma III.2 and Lemma III.3, a distributed

Algorithm 1: Distributed Data-Driven Optimal Control.

Input: $x_{0,i}$, W_i, d
$\gamma_i = W_i^{\dagger}\,\mathrm{col}(x_{0,i}, 0)$
$K_i = \mathrm{Basis}(\mathrm{Ker}(W_i))$
Until convergence
  for $j \in \mathcal{N}_i$ **do**
    Receive: γ_j, K_j
    $\gamma_i^+ = \gamma_i + \begin{bmatrix} K_i & 0 \end{bmatrix} \begin{bmatrix} -K_i & K_j \end{bmatrix}^{\dagger} (\gamma_i - \gamma_j)$
    $K_i = \mathrm{Basis}(\mathrm{Im}(K_i) \cap \mathrm{Im}(K_j))$
    $\gamma_i = \gamma_i^+$
  end
  Transmit: γ_i, K_i
end
Return: $\beta_i = [\gamma_i]_{N+1:2N}$

solution to (2) can be obtained by solving (13) in a distributed manner. Our distributed algorithm for the agents to solve the minimization problem (2) via distributed computation is in Algorithm 1, where W_i is defined in (15) and d denotes the diameter of $\mathcal{G}_c$.[2] We now provide an informal description of the algorithm:

(S1) Initially, each agent i computes the minimum norm solution to $W_i\gamma_i = \mathrm{col}(x_{0,i}, 0_{mT}, 0_n, 0_{n(T+1)}, 0_{mT})$, where $W_i \in \mathbb{R}^{(n_i+m_iT+n_i+n_i(T+1)+m_iT)\times(N+N+n(T+1)+mT)}$

$$W_i = \begin{bmatrix} X_{0,i} & 0 & 0 & 0 \\ U_i & 0 & 0 & 0 \\ 0 & X_{0,i} & 0 & 0 \\ Q_i^{\frac{1}{2}} X_i & Q_i^{\frac{1}{2}} X_i & \varepsilon I_X^i & 0 \\ 0 & R_i^{\frac{1}{2}} U_i & 0 & \varepsilon I_U^i \end{bmatrix}, \tag{15}$$

and $K_i = \mathrm{Basis}(\mathrm{Ker}(W_i))$. Here, I_X^i (resp. I_U^i) is a matrix whose rows are the rows of I_{nT} (resp. I_{mT}) corresponding to the indices that extract $Q_i^{\frac{1}{2}} X_i$ (resp. $R_i^{\frac{1}{2}} U_i$) from $Q^{\frac{1}{2}} X$ (resp. $R^{\frac{1}{2}} U$). From the notation in (13), let $\gamma_i = \mathrm{col}(\alpha_i, \beta_i, v_i, w_i) \in \mathbb{R}^{N+N+n(T+1)+mT}$ be such solution.

(S2) At each iteration, each agent i transmits γ_i and K_i to its neighboring agents j, and receives γ_j and K_j from each neighbor j.

(S3) At each iteration, each agent i updates γ_i and K_i as

$$\gamma_i^+ = \gamma_i + \begin{bmatrix} K_i & 0 \end{bmatrix} \begin{bmatrix} -K_i & K_j \end{bmatrix}^{\dagger} (\gamma_i - \gamma_j),$$
$$K_i = \mathrm{Basis}(\mathrm{Im}(K_i) \cap \mathrm{Im}(K_j)).$$

(S4) Convergence of this iterative procedure is guaranteed after a number of steps equal to the diameter of communication graph (see below). Upon convergence, each agent returns the vector $\beta_i = [\gamma_i]_{N+1:2N}$, extracted from γ_i at the algorithm's final iteration.

[2]The diameter of a graph $\mathcal{G}$ is the maximum distance between any two nodes of $\mathcal{G}$.

A high level walkthrough of the algorithm is in order. Step (S1) is simply the initialization step, in which we compute a preliminary solution $\gamma_i = W_i^{\dagger}\,\mathrm{col}(x_{0,i}, 0)$ which uses only local data and is feasible for agent i. In step (S2) neighboring agents share the information that is needed to update the provisory solution γ_i. In step (S3) γ_i is updated to γ_i^+ with information from its neighbors in such a way that γ_i^+ is a feasible solution for agent i and its neighbors. More specifically we find the new γ_i^+ to satisfy

$$\gamma_i^+ = W_i^{\dagger}\,\mathrm{col}(x_{0,i}, 0) + K_i\kappa_i \tag{16a}$$
$$= W_j^{\dagger}\,\mathrm{col}(x_{0,j}, 0) + K_j\kappa_j, \tag{16b}$$

for all $j \in \mathcal{N}$, where κ_i and κ_j are vectors of appropriate dimension. In the proof (cf. Appendix E) we show how κ_i and κ_j such that γ_i^+ satisfies (16) always exist, and show how this can be computed through the procedure described in (S3). Finally, (S4) gives one condition for ending the algorithm once it converges, which is also detailed in the proof.

Theorem III.4 (Distributed learning of data-driven optimal controls): Let $\mathcal{G}_c$ be a strongly connected communication graph. Let $\beta_i(\varepsilon)$ be the value returned by Algorithm 1 when W_i is as in (15), and let $\alpha^*(\varepsilon)$, $\beta^*(\varepsilon)$, $v^*(\varepsilon)$, $w^*(\varepsilon)$ be minimizers of problem (13), for some $\varepsilon > 0$. Then, for all $i \in \{1, \ldots, M\}$, $\beta_i(\varepsilon) = \beta^*(\varepsilon)$. □

From Theorem III.4, Algorithm 1 returns in a finite number of steps the solution of the minimization problem (13). Due to Lemma III.3, such solution yields the minimizer of (2) as the parameter ε decreases to zero. In fact, for any finite value of ε, the sub-optimality gap between the minimizer of (2) and the input $U\beta^*(\varepsilon)$ reconstructed from the minimizer of (13) can also be quantified. Let

$$Y = \begin{bmatrix} X_0 & 0 \\ U & 0 \\ 0 & X_0 \end{bmatrix}, H = \begin{bmatrix} Q^{\frac{1}{2}} X & Q^{\frac{1}{2}} X \\ 0 & R^{\frac{1}{2}} U \end{bmatrix}, \bar{x}_0 = \begin{bmatrix} x_0 \\ 0 \\ 0 \end{bmatrix}. \tag{17}$$

Lemma III.5 (Optimality gap of (14) for finite ε): Let $\beta^*(\varepsilon)$ be the minimizer of Problem (13), and let $\delta(\varepsilon) = \|\mathbf{u}_T^* - U\beta^*(\varepsilon)\|$. Then,

$$\delta(\varepsilon) \leq \left\| U \begin{bmatrix} 0_{N,N} & I_N \end{bmatrix} K_Y Z(\varepsilon) K_Y^{\top} H^{\top} H Y^{\dagger} x_0 \right\|, \tag{18}$$

where $Z(\varepsilon) = (K_Y^{\top}(\varepsilon^2 I + H^{\top}H)K_Y)^{\dagger} - (K_Y^{\top} H^{\top} H K_Y)^{\dagger}$, and $K_Y = \mathrm{Basis}(\mathrm{Ker}\, Y)$. □

Using Lemma III.5, Algorithm 1 can be used to compute a sub-optimal solution to (2) in a finite number of distributed calculations and within any desired sub-optimality guarantee. In fact, once β_i is computed by each agent i, the sub-optimal and local control at each agent is simply $\mathbf{u}_{T,i} = U_i\beta_i$.

Remark III.6 (Convergence of Algorithm 1): Algorithm 1 converges after d iterations, where d is the diameter of $\mathcal{G}_c$ (cf. Appendix E). When d is not available the algorithm can be stopped after n iterations, since n is always available to each agent through the size of the vector v_i, and $d \leq n$ always holds.

IV. ROBUSTNESS OF DATA-DRIVEN CONTROL INPUTS

So far, we discussed a distributed approach to learn optimal controls in a multiagent environment through noiseless data. This assumption on the data is often too restrictive in practice. In what follows we study how noisy datasets affect the results of Section III, assuming different degrees of knowledge on the noise distribution. In particular, we are interested in characterizing the robustness of our approach by bounding, in probability, the distance between the true cost of the optimal control problem in (2) and the one computed through our data-driven approach with a noisy dataset.

Assume that the state trajectory in (3) is collected as

$$\tilde{X}_{1:T} = X_{1:T} + \Delta_X,$$

where $X_{1:T}$ is the ground truth data and Δ_X is a matrix containing stochastic perturbations. We highlight that perturbation Δ_X affects the state trajectory at times $t = \{1, \dots, T\}$. However, the analysis that follows can be carried out in a similar fashion for noisy initial states and inputs, as well as for alternative data-driven expressions, e.g., (9). Let $F : \mathbb{R}^{mT\times N} \times \mathbb{R}^{n\times N} \times \mathbb{R}^{n(T-1)\times N} \to \mathbb{R}^{mT+n}$ be the data-driven map (11). We define the perturbed control map

$$\begin{bmatrix} x_0 \\ \tilde{\mathbf{u}}_T \end{bmatrix} = F(U, X_0, \tilde{X}_{1:T}) = \tilde{P}^{-\frac{1}{2}} \left(\begin{bmatrix} I & 0 \end{bmatrix} \tilde{P}^{-\frac{1}{2}} \right)^{\dagger} x_0, \quad (19)$$

where

$$\tilde{P} = \left(\begin{bmatrix} X_0 \\ \tilde{X}_{1:T} \end{bmatrix} \begin{bmatrix} X_0 \\ U \end{bmatrix}^{\dagger} \right)^{\mathsf{T}} Q \left(\begin{bmatrix} X_0 \\ \tilde{X}_{1:T} \end{bmatrix} \begin{bmatrix} X_0 \\ U \end{bmatrix}^{\dagger} \right) + \begin{bmatrix} 0 & 0 \\ 0 & R \end{bmatrix}.$$

Assumption IV.1 (Invertibility of $\tilde{P}$): We assume that matrix $\tilde{P}$ is almost surely invertible. □

Assumption IV.1 is required for the (almost sure) existence of the map (19). This is a technical assumption which is typically verified in practice. Following a procedure similar to [?] we let $\text{supp}(\Delta_X)$ denote the set of corrupted entries of $X_{1:T}$ and $\text{supp}(\Delta_X) = \{i : \delta_{X,i} \neq 0\}$, where $\delta_{X,i} = \text{vec}(\Delta_X)_i$ is the i-th entry of $\text{vec}(\Delta_X)$. Further, since $F(U, X_0, \tilde{X}_{1:T})$ is Fréchet-differentiable with respect to $X_{1:T}$ [41],[3] we can write it through its Taylor expansion

$$\begin{aligned} F(U, X_0, \tilde{X}_{1:T}) = {} & F(U, X_0, X_{1:T}) \\ & + \nabla F_X(U, X_0, X_{1:T}) \,\text{vec}(\Delta_X) \\ & + r(U, X_0, X_{1:T}, \Delta_X), \end{aligned} \quad (20)$$

with $\lim_{\|\Delta_X\|\to 0} \frac{\|r(U,X_0,X,\Delta_X)\|}{\|\Delta_X\|} = 0$ and where ∇F_X is the Jacobian matrix of $F(U, X_0, X_{1:T})$ with respect to $X_{1:T}$.

Finally, we let J and $\tilde{J}$ be the cost function of (2) obtained from x_0 through $\mathbf{u}_T$ and $\tilde{\mathbf{u}}_T$ in (11) and (19), respectively, and we let the error on the cost function be

$$\Delta J = |\tilde{J} - J|. \quad (21)$$

[3]The fact that F is differentiable in X can be proved by noticing that the partial derivatives of F with respect to each element of X exist (cf. (52)) and that they are continuous in an arbitrarily small neighborhood of X.

We are now ready to give probabilistic bounds with respect to ΔJ on control (11) for different assumptions on the noise Δ_X. In particular, given tolerance τ, and assuming that $r(\cdot) = 0$ in (20), we characterize the probability that ΔJ is greater than such tolerance. In this context, a smaller probability $\mathbb{P}[\Delta J \geq \tau]$ translates to a more robust controller, while a probability close to one implies a poor robustness performance with respect to the desired tolerance. In the following we let

$$V = \left(\begin{bmatrix} O_T & F_T \end{bmatrix} \right)^{\mathsf{T}} Q \left(\begin{bmatrix} O_T & F_T \end{bmatrix} \right) + \begin{bmatrix} 0 & 0 \\ 0 & R \end{bmatrix}, \quad (22)$$

and we refer the reader to Appendix G for a detailed derivation of the results that follow.

Theorem IV.2 (Probabilistic upper bound on ΔJ): Let ΔJ be as in (21). Then, for any $\tau > 0$,

$$\mathbb{P}[\Delta J \geq \tau] \leq \frac{1}{\sqrt{\tau}} \left(\sum_{i\in \text{supp}(\Delta_X)} c_{X,i}\, \mathbb{E}[|\delta_{X,i}|] \right), \quad (23)$$

where $c_{X,i} = \|V^{\frac{1}{2}} \nabla F_{X,i}\|$ and $\nabla F_{X,i}$ is the ith column of the Jacobian matrix of (19) with respect to $X_{1:T}$. □

Theorem IV.2 provides a simple upper bound on the accuracy of the data-driven control (11). In fact, bound (23) captures the sensitivity of (11) to perturbations of the dataset: the less sensitive the data-driven map is, i.e., the smaller $c_{X,i}$ is in absolute value, the closer the control input computed through the perturbed dataset in (19) is to the optimal input in (11) and, in turn, the smaller ΔJ is. The bound (23) requires that the noise distribution has a defined variance, but is typically loose. For instance, in the case of i.i.d. Gaussian perturbations $\delta_{X,i} \sim \mathcal{N}(0, \sigma^2)$, it holds that $\mathbb{E}[|\delta_{X,i}|] = \sigma\sqrt{2/\pi}$ and (23) reads as

$$\mathbb{P}[\Delta J \geq \tau] \leq \sigma \sqrt{\frac{2}{\tau\pi}} \left(\sum_{i\in \text{supp}(\Delta_X)} c_{X,i} \right), \quad (24)$$

which exhibits a square root decay in the ratio τ/σ^2. More informative bounds can be established for specific classes of perturbations, as we show next for the Gaussian case.

Theorem IV.3 (Probabilistic upper bound on ΔJ for Gaussian perturbations): Let ΔJ be as in (21) and assume that the entries of Δ_X are i.i.d. Gaussian random variables with zero mean and variance σ^2. Then, for any $\tau > 0$,

$$\mathbb{P}[\Delta J \geq \tau] \leq (n + mT + 1) \exp\left(-\frac{1}{2\sigma^2} \frac{\tau}{\sum_{i\in \text{supp}(\Delta_X)} c_{X,i}^2} \right) \quad (25)$$

where $c_{X,i} = \|V^{\frac{1}{2}} \nabla F_{X,i}\|$. □

Notice that the bound in Theorem IV.3 decays exponentially in the ratio τ/σ^2, and is therefore tighter than (24), as shown in the numerical simulations in Fig. 2.

We next investigate how the sensitivity of the data-driven map (as quantified by the norm of the Jacobian matrices $\nabla F_{X,i}$) is related to the number of experiments N. This study

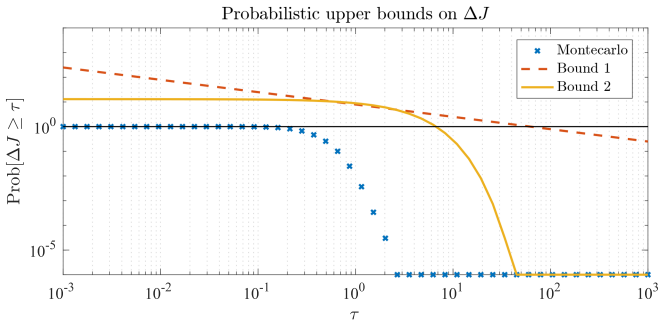

FIGURE 2. In this figure we plot the probabilistic bounds discussed in Section IV for a random (unstable) system with $n = 5$, $m = 2$ and compare them with a Montecarlo simulation (averaged over 10^6 simulations, with $T = 5$ and $N = n + mT = 12$). In particular, Bound 1 refers to (24) and Bound 2 to (25). The thin horizontal line is the implicit probability bound $\mathbb{P}[\cdot] \leq 1$. $\Delta_X \sim \mathcal{N}(0, \sigma^2 I)$, with $\sigma = 1$. Probability values are rounded up to 10^{-6}.

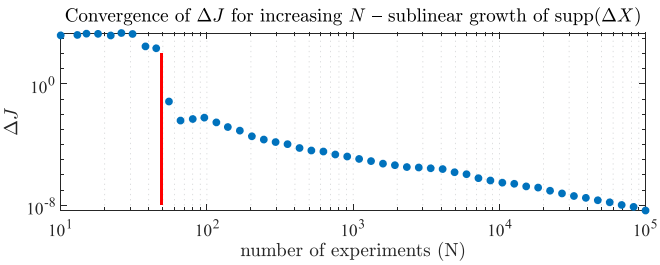

FIGURE 3. This figure shows the convergence of ΔJ in (21) computed with noisy data for increasing number of experiments N. We remark that $\bar{J}$ is the cost function (2) from the perturbed control map (19) computed when supp(Δ_X) grows sublinearly in N and $\delta_i \sim \mathcal{N}(1, 0)$. Equivalent results can be obtained for any distribution on supp(Δ_X) that satisfies the conditions on Theorem IV.5. Data are gathered from a randomly generated unstable system with $n = 5$, $m = 3$ and $T = 15$. Assumption II.1 is met for all N to the right of the red solid vertical line.

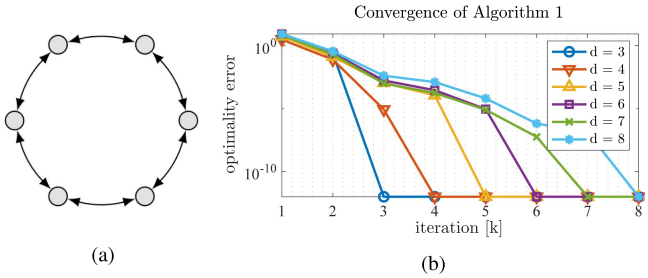

FIGURE 4. This figure shows the results associated with the experiment of Section V-A. In panel (a) we show the communication graph of an example network, with $M = 6$ and $d = 3$. All other networks in this example are similar in structure (ring network), with varying diameters $d \in \{3, \ldots, 8\}$. All experiments shown in panel (b) are performed with $T = 5$, $\epsilon = 10^{-3}$ and the tolerance for the pseudoinverse operation is set to $\texttt{tol} = 10^{-8}$.

is especially interesting in a scenario in which an attacker can affect some entries of $\tilde{X}_{1:T}$ deliberately, while supp(Δ_X) grows sublinearly with respect to N.

Lemma IV.4 (Asymptotic behavior of $\|\nabla F_{X,i}\|$ for large N): Assume that the entries of X_0, U are independent of N and $\sigma^2_{\min}([X_0^\mathsf{T} U^\mathsf{T}]^\mathsf{T}) \geq cN$ where $c > 0$ is a constant independent of N. Then, for all $i \in$ supp(Δ_X), $\|\nabla F_{X,i}\| \leq k_{X,i}/N$ where $k_{X,i} > 0$ is independent of N.[4] □

We remark that the condition $\sigma^2_{\min}([X_0^\mathsf{T} U^\mathsf{T}]^\mathsf{T}) \geq cN$ is typically satisfied for random i.i.d. initial conditions and inputs.[5] Thus, Lemma IV.4 shows that all $\|\nabla F_{X,i}\|$ typically converge to zero as the number of experiments N increases. Under this scenario, the map F becomes less sensible to corrupted data as more data becomes available. This conclusion is instrumental to prove the following result.

Theorem IV.5 (Asymptotically vanishing perturbation for large N): In addition to the assumption in Lemma IV.4, assume that the distributions of the entries of Δ_X are independent of N and $|\delta_{x,i}|$ have finite mean. Then, if supp(Δ_X) grows sublinearly with N, for any $\tau > 0$,

$$\lim_{N \to \infty} \mathbb{P}[\Delta J \geq \tau] = 0. \tag{26}$$

□

Under the assumptions of Theorem IV.5 we can guarantee that the error on ΔJ goes to zero for increasing N, regardless of Δ_X, as shown in Fig. 3, ensuring the robustness of the control action under attack on the data set.

V. NUMERICAL RESULTS AND ILLUSTRATIVE EXAMPLES

In this section we provide numerical validations of the results presented in this paper. First, we show how Algorithm 1 and Theorem III.4 can be used to solve the problem in (2) when the system (1) is unknown but data (8) are available. We further discuss how our formulas can be used directly, for a fixed-window control, in a receding horizon fashion. Finally, we compare and discuss our approach to alternative approaches in the literature.

A. AN APPLICATION TO SCALING RING NETWORKS

We assess the results of Section III by analyzing the convergence of Algorithm 1 for networks of varying diameter. We consider randomly generated ring networks, with communication graph $\mathcal{G}_c$ as in Fig. 4(a), here shown for a ring network with $M = 6$ and $d = 3$. In Fig. 4(b) we plot the error between the solution of (2) and the result of Algorithm 1. For the kth iteration of Algorithm 1, subnetwork i computes its own control inputs as $u_i[k] = U_i \beta_i[k]$, where $\beta_i[k]$ is the interim value of β_i at iteration k. The dynamics of (1) when $u_i[k]$ is injected to all $i \in \{1, \ldots, M\}$ is compared to the one induced by the model-based optimal control (11). As discussed in Theorem III.4, Algorithm 1 converges to the solution in a number of steps equal to the diameter of the communication graph $\mathcal{G}_c$.

[4] We say that a random variable X is *independent of* a deterministic parameter N if the distribution of X is not a function of N.

[5] If U and X_0 have i.i.d. zero mean entries $\mathbb{P}[\lim_{N\to\infty} [X_0^\top \; U^\top]^\top [X_0^\top \; U^\top] = NI] = 1$ by the (strong) Law of Large Numbers [42]. We remark that N is the number of experiments (columns of X_0, U) and I is the identity matrix.

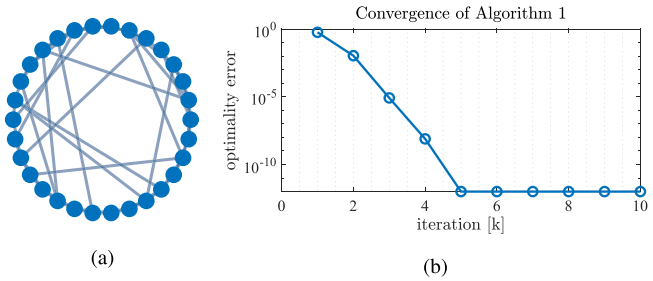

FIGURE 5. **This figure shows the results associated with the experiment of Section V-B. In panel (a) we show a randomly generated Watts-Strogats network with $M = 30$ agents. In panel (b) we show the convergence of Algorithm 1, comparing it to the model-based centralized solution of (2). Additional simulation parameters are $T = 5$, $\varepsilon = 10^{-3}$ and the tolerance for the pseudoinverse operation is set to $\texttt{tol} = 10^{-8}$.**

B. AN APPLICATION TO WATTS-STROGATZ NETWORKS

To prove the effectiveness of this approach in more complex network structures, we test Algorithm 1 on a Watts-Strogats network of larger size [43]. In Fig. 5 we run Algorithm 1 over a randomly generated Watts-Strogats network with $M = 30$, mean node degree 4 and rewiring probability of 0.15, see the documentation for the `WattsStrogatz` function in MATLAB [44]. The particular realization of Fig. 5(a) is a network with diameter 5, as testified by the convergence of Algorithm 1 shown in Fig. 5(b).

C. RECEDING HORIZON IMPLEMENTATION

Theorem III.1 assumes that T is the control horizon of problem (2) as well as of the dataset (8). It is possible to lift this requirement by implementing Algorithm 1 in a receding horizon fashion. That is, once Algorithm 1 is executed, each agent applies only a finite horizon $h \leq T$ of the computed controller $\mathbf{u}_{T,i} = U_i \beta_i$. After h time steps, then, Algorithm 1 is executed again, and a new controller is found for the subsequent horizon h. This approach is formally described in Algorithm 2. Clearly, there is no limit to the number of times that this algorithm can be run, i.e., we can use this approach to design an arbitrarily long controller. A receding horizon implementation is common throughout the literature, often termed Model Predictive Control (MPC), and is also implemented in related works [28], [30]. Comparison of these approaches with our method are discussed next.

D. COMPARISON WITH SPLITTING METHODS

Several strategies have been proposed to solve MPC problems in a distributed fashion. Among these, splitting methods [35], [?] have been explored in a model-based [45] and data-driven [28] setting. These consist in *splitting* one optimization problem in a family of smaller optimization problems. In the context of the problem setup (2), each agent needs to solve a local optimization problem, while iteratively exchanging information with other agents in order to properly converge to the global optimal solution. The accuracy of the approach is closely related to the number of iterations of the algorithm, a higher number of them leading to a more accurate solution. A known downside of these approaches is their slow convergence; typically, a large number of iterations is needed to converge to an acceptable solution (measured as its distance from the exact optimal solution). Recently [28] proposed splitting (2) trough a primal-dual flow. In Fig. 6 we compare the convergence properties of Algorithm 2 with a primal-dual algorithm based on [28]. In particular, we keep the problem formulation and data-collecting phase of [28], while modifying the cost function through the Augmented Lagrangian method, which is known to improve the convergence speed of the flow. As a standard practice, we distribute the Augmented Lagrangian through the Alternating Directions Method of Multipliers (ADMM), see [?, Chapter 8]. As highlighted in Fig. 6, Algorithm 2 returns a stabilizing controller even for higher noise values.

Algorithm 2: Receding Horizon Algorithm 1.

Input: T_{sim}, $\{X_1, \dots, X_M\}$, $\{U_1, \dots, U_M\}$, $\{X_{F,1}, \dots, X_{F,M}\}$, $\{X_{0,1}, \dots, X_{0,M}\}$, ε, Q, R, $\{x_{0,1}, \dots, x_{0,M}\}$, h

set $t = 0$

for T_{sim} *times* **do**

  run Algorithm 1, where each agent computes $\mathbf{u}_{T,i} = U_i \beta_i$

  let each agent apply h steps of the computed control, i.e., $u_i(t) = \mathbf{u}_{T,i}(1 : m_i)$

  set t to $t + 1$ and $x_{0,i}$ to the updated state $x_i(t + 1)$, for all $i = \{1, \dots, M\}$

end

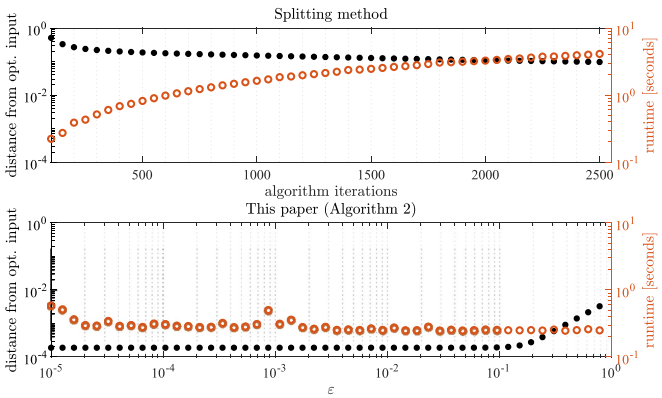

FIGURE 6. **This figure shows a comparison between the approach in Algorithm 2 and a distributed data-driven solver based on [28] (splitting method). The experiments are performed with noiseless data from the same system, a randomly generated network with $M = 3$ agents, and diverse state size. This comparison highlights the *computational time to accuracy tradeoff* that the approaches have with respect to a particular design parameter, namely the number of iterations of the splitting method, and the parameter ε for Algorithm 2. The runtime is reported in seconds (lower is better), and the distance from the optimal input is computed as the norm of the difference of the solution and the exact optimal control computed through a model-based LQR solver (lower is better). Additional simulation parameters are $T = 5$, $\texttt{tol} = 10^{-8}$.**

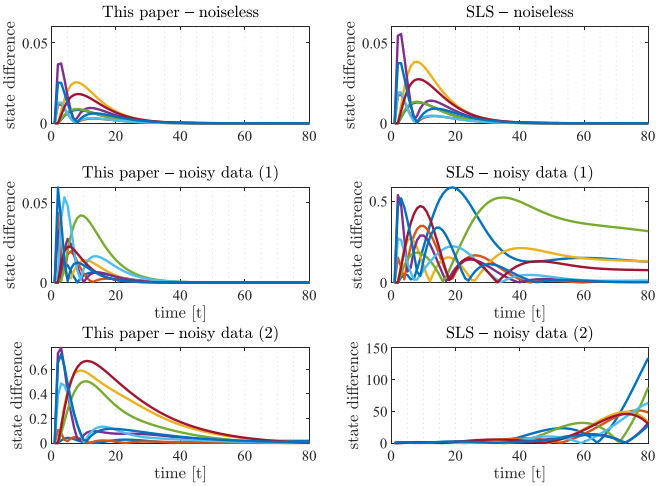

FIGURE 7. This figure compares the robustness of two different distributed data driven controllers when the recorded data are affected by noise. The underlying model is a chain of four interconnected bi-dimensional oscillators, as discussed in [30, Sec. VI]. Problem (2) is considered with $Q = 1$, $R = 1$, and random initial condition. The top panel shows the difference between the reference state evolution and the solution of Algorithm 2 (left) and of [30] (right) when data are collected without noise. As expected, both methods converge to the equilibrium (the planning horizon for both methods is set to $T = 10$). In the middle panel the same comparison is shown when data are collected with noise (each input and state trajectory is perturbed with an additive i.i.d. disturbance with zero mean and variance $\sigma^2 = 0.1$). Finally, in the bottom panel, the same simulation is when data are collected with variance $\sigma^2 = 5$. Additional parameters are $\varepsilon = 0.01$ and $\texttt{tol} = 10^{-8}$.

E. COMPARISON WITH [30] WITH NOISY DATA

We now compare the performance of our approach to [30], when data are collected with noise. We consider the same system used in [30, Sec. VI], a ring network with $M = 4$ subsystems, i.e., $\mathcal{E}_c = \{(i, i+1), (i+1, i), i = 1, \dots, M\} \cup \{(1, M), (M, 1)\}$. Each subsystem (7) is a two-dimensional linearized and discretized ($\Delta t = 0.2$) swing dynamics, with

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

and $B_{ii} = [0 \quad 1]^\mathsf{T}$. The design parameters are drawn randomly from continuous uniform distributions $m_i \sim \mathcal{U}[0, 2]$, $d_i \sim \mathcal{U}[0.5, 1]$, and $k_{ij} \sim \mathcal{U}[1, 1.5]$, with $k_i = \sum_{j \in \mathcal{N}_i} k_{ij}$. We run the same experiment three times with $Q = I$ and $R = I$, the first assuming that $\Delta_X = 0$ (i.e., noiseless case), the second that $\Delta_X \sim \mathcal{N}(0, \sigma^2 I)$, with $\sigma^2 = 0.1$, and finally with the same noise distribution but with $\sigma^2 = 5.0$. We run our algorithm with the receding horizon implementation described in Algorithm 2, with $h = 1$. We notice from Fig. 7 that our method and that of [30] perform well when $\Delta_X = 0$. Crucially, however, the convergence of [30] to a stabilizing controller is significantly slower than Algorithm 2, when data are corrupted by noise.

VI. CONCLUSION

In this paper we propose an algorithm to distributedly learn optimal controllers for an unknown network system through data. Our approach provably converges to a suboptimal solution in a finite number of steps, with a subopotimality gap that can be characterized as a function of the available data. Moreover, although data are distributed among multiple agents in the network, and communication between agents is therefore necessary to find a globally optimal control, this approach does not require to directly share trajectory data between agents. We discuss how these features are attractive, especially when compared to alternative approaches in the literature. Finally, we characterize the robustness properties of our approach and show that we can bound, in probability, the error on the cost function of the control computed with corrupted data.

APPENDIX

A. DATA-DRIVEN TRAJECTORIES OF (1)

We start by recalling a structural Lemma, which appeared for input-output trajectories in [39], and is here adapted for input-state trajectories.

Lemma A.1 (Data-driven trajectories of (1)): Let (3) be the data generated by the system (1) with $T \geq n$, and let $\bar{\mathbf{x}}_T$ be the state trajectory of (1) generated with some initial condition and control input. Then,

$$\bar{\mathbf{x}}_T = \begin{bmatrix} X K_U & X K_0 \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \end{bmatrix}, \tag{27}$$

for some vectors α and β. □

Proof: Let $\bar{x}_0$ and $\bar{\mathbf{u}}_T$ be the initial condition and input to (1). Since the matrices $X_0 K_U$ and $U K_0$ are full-row rank (see Assumption II.1), there exists α and β such that

$$\bar{x}_0 = X_0 K_U \alpha \quad \text{and} \quad \bar{\mathbf{u}}_T = U K_0 \beta. \tag{28}$$

From (4) we have

$$\begin{aligned} \bar{\mathbf{x}}_T &= O_T \bar{x}_0 + F_T \bar{\mathbf{u}}_T = O_T X_0 K_U \alpha + F_T U K_0 \beta \\ &= X K_U \alpha + X K_0 \beta, \end{aligned}$$

where the last equality follows from (5). ■

Lemma A.1 shows how any trajectory of (1) can be written as a linear combination of the available data. In particular, state trajectories are obtained in (27) as the sum of the free ($X K_U \alpha$) and forced responses ($X K_0 \beta$), which are reconstructed from data of arbitrary control experiments.

B. PROOF OF THEOREM III.1

Consider the following problem

$$\begin{aligned} &\arg\min_{\gamma} \left\| P^{\frac{1}{2}} \gamma \right\|^2 \\ &\text{subject to } [I \quad 0]\, \gamma = x_0. \end{aligned} \tag{29}$$

where P is as in (10). Note that, the solution to the above problem is $\gamma^* = \text{col}(x_0, \mathbf{u}_T^*)$ with $\mathbf{u}_T^*$ being the solution to (2).

This follows from (4) and the fact that

$$P = \left(X\begin{bmatrix}X_0\\U\end{bmatrix}^{\dagger}\right)^{\mathsf{T}} Q \left(X\begin{bmatrix}X_0\\U\end{bmatrix}^{\dagger}\right) + \begin{bmatrix}0 & 0\\0 & R\end{bmatrix}$$
$$= \begin{bmatrix}O_T & F_T\end{bmatrix}^{\mathsf{T}} Q \begin{bmatrix}O_T & F_T\end{bmatrix} + \begin{bmatrix}0 & 0\\0 & R\end{bmatrix} \tag{30}$$

when Assumption II.1 holds. If P is invertible, letting $v = P^{1/2}\gamma$ problem (29) becomes

$$\begin{aligned}\arg\min_{v} \quad & \|v\|^2\\ \text{subject to} \quad & \begin{bmatrix}I & 0\end{bmatrix}P^{-1/2}v = x_0.\end{aligned} \tag{31}$$

whose solution is $v_T^* = (\begin{bmatrix}I & 0\end{bmatrix}P^{-1/2})^{\dagger}x_0$. Thus, the solution to (29) is

$$\gamma^* = \begin{bmatrix}x_0\\ \mathbf{u}_T^*\end{bmatrix} = P^{-1/2}v_T^* = P^{-1/2}(\begin{bmatrix}I & 0\end{bmatrix}P^{-1/2})^{\dagger}x_0. \tag{32}$$

We show that if $Q^{1/2}O_T$ has full column rank, then $P \succ 0$, which in turn implies that P is invertible. To this end, notice that, from (30),

$$P = \underbrace{\begin{bmatrix}O_T^{\mathsf{T}}QO_T & O_T^{\mathsf{T}}QF_T\\ F_T^{\mathsf{T}}QO_T & F_T^{\mathsf{T}}QF_T\end{bmatrix}}_{P_1} + \begin{bmatrix}0 & 0\\0 & R\end{bmatrix}.$$

Since $O_T^{\mathsf{T}}QO_T \succ 0$ (which directly follows from the assumption that $Q^{1/2}O_T$ has full column rank) and $P_1 \succeq 0$, the Schur complement of the block $O_T^{\mathsf{T}}QO_T$ of P_1 must be positive semidefinite [?, Theorem 1.12], namely $S = F_T^{\mathsf{T}}QF_T - F_T^{\mathsf{T}}QO_T^{\mathsf{T}}(O_T^{\mathsf{T}}QO_T)^{-1}O_T^{\mathsf{T}}QF_T \succeq 0$. Next, since $S \succeq 0$ and $R \succ 0$, the Schur complement of the block $O_T^{\mathsf{T}}QO_T$ of P, namely $S + R$, is positive definite. Since the block $O_T^{\mathsf{T}}QO_T$ of P and its Schur complement are both positive definite, follows that $P \succ 0$ [?, Theorem 1.12]. □

We complement the proof by noticing that, for P not invertible, and for any $Q \succ 0$, problem (29) becomes

$$\begin{aligned}\arg\min_{v_T, w} \quad & \|v_T\|^2\\ \text{subject to} \quad & \begin{bmatrix}I & 0\end{bmatrix}((P^{1/2})^{\dagger}v_T + K_P w) = x_0,\end{aligned} \tag{33}$$

where w is a vector of appropriate size, from which a solution alternative to (32) can be found.

C. PROOF OF LEMMA III.2

From Lemma A.1 we can write

$$\mathbf{x}_T = \begin{bmatrix}XK_U & XK_0\end{bmatrix}\begin{bmatrix}\bar{\alpha}\\ \bar{\beta}\end{bmatrix},$$

with $x_0 = X_0K_U\bar{\alpha}$ and $\mathbf{u}_T = UK_0\bar{\beta}$. Equivalently, we can write

$$\mathbf{x}_T = \begin{bmatrix}X & X\end{bmatrix}\begin{bmatrix}\alpha\\ \beta\end{bmatrix}$$

where $\alpha = K_U\bar{\alpha}$ (i.e., $U\alpha = 0$), and $\beta = K_0\bar{\beta}$ (i.e., $X_0\beta = 0$), and consequently $x_0 = X_0\alpha$, $\mathbf{u}_T = U\beta$ and the cost function of (12) equals that of (2). When $\mathbf{u}_T^*$ is the solution to (2) and α^*, β^* is the solution to (12), it is then straightforward to see that $\mathbf{u}_T^* = U\beta^*$.

We can further derive an explicit form of the optimal vectors α^* and β^* instrumental to other results in the paper. By substituting the constraints of (12) into the cost function, and for H, Y, and $\bar{x}_0$ as in (17), problem (12) can be written as

$$\arg\min_{w} \left\|H\left(Y^{\dagger}\bar{x}_0 + K_Y w\right)\right\|^2, \tag{34}$$

where $K_Y = \text{Basis}(\text{Ker}(Y))$. The minimizers to (34) define the set of optimal vectors α^*, β^* via

$$\begin{bmatrix}\alpha^*\\ \beta^*\end{bmatrix} = Y^{\dagger}\bar{x}_0 + K_Y w^*$$
$$= \left(I - K_Y(HK_Y)^{\dagger}H\right)Y^{\dagger}\bar{x}_0 + r \tag{35}$$

where $r \in \text{Ker}(H) \subseteq \text{Ker}(U)$. The resulting, unique, optimal control $\mathbf{u}_T = U\beta^*$ is the solution to (2). □

D. PROOF OF LEMMA III.3

From the constraints in (13) it holds, for all $\varepsilon > 0$,

$$v = -\frac{1}{\varepsilon}Q^{\frac{1}{2}}X(\alpha + \beta), \quad w = -\frac{1}{\varepsilon}R^{\frac{1}{2}}U\beta.$$

Substituting these equations in the cost function, and for H, Y and $\bar{x}_0$ as in (17), problem (13) can be rewritten as

$$\gamma^*(\varepsilon) = \begin{bmatrix}\alpha^*(\varepsilon)\\ \beta^*(\varepsilon)\end{bmatrix} = \arg\min_{\alpha,\beta} \quad \varepsilon^2\left\|\begin{bmatrix}\alpha\\ \beta\end{bmatrix}\right\|^2 + \left\|H\begin{bmatrix}\alpha\\ \beta\end{bmatrix}\right\|^2$$
$$\text{s.t.} \quad Y\begin{bmatrix}\alpha\\ \beta\end{bmatrix} = \bar{x}_0. \tag{36}$$

If $\gamma^*(\varepsilon)$ is bounded as $\varepsilon \to 0^+$, then $\gamma^*(\varepsilon)$ converges to the (minimum norm) solution to problem (12) as $\varepsilon \to 0^+$, and (14) holds. Thus, it remains to prove the boundedness of $\gamma^*(\varepsilon)$ as $\varepsilon \to 0^+$. To this end, we note that by writing $\gamma^*(\varepsilon) = \gamma_1(\varepsilon) + \gamma_2(\varepsilon)$ with

$$\gamma_1(\varepsilon) \perp \text{Ker}\, H \cap \text{Ker}\, Y \text{ and } \gamma_2(\varepsilon) \in \text{Ker}\, H \cap \text{Ker}\, Y,$$

the cost of (36) evaluated at $\gamma^*(\varepsilon)$ reads as

$$C(\gamma^*(\varepsilon)) = \varepsilon^2\|\gamma_1(\varepsilon)\|^2 + \varepsilon^2\|\gamma_2(\varepsilon)\|^2 + \|H\gamma_1(\varepsilon)\|^2, \tag{37}$$

since $\gamma_2(\varepsilon) \in \text{Ker}\, H$. Further, the vector $\gamma_1(\varepsilon)$ satisfies $Y\gamma_1(\varepsilon) = Y\gamma^*(\varepsilon) = \bar{x}_0$ since $\gamma_2(\varepsilon) \in \text{Ker}\, Y$. From the latter fact and (37), it follows that it must be $\gamma_2(\varepsilon) = 0$, $\forall\varepsilon > 0$, for $\gamma^*(\varepsilon)$ to be optimal. Further, from (37) and the fact that there always exists a γ (e.g., $\gamma = Y^{\dagger}\bar{x}_0$) which is independent of ε and satisfies the constraint in (36) (thus yielding a cost which is bounded $\forall\varepsilon > 0$), it follows that $\gamma_1(\varepsilon) = \gamma^*(\varepsilon)$ must be bounded as $\varepsilon \to 0^+$. □

E. PROOF OF THEOREM III.4

The proof follows an argument similar to the one of [46, Theorem 3.3]. Let γ_i be the estimate of agent i, W_i be defined as in (15), and $K_i = \text{Basis}(\text{Ker}(W_i))$. Observe that $\gamma_i = W_i^\dagger \,\text{col}(x_{i,0}, 0, 0, 0, 0) \perp \text{Ker}(W_i)$. Let i and j be two neighboring agents, i.e, $(i, j) \in \mathcal{E}_c$, then there exist two vectors κ_i and κ_j such that $\gamma_i + K_i\kappa_i = \gamma_j + K_j\kappa_j$. In particular, such vectors can be chosen as

$$\begin{bmatrix}\kappa_i \\ \kappa_j\end{bmatrix} = \begin{bmatrix}-K_i & K_j\end{bmatrix}^\dagger (\gamma_i - \gamma_j).$$

Substituting κ_i back in γ_i we have that the vector

$$\gamma_i^+ = \gamma_i + \begin{bmatrix}K_i & 0\end{bmatrix}\begin{bmatrix}-K_i & K_j\end{bmatrix}^\dagger(\gamma_i - \gamma_j)$$

is such that $\text{col}(x_{0,i}, 0, 0, 0, 0) = W_i\gamma_i^+$ and $\text{col}(x_{0,j}, 0, 0, 0, 0) = W_j\gamma_i^+$. Moreover, we have that $\gamma_i^+ \perp (\text{Im}(K_i) \cap \text{Im}(K_j))$, since

$$\begin{bmatrix}\kappa_i \\ \kappa_j\end{bmatrix} \perp \text{Ker}\left(\begin{bmatrix}-K_i & K_j\end{bmatrix}\right).$$

We notice that $K_i\kappa_i \perp \text{Im}(K_j)$; by contradiction, if $K_i\kappa_i \not\perp \text{Im}(K_j)$, then one can find $\kappa_i = \tilde{\kappa}_i + \bar{\kappa}_i$, where $K_i\tilde{\kappa}_i \perp \text{Im}(K_j)$ and $K_i\bar{\kappa}_i \in \text{Im}(K_j)$. Let $\bar{\kappa}_j = K_j^\dagger K_i\bar{\kappa}_i$ and $\tilde{\kappa}_j = \kappa_j - \bar{\kappa}_j$. Then $\text{col}(\bar{\kappa}_i, \bar{\kappa}_j) \in \text{Ker}([-K_i \;\; K_j])$ and hence $\text{col}(\bar{\kappa}_i, \bar{\kappa}_j) \not\perp \text{Ker}([-K_i \;\; K_j])$, which contradicts the hypothesis. We conclude that $[K_i \;\; 0][-K_i \;\; K_j]^\dagger(\gamma_i - \gamma_j) \perp \text{Im}(K_j)$, and, since $\gamma_i \perp \text{Im}(K_i)$, we can conclude that $\gamma_i^+ \perp (\text{Im}(K_i) \cap \text{Im}(K_j))$. The theorem follows by noticing that after a number of steps equal to the diameter of $\mathcal{G}_c$, each vector γ_i verifies all the measurements, since we will have that $\gamma_i \perp \bigcap_j^M \text{Im}(K_j)$. Finally, we remark that the solution we are interested in involves only the second N elements of γ_i, corresponding to β_i. □

F. PROOF OF LEMMA III.5

For H, Y and $\bar{x}_0$ as in (17), problem (13) can be rewritten as

$$\begin{bmatrix}\alpha^*(\varepsilon) \\ \beta^*(\varepsilon)\end{bmatrix} = \arg\min_{\alpha,\beta} \left\|\begin{bmatrix}\varepsilon I \\ H\end{bmatrix}\begin{bmatrix}\alpha \\ \beta\end{bmatrix}\right\|^2 \quad \text{s.t.} \quad Y\begin{bmatrix}\alpha \\ \beta\end{bmatrix} = \bar{x}_0. \tag{38}$$

The solution to (38) is $Y^\dagger\bar{x}_0 - K_Y w^*(\varepsilon)$ with

$$\begin{aligned} w^*(\varepsilon) &= \arg\min_w \left\|\begin{bmatrix}\varepsilon I \\ H\end{bmatrix}(Y^\dagger\bar{x}_0 - K_Y w)\right\|^2 \\ &= \left(\begin{bmatrix}\varepsilon I \\ H\end{bmatrix}K_Y\right)^\dagger\begin{bmatrix}\varepsilon I \\ H\end{bmatrix}Y^\dagger\bar{x}_0 \\ &= \left(K_Y^\top(\varepsilon^2 I + H^\top H)K_Y\right)^\dagger K_Y^\top(\varepsilon^2 I + H^\top H)Y^\dagger\bar{x}_0 \\ &= \left(K_Y^\top(\varepsilon^2 I + H^\top H)K_Y\right)^\dagger K_Y^\top H^\top H Y^\dagger\bar{x}_0, \end{aligned} \tag{39}$$

where in the last-but-one step we used that $A^\dagger = (A^\top A)^\dagger A^\top$ for any matrix A, and in the last step $K_Y^\top Y^\dagger = 0$. Similarly, we can rewrite any minimizer to Problem (12) as (cf. (35))

$$\begin{aligned} \begin{bmatrix}\alpha^* \\ \beta^*\end{bmatrix} &= \left(I - K_Y\left(HK_Y\right)^\dagger H\right)Y^\dagger\bar{x}_0 + r \\ &= \left(I - K_Y\left(K_Y^\top(H^\top H)K_Y\right)^\dagger K_Y^\top H^\top H\right)Y^\dagger\bar{x}_0 + r, \end{aligned} \tag{40}$$

where $r \in \text{Ker}(H) \subseteq \text{Ker}(U)$ and $r \in \text{Ker}(H) \subseteq \text{Ker}(X_0)$. From (39) and (40) and from the fact that $\|\text{col}(\alpha, \beta)\| \geq \|\beta\|$ for any α and β, it follows that $\|U(\beta^* - \beta^*(\varepsilon))\| = \|\mathbf{u}_T^* - U\beta^*(\varepsilon)\|$ is bounded as in (18). □

G. CHARACTERIZING BOUNDS FOR ΔJ.

Since $F(U, X_0, \tilde{X}_{1:T})$ is Fréchet-differentiable with respect to $X_{1:T}$ [41], we rewrite F as in (20). We write (20) in compact form as

$$\tilde{F} = F + \nabla F_X \,\text{vec}(\Delta_X) + r. \tag{41}$$

We then define $J = F^\mathsf{T} V F = (V^{\frac{1}{2}}F)^\mathsf{T}(V^{\frac{1}{2}}F) = G^\mathsf{T}G$ and $\tilde{J} = \tilde{F}^\mathsf{T} V \tilde{F} = \tilde{G}^\mathsf{T}\tilde{G}$, where $\tilde{G} := V^{\frac{1}{2}}\tilde{F}$ and

$$V = \begin{bmatrix}O_T & F_T\end{bmatrix}^\mathsf{T} Q \begin{bmatrix}O_T & F_T\end{bmatrix} + \begin{bmatrix}0 & 0 \\ 0 & R\end{bmatrix}$$

For technical reasons, we let

$$\Delta G = \left\|\tilde{G} - G\right\| \tag{42}$$

be a proxy for the optimality error ΔJ (21). In particular, in the following we study bounds on ΔG which we can then easily express as a function of ΔJ. To do this we need the following lemma.

Lemma G.1 (Relating ΔG and ΔJ): It holds that

$$\Delta J = \Delta G^2 \quad \text{almost surely (a.s.).} \tag{43}$$

Proof: We recall that

$$\Delta J = \left\|\tilde{J} - J\right\| = |\tilde{G}^\mathsf{T}\tilde{G} - G^\mathsf{T}G|$$

and we notice that

$$\Delta G^2 = \left\|\tilde{G} - G\right\|^2 = |\tilde{G}^\mathsf{T}\tilde{G} - G^\mathsf{T}G - 2(\tilde{G} - G)^\mathsf{T}G|.$$

Then, condition (43) holds when $G^\mathsf{T}G = \tilde{G}^\mathsf{T}G$, that is

$$\tilde{F}^\mathsf{T} V F = F^\mathsf{T} V F. \tag{44}$$

The above can be shown by noticing that

$$\begin{aligned} VF &= VP^{-\frac{1}{2}}\left([I \;\; 0]P^{-\frac{1}{2}}\right)^\dagger \\ &= VP^{-1}\begin{bmatrix}I \\ 0\end{bmatrix}\left([I \;\; 0]P^{-1}\begin{bmatrix}I \\ 0\end{bmatrix}\right)^{-1} = \begin{bmatrix}\left([P^{-1}]_{1:n,1:n}\right)^{-1} \\ 0\end{bmatrix}, \end{aligned}$$

where we used the facts that $A^\dagger = A^\top(AA^\top)^{-1}$ when $A \in \mathbb{R}^{n\times m}$ is full-row rank, and $VP^{-1} = I$ when Assumption II.1

holds. Through a similar argument, in view of Assumption IV.1, we notice that

$$\tilde{F} = \tilde{P}^{-\frac{1}{2}} \left([I \quad 0] \tilde{P}^{-\frac{1}{2}} \right)^{\dagger} = [I] \text{ a.s.},$$

and

$$F = P^{-\frac{1}{2}} \left([I \quad 0] P^{-\frac{1}{2}} \right)^{\dagger} = [I].$$

We can therefore conclude that (44) holds and, in particular, $\tilde{F}^{\mathsf{T}} V F = F^{\mathsf{T}} V F = ([P^{-1}]_{1:n,1:n})^{-1}$ almost surely. ■

Lemma G.2 (First order approximation of $\tilde{G}$): Let $\nabla F_{X,i}$ denote the ith column of ∇F_X in (41) and let

$$\Sigma = \left\| \tilde{G} - G - \sum_{i \in \text{supp}(\Delta_X)} \delta_{X,i} V^{\frac{1}{2}} \nabla F_{X,i} \right\|.$$

Then, for any $\tau > 0$,

$$\lim_{\mathbb{E}[\|\text{vec}(\Delta_X)\|] \to 0} \mathbb{P}\left[\Sigma \geq \tau \sqrt{\mathbb{E}[\|\text{vec}(\Delta_X)\|]} \right] = 0. \tag{45}$$

□

Proof: From (41) let $\tilde{F} = F + \sum_{i=1}^{d} \delta_i \nabla F_{X,i} + r$. Multiplying both sides by $V^{\frac{1}{2}}$ we obtain $V^{\frac{1}{2}} r = \tilde{G} - G - \sum_1^d \delta_i V^{\frac{1}{2}} \nabla F_{X,i}$. Noting that

$$\mathbb{P}\left[\left\| V^{\frac{1}{2}} r \right\| \geq \tau \sqrt{\mathbb{E}[\|\text{vec}(\Delta_X)\|]} \right] \to 0$$

as $\mathbb{E}[\|\text{vec}^{-1}(\Delta_X)\|] \to 0$ the proof follows verbatim from [41, Theorem 3.1.1]. ■

Lemma G.2 confirms that when the expected value of the perturbation on $X_{1:T}$ is small so is the residual r of the expansion (41). In the following we let

$$\tilde{G} = G + \sum_{i \in \text{supp}(\Delta_X)} \delta_{X,i} V^{\frac{1}{2}} \nabla F_{X,i}, \tag{46}$$

that is, we assume $\|V^{\frac{1}{2}} r\| = 0$. We highlight that vector $\nabla F_{X,i}$, consisting of the partial derivatives of the map F with respect to the entries of $X_{1:T}$, captures the sensitivity of the data-driven control inputs to perturbations of $X_{1:T}$.

The results that follow are given as a function of ΔG but can easily be given as a function of ΔJ by noticing that $\mathbb{P}[x^2 > \tau] = \mathbb{P}[x > \sqrt{\tau}]$, when $x > 0$.

H. PROOF OF THEOREM IV.2

From (46) we write ΔG as,

$$\Delta G = \left\| \tilde{G} - G \right\| = \left\| \sum_{i \in \text{supp}(\Delta_X)} \delta_{X,i} V^{\frac{1}{2}} \nabla F_{X,i} \right\|,$$

$$\leq \sum_{i \in \text{supp}(\Delta_X)} |\delta_{X,i}| \left\| V^{\frac{1}{2}} \nabla F_{X,i} \right\| = \overline{\Delta G}. \tag{47}$$

Hence, for any $\tau > 0$, the set inclusion $\{\Delta G \geq \tau\} \subseteq \{\overline{\Delta G} \geq \tau\}$ holds, which implies $\mathbb{P}[\Delta G \geq \tau] \leq \mathbb{P}[\overline{\Delta G} \geq \tau]$, by the monotonicity of probability measures. Note that ΔG is a non-negative random variable. Thus, by Markov's inequality [42] and the linearity of the expected value, for any $\tau > 0$,

$$\mathbb{P}\left[\overline{\Delta G} \geq \sqrt{\tau}\right] \leq \frac{1}{\sqrt{\tau}} \left(\sum_{i \in \text{supp}(\Delta_X)} \left\| V^{\frac{1}{2}} \nabla F_{X,i} \right\| \mathbb{E}[|\delta_{X,i}|] \right) \tag{48}$$

which yields (23) following $\mathbb{P}[\Delta J > \tau] = \mathbb{P}[\Delta G > \sqrt{\tau}]$. □

I. PROOF OF THEOREM IV.3

From the tail bound on rectangular matrix Gaussian series in [47, Theorem 1.5], it holds

$$\mathbb{P}\left[\Delta G \geq \sqrt{\tau}\right] = \mathbb{P}\left[\left\| \sum_{i \in \text{supp}(\Delta_X)} \delta_{X,i} V^{\frac{1}{2}} \nabla F_{X,i} \right\| \geq \sqrt{\tau} \right]$$

$$\leq (n + mT + 1) \exp\left(-\frac{\tau}{2\bar{\sigma}^2} \right) \tag{49}$$

where

$$\bar{\sigma}^2 := \sigma^2 \max \left\{ \left\| \sum_{i \in \text{supp}(\Delta_X)} (V^{\frac{1}{2}} \nabla F_{X,i})(V^{\frac{1}{2}} \nabla F_{X,i})^{\mathsf{T}} \right\|, \right.$$

$$\left. \left\| \sum_{i \in \text{supp}(\Delta_X)} (V^{\frac{1}{2}} \nabla F_{X,i})^{\mathsf{T}} (V^{\frac{1}{2}} \nabla F_{X,i}) \right\| \right\}.$$

Equation (25) follows from (49) and the bound $\bar{\sigma}^2 \leq \sigma^2 \sum_{i \in \text{supp}(\Delta_X)} \|V^{\frac{1}{2}} \nabla F_{X,i}\|^2$. □

J. PROOF OF LEMMA IV.4

Let $\text{vec}(\Delta_X)$ be the set of corrupted entries of $X_{1:T}$, with $x_i := \text{vec}(\Delta_X)_i$ and rewrite (11) as

$$F(U, X_0, X_{1:T}) = P^{-\frac{1}{2}} \left([I \quad 0] P^{-\frac{1}{2}} \right)^{\dagger} x_0$$

$$= P^{-1} \begin{bmatrix} I \\ 0 \end{bmatrix} \left([I \quad 0] P^{-1} [I 0] \right)^{-1} x_0. \tag{50}$$

It holds that (51)–(53) shown at the bottom of the next page. Notice that

$$P = \left(\begin{bmatrix} X_0 \\ X_{1:T} \end{bmatrix} \begin{bmatrix} X_0 \\ U \end{bmatrix}^{\dagger} \right)^{\mathsf{T}} Q \left(\begin{bmatrix} X_0 \\ X_{1:T} \end{bmatrix} \begin{bmatrix} X_0 \\ U \end{bmatrix}^{\dagger} \right) + \begin{bmatrix} 0 & 0 \\ 0 & R \end{bmatrix}$$

$$= \begin{bmatrix} O_T & F_T \end{bmatrix}^{\mathsf{T}} Q \begin{bmatrix} O_T & F_T \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & R \end{bmatrix}$$

when Assumption II.1 holds. Moreover $\nabla F_{X,i}$ can be written as in (52), where Γ_i is a $(n-1)T \times N$ matrix with one entry (corresponding to the perturbed element x_i) equal to one and zeros otherwise, and where we used that $\frac{\partial P^{-1}}{\partial x_i} = P^{-1} \frac{\partial P}{\partial x_i} P^{-1}$ (e.g., see [48]). From (52),

$$\|\nabla F_{X,i}^{(1)}\| \leq \ell_{X,i}^{(1)} \left\| \Gamma_i \begin{bmatrix} X_0 \\ U \end{bmatrix}^{\dagger} \right\|$$

$$\leq \ell_{X,i}^{(1)} \left\| \Gamma_i \begin{bmatrix} X_0 \\ U \end{bmatrix}^{\mathsf{T}} \left(\begin{bmatrix} X_0 \\ U \end{bmatrix} \begin{bmatrix} X_0 \\ U \end{bmatrix}^{\mathsf{T}} \right)^{-1} \right\|$$

$$\leq \ell_{X,i}^{(1)} \left\| \Gamma_i \begin{bmatrix} X_0 \\ U \end{bmatrix}^{\mathsf{T}} \right\| \sigma_{\min}^{-2} \left(\begin{bmatrix} X_0 \\ U \end{bmatrix} \right)$$

$$\leq \ell_{X,i}^{(1)} \left\| \Gamma_i \begin{bmatrix} X_0 \\ U \end{bmatrix}^{\mathsf{T}} \right\| \frac{1}{cN} \tag{54}$$

where

$$\ell_{X,1}^{(1)} = 2\|P^{-1}\|^2 \|Q\| \left\| \begin{bmatrix} O_T & F_T \end{bmatrix} \right\| \cdot \cdot \left\| \begin{bmatrix} I \\ 0 \end{bmatrix} \left([I \quad 0] P^{-1} \begin{bmatrix} I \\ 0 \end{bmatrix} \right)^{-1} x_0 \right\| \tag{55}$$

does not depend on N. In the first step of (54) we used the submultiplicativity of matrix 2-norm, in the second step the fact that $A^\dagger = A^{\mathsf{T}}(AA^{\mathsf{T}})^{-1}$ when A is full-row rank, the third step from $\|A\| \geq \sigma_{\min}(A)$, and the fourth step follows by the assumption on $\sigma_{\min}^2([X_0^{\mathsf{T}} U^{\mathsf{T}}]^{\mathsf{T}}) \geq cN$. Finally, since the matrix $\Gamma_i \begin{bmatrix} X_0^{\mathsf{T}} & U^{\mathsf{T}} \end{bmatrix}$ has only one row different from zero and the entries of such row are independent of N by assumption, (54) implies that $\|\nabla F_{X,i}^{(1)}\| \leq k_{X,i}^{(1)}/N$, where $k_{X,i}^{(1)} > 0$ does not depend on N. Next, notice that we can write $\nabla F_{X,i}^{(2)}$ as in (52) and, similarly as above,

$$\|\nabla F_{X,i}^{(2)}\| \leq \ell_{X,i}^{(2)} \left\| \Gamma_i \begin{bmatrix} X_0 \\ U \end{bmatrix}^{\dagger} \right\| \leq \ell_{X,i}^{(2)} \left\| \Gamma_i \begin{bmatrix} X_0 \\ U \end{bmatrix}^{\mathsf{T}} \right\| \frac{1}{cN} \tag{56}$$

where

$$\ell_{X,1}^{(2)} = \|Q\| \, \|[O_T \quad F_T]\| \left\| \left([I \quad 0] P^{-1} \begin{bmatrix} I \\ 0 \end{bmatrix} \right)^{-1} \right\|^2 \|x_0\| \tag{57}$$

$$\nabla F_{X,i} = \frac{\partial F(U, X_0, X_{1:T})}{\partial x_i} = \underbrace{\frac{\partial P^{-1}}{\partial x_i} \begin{bmatrix} I \\ 0 \end{bmatrix} \left([I \quad 0] P^{-1} \begin{bmatrix} I \\ 0 \end{bmatrix} \right)^{-1} x_0}_{=\nabla F_{X,i}^{(1)}}$$

$$+ \underbrace{P^{-1} \begin{bmatrix} I \\ 0 \end{bmatrix} \frac{\partial \left([I \quad 0] P^{-1} \begin{bmatrix} I \\ 0 \end{bmatrix} \right)^{-1}}{\partial x_i} x_0}_{=\nabla F_{X,i}^{(2)}} . \tag{51}$$

$$\nabla F_{X,i}^{(1)} = \frac{\partial P^{-1}}{\partial x_i} \begin{bmatrix} I \\ 0 \end{bmatrix} \left([I \quad 0] P^{-1} \begin{bmatrix} I \\ 0 \end{bmatrix} \right)^{-1} x_0 = P^{-1} \frac{\partial P}{\partial x_i} P^{-1} \begin{bmatrix} I \\ 0 \end{bmatrix} \left([I \quad 0] P^{-1} \begin{bmatrix} I \\ 0 \end{bmatrix} \right)^{-1} x_0$$

$$= P^{-1} \left(\left(\Gamma_i \begin{bmatrix} X_0 \\ U \end{bmatrix}^{\dagger} \right)^{\mathsf{T}} QX \begin{bmatrix} X_0 \\ U \end{bmatrix}^{\dagger} + \left(X \begin{bmatrix} X_0 \\ U \end{bmatrix}^{\dagger} \right)^{\mathsf{T}} Q \Gamma_i \begin{bmatrix} X_0 \\ U \end{bmatrix}^{\dagger} \right) P^{-1} \begin{bmatrix} I \\ 0 \end{bmatrix} \left([I \quad 0] P^{-1} \begin{bmatrix} I \\ 0 \end{bmatrix} \right)^{-1} x_0$$

$$= P^{-1} \left(\left(\Gamma_i \begin{bmatrix} X_0 \\ U \end{bmatrix}^{\dagger} \right)^{\mathsf{T}} Q \begin{bmatrix} O_T & F_T \end{bmatrix} + \begin{bmatrix} O_T & F_T \end{bmatrix}^{\mathsf{T}} Q \Gamma_i \begin{bmatrix} X_0 \\ U \end{bmatrix}^{\dagger} \right) P^{-1} \begin{bmatrix} I \\ 0 \end{bmatrix} \left([I \quad 0] P^{-1} \begin{bmatrix} I \\ 0 \end{bmatrix} \right)^{-1} x_0, \tag{52}$$

$$\nabla F_{X,i}^{(2)} = P^{-1} \begin{bmatrix} I \\ 0 \end{bmatrix} \frac{\partial \left([I \quad 0] P^{-1} \begin{bmatrix} I \\ 0 \end{bmatrix} \right)^{-1}}{\partial x_i} x_0$$

$$= P^{-1} \begin{bmatrix} I \\ 0 \end{bmatrix} \left([I \quad 0] P^{-1} \begin{bmatrix} I \\ 0 \end{bmatrix} \right)^{-1} [I \quad 0] \frac{\partial P^{-1}}{\partial x_i} \begin{bmatrix} I \\ 0 \end{bmatrix} \left([I \quad 0] P^{-1} \begin{bmatrix} I \\ 0 \end{bmatrix} \right)^{-1} x_0$$

$$= P^{-1} \begin{bmatrix} I \\ 0 \end{bmatrix} \left([I \quad 0] P^{-1} \begin{bmatrix} I \\ 0 \end{bmatrix} \right)^{-1} [I \quad 0] P^{-1} \frac{\partial P}{\partial x_i} P^{-1} \begin{bmatrix} I \\ 0 \end{bmatrix} \left([I \quad 0] P^{-1} \begin{bmatrix} I \\ 0 \end{bmatrix} \right)^{-1} x_0. \tag{53}$$

is a constant independent of N. Thus, from (56), $\|\nabla F_{X,i}^{(2)}\| \leq k_{X,i}^{(2)}/N$, where $k_{X,i}^{(2)} > 0$ does not depend on N. Finally, from (51) and $\|\nabla F_{X,i}\| \leq \|\nabla F_{X,i}^{(1)}\| + \|\nabla F_{X,i}^{(2)}\|$, we conclude that $\|\nabla F_{X,i}\| \leq k_{X,i}/N$, where $k_{X,i}$ is independent of N. □

K. PROOF OF THEOREM IV.5

By Theorem IV.2,

$$\begin{aligned}\mathbb{P}[\Delta G \geq \tau] &\leq \frac{1}{\tau}\left(\sum_{i\in\text{supp}(\Delta_X)} c_{X,i}\,\mathbb{E}[|\delta_{X,i}|]\right)\\ &\leq \frac{\|V^{\frac{1}{2}}\|}{\tau}\left(\sum_{i\in\text{supp}(\Delta_X)} \|\nabla F_{X,i}\|\,\mathbb{E}[|\delta_{X,i}|]\right)\\ &\leq \frac{\|V^{\frac{1}{2}}\|}{\tau}\left(|\,\text{supp}(\Delta_X)|\,\max_i\left\{\|\nabla F_{X,i}\|\,\mathbb{E}[|\delta_{X,i}|]\right\}\right)\end{aligned} \tag{58}$$

where in the second step we used that $c_{X,i} = \|V^{\frac{1}{2}}\nabla F_{X,i}\| \leq \|V^{\frac{1}{2}}\|\|\nabla F_{X,i}\|$. Since the distributions of $\delta_{X,i}$ are independent of N so are $\mathbb{E}[|\delta_{X,i}|]$. Hence, by Lemma IV.4, it follows that $\max_i\{\|\nabla F_{X,i}\|\,\mathbb{E}[|\delta_{X,i}|]\} \leq \max_i\|\nabla F_{X,i}\|\max_i\mathbb{E}[|\delta_{X,i}|] \leq \kappa_{X,i}/N$, where $\kappa_{X,i}$ is independent of N and we used the assumption $\mathbb{E}[|\delta_{x_i}|] < \infty$. If $\text{supp}(\Delta_X)$ is a sublinear function of N, (26) follows from the latter inequalities and (58). □

REFERENCES

[1] L. Pallottino, V. G. Scordio, A. Bicchi, and E. Frazzoli, "Decentralized cooperative policy for conflict resolution in multivehicle systems," *IEEE Trans. Robot.*, vol. 23, no. 6, pp. 1170–1183, Dec. 2007.

[2] K. K. Oh, M. C. Park, and H. S. Ahn, "A survey of multi-agent formation control," *Automatica*, vol. 53, pp. 424–440, 2015.

[3] W. Ren and R. W. Beard, "Consensus seeking in multi-agent systems under dynamically changing interaction topologies," *IEEE Trans. Autom. Control*, vol. 50, no. 5, pp. 655–661, May 2005.

[4] M. Mesbahi and M. Egerstedt, *Graph Theoretic Methods in Multiagent Networks*. Princeton, NJ, USA: Princeton Univ. Press, 2010.

[5] D. Moore, J. Leonard, D. Rus, and S. Teller, "Robust distributed network localization with noisy range measurements," in *Proc. ACM Conf. Embedded Networked Sensor Syst.*, 2004, pp. 50–61.

[6] F. Bullo, J. Cortes, and S. Martinez, *Distributed Control of Robotic Networks: A Mathematical Approach to Motion Coordination Algorithms*. Princeton, NJ, USA: Princeton Univ. Press, 2009.

[7] D. K. Molzahn et al., "A survey of distributed optimization and control algorithms for electric power systems," *IEEE Trans. Smart Grid*, vol. 8, no. 6, pp. 2941–2962, Nov. 2017.

[8] C. T. Butts, "Network inference, error, and informant (in)accuracy: A Bayesian approach," *Social Netw.*, vol. 25, no. 2, pp. 103–140, 2003.

[9] L. A. N. Amaral, "A truer measure of our ignorance," *Proc. Nat. Acad. Sci.*, vol. 105, no. 19, pp. 6795–6796, 2008.

[10] L. Ljung, *System Identification: Theory for the User*. Hoboken, NJ, USA: Prentice Hall, 1987.

[11] T. Katayama, *Subspace Methods for System Identification* (Communications and Control Engineering Series). London, U.K.: Springer-Verlag, 2005.

[12] M. T. Angulo, J. A. Moreno, G. Lippner, A.-L. Barabási, and Y.-Y. Liu, "Fundamental limitations of network reconstruction from temporal data," *J. Roy. Soc. Interface*, vol. 14, no. 127, 2017, Art. no. 20160966.

[13] D. Achlioptas, A. Clauset, D. Kempe, and C. Moore, "On the bias of traceroute sampling: Or, power-law degree distributions in regular graphs," *J. ACM*, vol. 56, no. 4, pp. 1–28, 2009.

[14] K. Jung, D. Shah, and J. Shin, "Distributed averaging via lifted Markov chains," *IEEE Trans. Inf. Theory*, vol. 56, no. 1, pp. 634–647, Jan. 2010.

[15] J. G. Ziegler and N. B. Nichols, "Optimum settings for automatic controllers," *Trans. ASME*, vol. 64, no. 8, pp. 759–765, 1942.

[16] G. Baggio, V. Katewa, and F. Pasqualetti, "Data-driven minimum-energy controls for linear systems," *IEEE Control Syst. Lett.*, vol. 3, no. 3, pp. 589–594, Jul. 2019.

[17] G. Baggio, D. S. Bassett, and F. Pasqualetti, "Data-driven control of complex networks," *Nature Commun.*, vol. 12, no. 1, 2021, Art. no. 1429.

[18] N. Monshizadeh, "Amidst data-driven model reduction and control," *IEEE Control Syst. Lett.*, vol. 4, no. 4, pp. 833–838, Oct. 2020.

[19] I. Markovsky and P. Rapisarda, "Data-driven simulation and control," *Int. J. Control*, vol. 81, no. 12, pp. 1946–1959, 2008.

[20] 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.

[21] H. J. Van Waarde, J. Eising, H. L. Trentelman, and M. K. Camlibel, "Data informativity: A new perspective on data-driven analysis and control," *IEEE Trans. Autom. Control*, vol. 65, no. 11, pp. 4753–4768, Nov. 2020.

[22] J. Berberich, A. Koch, C. W. Scherer, and F. Allgöwer, "Robust data-driven state-feedback design," in *Proc. IEEE Amer. Control Conf.*,, 2020, pp. 1532–1538.

[23] J. Coulson, J. Lygeros, and F. Dörfler, "Data-enabled predictive control: In the shallows of the DeePC," in *Proc. IEEE Eur. Control Conf.*, 2019, pp. 307–312.

[24] J. Berberich, J. Köhler, M. A. Müller, and F. Allgöwer, "Data-driven model predictive control with stability and robustness guarantees," *IEEE Trans. Autom. Control*, vol. 66, no. 4, pp. 1702–1717, Apr. 2021.

[25] P. Tabuada, W. L. Ma, J. Grizzle, and A. D. Ames, "Data-driven control for feedback linearizable single-input systems," in *Proc. IEEE Conf. Decis. Control*, 2017, pp. 6265–6270.

[26] J. Coulson, J. Lygeros, and F. Dörfler, "Distributionally robust chance constrained data-enabled predictive control," *IEEE Trans. Autom. Control*, vol. 67, no. 7, pp. 3289–3304, Jul. 2022.

[27] H. J. Van Waarde, M. K. Camlibel, and M. Mesbahi, "From noisy data to feedback controllers: Non-conservative design via a matrix S-Lemma," *IEEE Trans. Autom. Control*, vol. 67, no. 1, pp. 162–175, Jan. 2022.

[28] 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.

[29] F. Celi, G. Baggio, and F. Pasqualetti, "Distributed learning of optimal controls for linear systems," in *Proc. IEEE 60th Conf. Decis. Control*, 2021, pp. 5764–5769.

[30] C. A. Alonso, F. Yang, and N. Matni, "Data-driven distributed and localized model predictive control," *IEEE Open J. Control Syst.*, vol. 1, pp. 29–40, 2022.

[31] 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. IEEE Conf. Decis. Control*, 2021, pp. 466–471.

[32] J. Eising and J. Cortés, "Informativity for centralized design of distributed controllers for networked systems," in *Proc. IEEE Eur. Control Conf.*, 2022, pp. 681–686.

[33] X. Wang, J. Sun, G. Wang, F. Allgöwer, and J. Chen, "Data-driven control of distributed event-triggered network systems," *IEEE/CAA J. Automatica Sinica*, vol. 10, no. 2, pp. 351–364, Feb. 2023.

[34] Y. Li, X. Wang, J. Sun, G. Wang, and J. Chen, "Data-driven consensus control of fully distributed event-triggered multi-agent systems," *Sci. China Inf. Sci.*, vol. 66, no. 5, 2023, Art. no. 152202.

[35] A. Cherukuri, B. Gharesifard, 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.

[36] 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.

[37] 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.

[38] P. Lancaster and L. Rodman, *Algebraic Riccati Equations*. Oxford, U.K.: Clarendon Press, 1995.

[39] F. Celi and F. Pasqualetti, "Data-driven meets geometric control: Zero dynamics, subspace stabilization, and malicious attacks," *IEEE Control Syst. Lett.*, vol. 6, pp. 2569–2574, 2022.

[40] F. Celi, G. Baggio, and F. Pasqualetti, "Closed-form estimates of the LQR gain from finite data," in *Proc. IEEE Conf. Decis. Control*, 2022, pp. 4016–4021.
[41] T. Kollo and D. V. Rosen, *Advanced Multivariate Statistics With Matrices. Mathematics and Its Applications*, Berlin, Germany: Springer-Verleg, 2005.
[42] A. W. Van Der Vaart, *Asymptotic statistics* (Cambridge Series in Statistical and Probabilistic Mathematics), vol. 3. Cambridge, U.K.: Cambridge Univ. Press, 2000.
[43] D. J. Watts and S. H. Strogatz, "Collective dynamics of 'small-world' networks," *Nature*, vol. 393, no. 6684, pp. 440–442, 1998.
[44] "Build watts-strogatz small world graph model". Accessed: Jan. 16, 2023. [Online]. Available: https://www.mathworks.com/help/matlab/math/build-watts-strogatz-small-world-graph-model.html
[45] F. Farokhi, I. Shames, and K. H. Johansson, "Distributed MPC via dual decomposition and alternative direction method of multipliers," in *Distributed Model Predictive Control Made Easy*. Netherlands, Dordrecht: Springer, 2014, pp. 115–131.
[46] F. Pasqualetti, R. Carli, and F. Bullo, "Distributed estimation via iterative projections with application to power network monitoring," *Automatica*, vol. 48, no. 5, pp. 747–758, 2012.
[47] J. A. Tropp, "User-friendly tail bounds for sums of random matrices," *Found. Comput. Math.*, vol. 12, no. 4, pp. 389–434, 2012.
[48] D. S. Bernstein, *Matrix Mathematics*, 2nd ed., Princeton, NJ, USA: Princeton Univ. Press, 2009.

FEDERICO CELI (Student Member, IEEE) received the Laurea degree (B.Sc. equivalent) in computer engineering and the Laurea Magistrale (M.Sc. equivalent) degree in robotics and automation engineering from the University of Pisa, Pisa, Italy, in 2014 and 2018, respectively. He is currently working toward the Ph.D. degree in mechanical engineering with the University of California, Riverside, CA, USA. From December 2017 to May 2018, he was a Visiting Scholar with the Institute for Robotics and Intelligent Machines, Georgia Tech, Atlanta, GA, USA. His main research interests lie at the intersection of networked and data-driven controls.

GIACOMO BAGGIO (Member, IEEE) received the Ph.D. degree in information engineering from the University of Padova, Padua, Italy, in 2018. He was a Visiting Scholar with the Department of Engineering, University of Cambridge, Cambridge, U.K., from September 2015 to May 2016, and Postdoctoral Scholar with the Department of Mechanical Engineering, University of California Riverside, CA, USA, from March 2018 to September 2019. He is currently an Assistant Professor with the Department of Information Engineering, University of Padova. His research interests include the area of systems and control theory with applications to networks and data science.

FABIO PASQUALETTI (Member, IEEE) received the Laurea degree (B.Sc. equivalent) in computer engineering and the Laurea Magistrale degree (M.Sc. equivalent) in automation engineering from the University of Pisa, Italy, in 2004, and 2007, respectively, and the Doctor of Philosophy degree in mechanical engineering with the University of California, Santa Barbara, CA, USA, in 2012. He is currently a Professor of mechanical engineering with the University of California, Riverside, CA. His main research interests include the analysis and control of network systems, security of cyber-physical systems, and network neuroscience. He is a Member SIAM.