Data-Driven Structured Policy Iteration for Homogeneous Distributed Systems

Siavash Alemzadeh†, Shahriar Talebi†, *Member, IEEE*,
and Mehran Mesbahi, *Fellow, IEEE*

***Abstract*— Control of networked systems, comprised of interacting agents, is often achieved through modeling the underlying interactions. Constructing accurate models of such interactions–in the meantime–can become prohibitive in applications. Data-driven control methods avoid such complications by directly synthesizing a controller from the observed data. In this paper, we propose an algorithm referred to as Data-driven Structured Policy Iteration (D2SPI), for synthesizing an efficient feedback mechanism that respects the sparsity pattern induced by the underlying interaction network. In particular, our algorithm uses temporary "auxiliary" communication links in order to enable the required information exchange on a (smaller) sub-network during the "learning phase"—links that will be removed subsequently for the final distributed feedback synthesis. We then proceed to show that the learned policy results in a stabilizing structured policy for the entire network. Our analysis is then followed by showing the stability and convergence of the proposed distributed policies throughout the learning phase, exploiting a construct referred to as the "Patterned monoid." The performance of D2SPI is then demonstrated using representative simulation scenarios.**

***Index Terms*— Structured control, Patterned monoids, Data-driven Policy Iteration, Networked control systems**

I. Introduction

In RECENT years, there has been a renewed interest in distributed control of large-scale systems. The unprecedented interdependence and size of the data generated by such systems have necessitated a distributed approach to policy computation in order to influence or direct their behavior and performance. In these scenarios, collective actions are often synthesized via local decisions, informed by a *structured* information exchange mechanism. An important roadblock for centralized control design methods, is thereby, their scalability and shortcomings in utilizing the underlying structure of large-scale interconnected systems.[1]

Structured control synthesis in the meantime is generally an NP-hard constrained optimization problem [4]. Hence, distributed control design for large-scale systems has often been pursued not necessarily to characterize optimal policies per se, but to devise efficient (possibly suboptimal) control mechanisms that exploit the inherent system structure. In parallel, recent advances in measurement technologies have made available an unprecedented amount of data, motivating how offline and online data-processing can be leveraged for data-driven decision-making on high-dimensional complex systems.

In this work, we propose the linear-quadratic regulator (LQR)-based algorithm, coined Data-driven Structured Policy Iteration (D2SPI), to iteratively learn stabilizing controllers for unknown but identical linear dynamical systems that are connected via a network induced by the coupling in their performance. The setup is a particular realization of cooperative game-theoretic decision-making (see remarks under Footnote 2). This class of synthesis problems is motivated by applications such as formation flight [5] and distributed camera systems [6], where the dynamics of the network nodes (agents) cannot be precisely parameterized. D2SPI is built upon a data-driven learning phase on a subgraph in a large network. This subgraph includes the agent with maximum degree in the network and requires enabling *auxiliary* links within this subgraph in order to iteratively learn a stabilizing structured controller (optimal for the subgraph) for the entire network–see Figure 1. This "extension" synthesis procedure utilizes a symmetry property of the networked systems, that we refer to as the *Patterned monoid* (see Section II-A).

The remainder of the paper is organized as follows. In §II we introduce the problem setup and motivation behind our work, and provide an overview of the related literature (§II-B). In §III, we present and analyze the D2SPI algorithm, followed by the theoretical analysis in §IV. Illustrative examples are provided in §V, followed by concluding remarks in §VI.

Notation. The operator $\mathrm{diag}(\cdot)$ makes a square diagonal matrix out of the elements of its argument; $\mathrm{vech}(\cdot)$ on the other hand, takes a square matrix and stacks the lower left triangular half (including the diagonal) into a single vector. We use

†S. Alemzadeh and S. Talebi contributed equally to this work.

The research of the authors has been supported by NSF grant SES-1541025, NSF grant ECCS-2149470, and AFOSR grant FA9550-20-1-0053.

This work was carried out while the first two authors were with the William E. Boeing department of Aeronautics and Astronautics at the University of Washington, Seattle. S. Alemzadeh is currently with Microsoft Corporation and S. Talebi is with the John A. Paulson School of Engineering and Applied Sciences at Harvard University; emails: alems@uw.edu, talebi@seas.harvard.edu, and mesbahi@uw.edu.

A preliminary version of this work has been presented at the 60th IEEE Conference on Decision and Control [1]. In comparison, the current manuscript expands on connections to broader related works, provides proofs of all results, as well as including the suboptimality guarantees. Extended simulation results have also been included in this version of the manuscript.

[1] $\mathcal{O}(n^3)$ complexity of solving the *Algebraic Riccati Equation* [2] and scalability issues of *Model Predictive Control* [3] are among such examples.

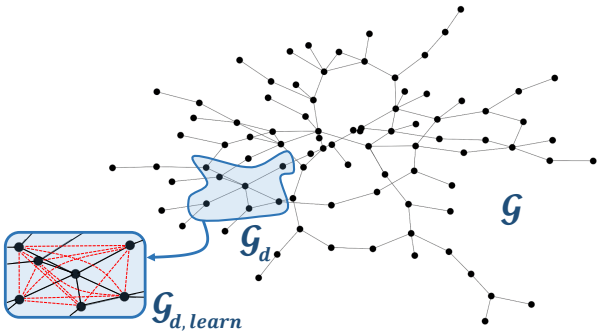


Fig. 1: Addition of auxiliary links (dashed red) to the subgraph $\mathcal{G}_d$ during the policy learning phase. The size of the subgraph depends on the maximum degree of the original graph $\mathcal{G}$.

$N \succ 0$ ($\succeq 0$) to declare N as a positive-(semi)definite matrix. The ith eigenvalue and spectral radius of M are denoted by $\lambda_i(M)$ and $\rho(M)$; M is called Schur stable when $\rho(M) < 1$. We say that an n-dimensional linear system parameterized by the pair (A, B) is *controllable* if the controllability matrix $\mathcal{C} = [B \quad AB \quad \dots \quad A^{n-1}B]$ has a full-rank. We denote the *Kronecker product* of two matrices by $\otimes$. For a block matrix $\mathbf{F}$, by $[\mathbf{F}]_{rk}$ we imply the rth row and kth column "block" component with appropriate dimensions. An (undirected) graph is characterized by $\mathcal{G} = (\mathcal{V}_\mathcal{G}, \mathcal{E}_\mathcal{G})$ where $\mathcal{V}_\mathcal{G}$ is the set of nodes and $\mathcal{E}_\mathcal{G} \subseteq \mathcal{V}_\mathcal{G} \times \mathcal{V}_\mathcal{G}$ denotes the set of edges. An edge exists from node i to j if (the unordered pair) $(i, j) \in \mathcal{E}_\mathcal{G}$; this is also specified by writing $j \in \mathcal{N}_i$, where $\mathcal{N}_i$ is the set of neighbors of node i. We designate the maximum degree of $\mathcal{G}$ by $d_{\max}(\mathcal{G})$. Finally, the graph $\mathcal{G}$ can be represented using matrices such as the Laplacian $\mathcal{L}_\mathcal{G}$ or the adjacency $\mathcal{A}_\mathcal{G}$. To distinguish between system dynamics quantities related to the entire graph and a subgraph, we utilize hat and tilde notation, respectively. A *semigroup* is a set and a binary operator in which the multiplication operation is associative (but its elements need not have inverses). A *monoid* is a semigroup with an identity element. A *group* is a monoid each of whose elements is invertible. We denote the set of symmetric $n \times n$ real matrices by $\mathbb{S}^n$, and the set invertible ones by $\mathrm{GL}(n, \mathbb{R})$—which is a group under matrix multiplication also known as the general linear group.

II. Problem Setup

Consider a network of identical agents with interdependencies induced by a network-level objective. In particular, we assume that the system contains N agents forming a graph $\mathcal{G} = (\mathcal{V}_\mathcal{G}, \mathcal{E}_\mathcal{G})$, where each node of the graph in $\mathcal{V}_\mathcal{G}$ represents a linear discrete-time system,

$$\boldsymbol{x}_{i,t+1} = A\boldsymbol{x}_{i,t} + B\boldsymbol{u}_{i,t}, \qquad i = 1, 2, \dots, N, \tag{1}$$

with $\boldsymbol{x}_{i,t} \in \mathbb{R}^n$ and $\boldsymbol{u}_{i,t} \in \mathbb{R}^m$ denoting the state and input of agent i at time-step t, respectively. The unknown system matrices $A \in \mathbb{R}^{n\times n}$ and $B \in \mathbb{R}^{n\times m}$ are assumed to form a controllable pair. The network dynamics can compactly be represented as,

$$\hat{\mathbf{x}}_{t+1} = \hat{\mathbf{A}}\hat{\mathbf{x}}_t + \hat{\mathbf{B}}\hat{\mathbf{u}}_t, \tag{2}$$

where $\hat{\mathbf{x}}_t \in \mathbb{R}^{Nn}$ and $\hat{\mathbf{u}}_t \in \mathbb{R}^{Nm}$ are comprised of the states and inputs of entire network, $\hat{\mathbf{x}}_t = \begin{bmatrix}\boldsymbol{x}_{1,t}^\intercal & \dots & \boldsymbol{x}_{N,t}^\intercal\end{bmatrix}^\intercal$, $\hat{\mathbf{u}}_t = \begin{bmatrix}\boldsymbol{u}_{1,t}^\intercal & \dots & \boldsymbol{u}_{N,t}^\intercal\end{bmatrix}^\intercal$, with $\hat{\mathbf{A}} \in \mathbb{R}^{Nn\times Nn}$ and $\hat{\mathbf{B}} \in \mathbb{R}^{Nn\times Nm}$ are in block diagonal forms $\hat{\mathbf{A}} = \mathrm{I}_N \otimes A$ and $\hat{\mathbf{B}} = \mathrm{I}_N \otimes B$. The agents' interconnections are represented by edges $\mathcal{E}_\mathcal{G} \subseteq \mathcal{V}_\mathcal{G} \times \mathcal{V}_\mathcal{G}$ that can facilitate a distributed feedback design. We do not assume that $\mathcal{G}$ is necessarily connected; the motivation for this becomes apparent subsequently. Let $\mathcal{N}_i$ denote the set of neighbors of node i in $\mathcal{G}$ (excluding itself). Then, based on the underlying communication graph and for any choice of positive integers a and b, we define a linear subspace of real-valued $aN \times bN$ matrices as,

$$\mathcal{U}_{a,b}^N(\mathcal{G}) := \big\{\mathbf{M} \in \mathbb{R}^{aN\times bN} \,|\, [\mathbf{M}]_{ij} = \mathbf{0} \text{ if } j \notin \mathcal{N}_i \cup \{i\}, \; [\mathbf{M}]_{ij} \in \mathbb{R}^{a\times b}, \; i, j = 1, \cdots, N\big\}.$$

Without having access to the system parameters A and B, we are interested in designing linear feedback gains, consistent with the desired sparsity pattern induced by the network, using data generated by (2). More precisely, given an initial condition $\hat{\mathbf{x}}_1$, the distributed (structured) optimal control problem assumes the form,

$$\begin{aligned} &\min_{\hat{\mathbf{K}}} \quad \textstyle\sum_{t=1}^\infty \hat{\mathbf{x}}_t^\intercal \hat{\mathbf{Q}} \hat{\mathbf{x}}_t + \hat{\mathbf{u}}_t^\intercal \hat{\mathbf{R}} \hat{\mathbf{u}}_t \\ &\text{s.t.} \quad (2), \quad \hat{\mathbf{u}}_t = \hat{\mathbf{K}}\hat{\mathbf{x}}_t, \quad \hat{\mathbf{K}} \in \mathcal{U}_{m,n}^N(\mathcal{G}), \end{aligned} \tag{3}$$

where $\hat{\mathbf{K}}$ stabilizes the pair $(\hat{\mathbf{A}}, \hat{\mathbf{B}})$ (i.e., $\rho(\hat{\mathbf{A}} + \hat{\mathbf{B}}\hat{\mathbf{K}}) < 1$), $\hat{\mathbf{R}} = \mathrm{I}_N \otimes R$ and $\hat{\mathbf{Q}} = I_N \otimes Q_1 + \mathcal{L}_\mathcal{G} \otimes Q_2$ for some given cost matrices $Q_1 \succ 0$, $Q_2 \succeq 0$, $R \succ 0$. Note that $\hat{\mathbf{Q}} \in \mathcal{U}_{n,n}^N(\mathcal{G})$ is positive definite. Such interdependence induced through the cost has been considered in a graph-based distributed control framework; see for instance [7]–[10]. In a nutshell, the first term in $\hat{\mathbf{Q}}$ encodes the cost pertinent to state regulation for each agent, while the second term, captures the "disagreement" cost between the neighbors.[2]

In this work, we propose a data-guided suboptimal solution for (3), not relying on knowledge of the system parameters $\hat{\mathbf{A}}$ and $\hat{\mathbf{B}}$. Instead, our approach relies on the system's input-state time series for synthesizing distributed feedback control on $\mathcal{G}$. A summary of challenges in analyzing this problem is listed as follows:

1) The constrained optimization problem in (3) is in general NP-hard [4], [12]. In particular, the problem of stabilization by decentralized static state feedback is NP-hard if one imposes a bound on the norm of the controller [12]. Even though the result is not shown for the case with no bound on the controller

[2]One instance of such an interactive cost among agents appears in the cooperative game setup where agent i aims to solve the minimization problem,

$$\min_{(\boldsymbol{u}_{i,t})_{t=0}^\infty \in \ell_2} \mathbf{J}_i(\hat{\mathbf{x}}_1, \hat{\mathbf{u}}_t) = N \sum_{t=0}^\infty \Big(\boldsymbol{x}_{i,t}^\intercal Q_1 \boldsymbol{x}_{i,t} + \boldsymbol{u}_{i,t}^\intercal R \boldsymbol{u}_{i,t} + \sum_{j\in\mathcal{N}_i} (\boldsymbol{x}_{j,t} - \boldsymbol{x}_{i,t})^\intercal Q_2 (\boldsymbol{x}_{j,t} - \boldsymbol{x}_{i,t}) \Big).$$

Then, it is well-known that the set of Pareto front solution of this game can be obtained by minimizing the parametric cost function,

$$\min_{(\boldsymbol{u}_{1,t})_{t=0}^\infty, \dots, (\boldsymbol{u}_{N,t})_{t=0}^\infty \in \ell_2} \sum_{i=1}^N \alpha_i \mathbf{J}_i(\hat{\mathbf{x}}_1, \hat{\mathbf{u}}_t),$$

parameterized by $\alpha_1, \dots, \alpha_N$ where $\alpha_i \in [0, 1]$ and $\sum_i \alpha_i = 1$ (see e.g. [11]). Therefore, a cost such as in (3) can be viewed as a special case of the fair Pareto optimal solution with the choice of $\alpha_i = 1/N$ for all i.

parameters, the general problem is commonly believed to be a computationally hard problem. Note that stabilization is a necessity for feasibility of optimization problem in (3) for arbitrary initial state $\mathbf{x}_0$ whenever $Q_1 \succ 0$. This is simply because the cost is unbounded for an unstable policy. Nonetheless, based on the complete knowledge of the system parameters, this problem has been investigated under variety of assumptions [7], [13]–[15], or approached directly with the aid of projected gradient-based policy updates [16], [17].

2) In the meantime, policies obtained via data-driven approaches, do not necessarily respect the hard constraints on $\hat{\mathbf{K}} \in \mathcal{U}_{m,n}^N(\mathcal{G})$ posed in (3). In particular, we point out that "projection" onto the intersection of the constraint imposed by the network and stabilizing controllers is not straightforward due to the intricate geometry of the set of stabilizing controllers [18].

3) Another key challenge in adopting data-driven methods for the entire network is rooted in the "curse of dimensionality" inherent in analysis of large-scale systems. In fact, even collecting data from the entire network can be prohibitive.

4) Finally, it is often impossible in applications to pause the operation of the network for data collection or decision-making purposes.[3]

A. Structures in the Problem and our Approach

In this work the sparsity requirement $\hat{\mathbf{K}} \in \mathcal{U}_{m,n}^N(\mathcal{G})$ is considered as a *hard constraint* for control synthesis, and as such, the corresponding optimization is challenging in general. Hence, we shift our attention from the optimal solution of (3) towards a "reasonable" suboptimal stabilizing distributed controller with a scalable computational cost. We aim to exploit the problem structure that is incurred due to the homogeneity of system dynamics and the pattern in performance index respecting the underlying graph topology—see the definition of $\hat{\mathbf{Q}}$ depending on the graph Laplacian $\mathcal{L}_\mathcal{G}$ in (3). Additionally, as system parameters in (2) are unknown, we adapt a $\mathcal{Q}$-learning procedure to our setup that provably converges to a distributed policy with suboptimality guarantees.

In the *absence* of the sparsity constraint $\hat{\mathbf{K}} \in \mathcal{U}_{m,n}^N(\mathcal{G})$, a learning approach for solving the optimal control problem (3) is the well-known $\mathcal{Q}$-learning procedure that was first introduced by Bradtke in [19], [20]. Adopting this approach for (3) would require utilizing the input-state data trajectories of the entire networked systems in (2) to implement a quasi-Newton method for iteratively updating $\hat{\mathbf{K}}$ introduced by Hewer in [21]. If the system is controllable, Hewer's algorithm converges to the globally optimal solution with a quadratic rate, and so does the Bradtke's algorithm if in addition the data trajectories are "informative" enough—this is usually satisfied by a sufficient condition on the input signal referred to as "persistence of excitation." The main issue with this approach is the fact that the policy obtained in this way, will not respect the hard constraint of $\hat{\mathbf{K}} \in \mathcal{U}_{m,n}^N(\mathcal{G})$ as was posed in the optimization (3). This issue is particularly critical when a projection of the iterated policy on the set $\mathcal{U}_{m,n}^N(\mathcal{G})$ is not practical, or even fails to be stabilizing. Additionally, collecting data trajectory from entire networked system in (2) can be expensive.

Inspired by the $\mathcal{Q}$-learning approach, we propose a model-free structured policy iteration scheme for the synthesis problem (3) with iterative stability, convergence, and reasonable performance guarantees. While the detailed algorithm is presented subsequently in Section III, in what follows we summarize the key steps of our design procedure for distributed data-driven policy iteration:

1) Inherent to our distributed learning algorithm is a synthesis sub-problem whose dimension is related only to the maximum degree in underlying graph rather than the dimension of the original network. In particular, we will reason that for the learning phase, our method only requires data collection from a (specific) smaller sub-network $\mathcal{G}_d \subseteq \mathcal{G}$ with size $d = d_{\max}(\mathcal{G}) + 1$. This subgraph is substantially smaller than the original graph whenever $d_{\max}(\mathcal{G})$ is significantly smaller than N, reflecting the empirical feature of many real-world networks. In the meantime, our approach requires adding temporary communication links to the subgraph $\mathcal{G}_d$ to make it a clique during the learning phase– we will discuss why this learning phase clique is required in Step 3 below—see Figure 1. The additional links are subsequently removed for the final data-driven feedback design. In robotic applications, such a clique can be established by temporary increasing the transmission range for neighboring nodes of the agent with maximum degree. Since longer range information transfer is generally costly, power levels for these agents are "dialed back" to their original settings following completion of the learning phase.

2) Then, we adapt the $\mathcal{Q}$-learning technique in order to learn a specific optimal policy for this sub-problem using data only from the systems in the subgraph $\mathcal{G}_d$. In particular, the proposed algorithm learns an optimal policy for the subgraph with a (off-)diagonal pattern consisting of two distinct system-level policies K^* and L^*, representing the "individual" and "cooperative" components, respectively. Specifically, for any integer $r \geq 2$, we can define a linear subspace of $\mathbb{R}^{rn\times rn}$ as

$$\mathrm{L}(r\times n,\mathbb{R}) := \big\{\mathbf{N}_r \in \mathbb{R}^{rn\times rn} \,\big|\, \mathbf{N}_r = \mathrm{I}_r \otimes (A-B) + \mathbb{1}_r\mathbb{1}_r^\intercal \otimes B, \text{ for some } A, B\big\},$$

which is also closed under matrix multiplication. Then, the policy learned for the subgraph $\mathcal{G}_d$ takes the form $\widetilde{\mathbf{K}}^* = \mathrm{I}_d \otimes \big(K^* - L^*\big) + \mathbb{1}\mathbb{1}^\intercal \otimes L^*$ which also implies that the closed-loop system $\widetilde{\mathbf{A}}_{\widetilde{\mathbf{K}}^*} := \widetilde{\mathbf{A}} + \widetilde{\mathbf{B}}\widetilde{\mathbf{K}}^*$ lies in $\mathrm{L}(d\times n,\mathbb{R})$. This is particularly useful from a design perspective as it becomes clear in the next step.

3) Next, we define the *Patterned monoid* as

$$\mathrm{PM}(r\times n,\mathbb{R}) := \big\{\mathbf{N}_r \in \mathrm{L}(r\times n,\mathbb{R}) \cap \mathrm{GL}(rn,\mathbb{R}) \,\big|\, \text{for some } A \in \mathrm{GL}(n,\mathbb{R}) \cap \mathbb{S}^n,\ B \in \mathbb{S}^n\big\}.$$

Note that the Patterned monoid $\mathrm{PM}(r\times n,\mathbb{R})$ is indeed a sub-monoid of $\mathrm{GL}(rn,\mathbb{R})$—following by Lemma 2—i.e., it is closed under matrix multiplications and contains the identity

[3]For example, consider an operational large-scale group of homogeneous aerial vehicles that need to coordinate their relative states (in addition to their respective state regulation) over a network induced by their proximity [5].

matrix[4]. The next observation underscores the relevance of Patterned monoids in system analysis.

Proposition 1: For a Schur stable matrix $A \in \mathrm{L}(r \times n, \mathbb{R})$ and $0 \prec Q \in \mathbb{S}^{rn}$, let P denote the unique solution to the corresponding discrete-time Lyapunov equation, i.e., $P = A^\intercal P A + Q$. Then, $P \in \mathrm{PM}(r \times n, \mathbb{R})$ if and only if $Q \in \mathrm{PM}(r \times n, \mathbb{R})$.

The invariance of the Lyapunov equation under the action of Patterned monoid has important implications for our data-driven synthesis for large-scale networks—see Proposition 2. Another key ingredient of our approach, motivated by utilizing the Patterned monoid in the context of $\mathcal{Q}$-learning, involves allowing temporary communication links on the subgraph $\mathcal{G}_d$ in order to make it a clique only *during the learning phase*—links that are subsequently removed. It then follows that the state cost parameter $\widetilde{\mathbf{Q}}$ for the subgraph $\mathcal{G}_d$ lies in $\mathrm{PM}(d \times n, \mathbb{R})$. Then, as the obtained closed-loop system $\widetilde{\mathbf{A}}_{\widetilde{\mathbf{K}}}$ at each iteration of the learning phase lies in $\mathrm{L}(d \times n, \mathbb{R})$, Proposition 1 would imply that the associated cost matrix $\widetilde{\mathbf{P}}_{\widetilde{\mathbf{K}}}$ must lie in $\mathrm{PM}(d \times n, \mathbb{R})$ with two components $P_1 \in \mathrm{GL}(n, \mathbb{R}) \cap \mathbb{S}^n$ and $P_2 \in \mathbb{S}^n$.

4) Next, an iterative $\mathcal{Q}$-learning procedure is designed based on both the patterned structure of the policy with system-level components K and L, and the cost matrix with P_1 and P_2 components. This procedure provably converges to an optimal policy for the subgraph $\mathcal{G}_d$–see Theorem 2.

5) We note that terminating the operation of the entire networked system for the purpose of data collection/learning is often infeasible in real-world applications. For example, disrupting the operation of power generators and loads for improving their respective network-level performance is highly undesirable. Therefore, in addition to learning the optimal policy for the subgraph, in this work, we aim to simultaneously devise and update a policy for the rest of the network. In this direction, our algorithm iteratively learns a "stability margin" τ according to the learned components of the policy at each iteration–see Proposition 3. This feature, together with homogeneity of the network, facilitates extending the policy synthesis procedure to a stabilizing policy for the entire network by utilizing the individual and cooperative components learned from the subgraph $\mathcal{G}_d$–see Theorem 1.

6) After the $\mathcal{Q}$-learning procedure has converged, the learning phase is terminated. The framework now allows removing the temporary links added to $\mathcal{G}_d$ during the "clique subgraph learning phase" and the stability of the final learned policy will be guaranteed–see Corollary 1.

7) Finally, note that the learned policy for the entire graph will be, in general, a suboptimal solution to the optimization in (3). Yet, we provide guarantees on its sub-optimality gap in Theorem 3 and illustrate its numerical performance in Section V.

The distributed control underpinning of the method proposed in this work follows its model-based analogues studied in [7], [9]. In this work, our contribution is to build on these approaches and propose a model-free structured policy iteration algorithm, which is not only computationally efficient, but also practical for operational large-scaled networked systems.

[4]The proposed Patterned monoid and characterizing its interplay with the Lyapunov equation is considered as a key contribution in our approach. Note that, with reference to [1], $\mathrm{PM}(r \times n, \mathbb{R})$ requires further qualifications to be a linear subgroup of $\mathrm{GL}(rn, \mathbb{R})$.The monoid characterization is sufficient for our purposes and further extensions are deferred to our future work.

B. Related Literature

Distributed control is a well-established area of research in systems theory. The roots of the field trace back to the socioeconomics literature in 1970's [22] and early works in the control literature followed suite later during that decade [23]. The main motivation for these works was lack of scalability in centralized planning and control, due to information or computational limitations [24], [25]. Fast forward a few decades, sufficient graph-theoretic conditions were provided for stability of formations comprised of identical vehicles [26] and, along the same lines, graph-based distributed controller synthesis was further examined independently in works such as [7], [10], [16], [27]. The topic was also studied from the perspective of spatial invariance [15], [28] and a compositional layered design [29], [30]. Moreover, from an agent-level perspective, the problem has been tackled for both homogeneous systems [7], [9], [10] and more recently heterogeneous ones [31].

Having access to the underlying system model is a common assumption in the literature on distributed control, where the goal is to find a distributed feedback mechanism that conforms to an underlying network topology. However, deriving dynamic models from first principles could be restrictive for large-scale systems and complex mission scenarios [32]. Such restrictions also hold for parametric perturbations that occur due to inefficient modeling or other unknown design factors. For instance, even the LQR solution with its strong input robustness properties, may have small stability margins for general parameter perturbations [33]. Robust synthesis approaches could alleviate this issue when the perturbations follow specific models, in both centralized [34] and distributed [35] cases. However, if the original estimates of system parameters are inaccurate or the perturbations violate the presumed model, then both stability and optimality of the proposed feedback mechanisms can be compromised. Data-driven control, on the other hand, circumvents such drawbacks and utilize the available data generated by the system when its model is unavailable. This point of view has historically been examined in the context of adaptive control and system identification [36], particularly, when asymptotic properties of the synthesized system are of interest. For more recent works that have adopted a non-asymptotic outlook on data-driven control, we mention [37]–[39] that used batched data for synthesis, as well as online iterative procedures [40], [41]. Furthermore, in regards to the adaptive nature of such algorithms, there is a close connection between online data-driven control and reinforcement learning [19], [42]. In these latter works, policy iteration has been extended to approximate LQR by avoiding the direct solution of Algebraic Riccati Equation (ARE); yet majority of these works do not have favorable scaling properties.

Control and estimation for large-scale systems offers its unique set of challenges due to higher levels of uncertainty,

scalability issues, and modeling errors. Nevertheless, model-free synthesis for large-scale systems, as a discipline, is still in its infancy. From a control theoretic perspective, the work [43] addresses some of the aforementioned issues using ideas from mean-field multiagent systems and with the key assumption of partial exchangeability. The work [44] on the other hand, provides a decentralized LQR algorithm based on network consensus that has low complexity, but potentially a high cost of implementation. Lastly, SDP projection-based analysis has been examined in [45], where each agent will have a sublinear regret as compared with the best fixed controller in hindsight. The problem has also been considered from a game-theoretic standpoint [46]–[48], where agents can have conflicting objectives.

In the following, we propose our algorithm that iteratively learns necessary control components from a sub-network that would be used to design a distributed controller for the entire network. We also show that depending on the structure of the problem, not only would our scheme be computationally efficient, but also more applicable when model-based control in high dimensions is (practically) infeasible. The structure of our distributed control design is inspired by [7], that was subsequently extended to discrete-time in [9].

III. Main Algorithm

In this section, we present and discuss the main algorithm of the paper, namely, D2SPI. Given the underlying communication graph $\mathcal{G}$, the networked system is considered as a black-box, whereas the designer is capable of injecting input signals to the system and observe the corresponding states. The goal of D2SPI is then to find a data-guided suboptimal solution for (3) without knowledge of system parameters $\hat{\mathbf{A}}$ and $\hat{\mathbf{B}}$. To this end, our approach involves considering the synthesis problem on a subgraph $\mathcal{G}_d \subseteq \mathcal{G}$, with the associated time-series data. Before presenting the main algorithm, we formalize two useful notions in order to facilitate the presentation.

Definition 1: Given a subgraph $\mathcal{G}' \subseteq \mathcal{G}$ and a node labeling, let $\text{Policy}\,(\mathcal{V}_{\mathcal{G}'})$ denote the concatenation of policies of the agents in $\mathcal{V}_{\mathcal{G}'}$, i.e., $\text{Policy}\,(\mathcal{V}_{\mathcal{G}'}) \coloneqq [\boldsymbol{u}_1^\intercal \;\; \boldsymbol{u}_2^\intercal \;\; \cdots \;\; \boldsymbol{u}_{|\mathcal{V}_{\mathcal{G}'}|}^\intercal]^\intercal$, where $\boldsymbol{u}_i$ is the feedback control policy of agent i in the subgraph $\mathcal{G}'$ as a mapping from $\{\boldsymbol{x}_j | j \in \mathcal{N}_i \cup \{i\}\}$ to $\mathbb{R}^m$. Furthermore, we use $\text{Policy}(\mathcal{V}_{\mathcal{G}'})|_t$ to denote the realization of these policies at time t. Similarly, we define, $\text{State}\,(\mathcal{V}_{\mathcal{G}'}) \coloneqq [\boldsymbol{x}_1^\intercal \;\; \boldsymbol{x}_2^\intercal \;\; \cdots \;\; \boldsymbol{x}_{|\mathcal{V}_{\mathcal{G}'}|}^\intercal]^\intercal$.

The D2SPI algorithm is introduced in Algorithm 1 with the following standard assumption.

Assumption 1: The initial controller K_1 is stabilizing for the controllable pair (A, B), and $\boldsymbol{e}_t$ in Algorithm 2 is such that $\text{Policy}(\mathcal{V}_{\mathcal{G}'})|_t$ remains persistently exciting (PE); more precisely, if we collect the state and input difference signals in the vector form ϕ_t as defined in Algorithm 2, then the PE condition requires that [20]: there exists a positive integer N_0 and positive constants $\epsilon_0 \leq \bar{\epsilon}_0$ such that

$$\epsilon_0 I \leq \frac{1}{N}\sum_{i=1}^{N} \phi_{t-i}\phi_{t-i}^\intercal \leq \bar{\epsilon}_0 I, \text{ for all } t \geq N_0 \text{ and } N \geq N_0.$$

Remark 1: Note that the controllability of the pair (A, B) and the PE condition are sufficient assumptions to guarantee well-posedness of the data-driven control problem that are commonly adopted in literature [49], [50]. This PE condition is an adaptation of "strong persistent excitation" in [49] to the least-squares problem of recovering unknown parameters of the so-called $\mathcal{Q}$-function. The PE condition can be easily satisfied in our setup by ensuring a rich randomness in the signal $\boldsymbol{e}_t$, such as a non-degenerate Guassian distribution. A more modern treatment of PE condition is stated by a rank condition on the input Hankel matrices in the context of Willems' Fundamental Lemma [50], [51]. We also refer to [51] for a discussion on how these assumptions can be relaxed. Furthermore, assuming initial stabilizing controller K_1 is also common in the policy iteration literature. For instance, in the case of open loop stable system A, K_1 is simply chosen to be zero. For an unknown and unstable pair (A, B), more elaborate online stabilization techniques have been adopted that are out of the scope of this work. Refer to [52] for one such method.

1) The Learning Phase: We refer to the main loop of the algorithm in Line 8 as the *learning phase*. During the learning phase, we include temporary "auxiliary" links to $\mathcal{G}_d$ and make the communication graph a clique. We show such distinction by $\mathcal{G}_{d,\text{learn}}$, where $|\mathcal{V}_{\mathcal{G}_d}| = \left|\mathcal{V}_{\mathcal{G}_{d,\text{learn}}}\right|$ but $\mathcal{G}_{d,\text{learn}}$ is a clique. Inherent to D2SPI is a policy iteration on $\mathcal{G}_{d,\text{learn}}$ that characterizes components K_k and L_k, intuitively representing "self" and "cooperative" controls at iteration k, respectively. In particular, during the learning phase, we utilize these control components in order to design and update an effective stabilizing controller for the rest of the network $\mathcal{G} \setminus \mathcal{G}_{d,\text{learn}}$.

We do so by ensuring that during the learning phase, information is exchanged uni-directionally from $\mathcal{G}_{d,\text{learn}}$ to the rest of the network; hence, the policy of the agents in $\mathcal{G} \setminus \mathcal{G}_{d,\text{learn}}$ is dependent on those in $\mathcal{G}_{d,\text{learn}}$, and not vice versa. After the learning phase terminates, we remove the temporary links added during the learning phase (re-initialize to the original network topology), and synthesize a suboptimal stabilizing control for the entire network. In the learning phase of D2SPI, we use a Recursive Least-Squares (RLS)-based recursion to estimate the unknown parameters of the $\mathcal{Q}$-function at iteration k, referred to as $\widetilde{\mathbf{H}}_k$.

2) The Subgraph Policy Evaluation (SPE) Subroutine: This process is performed in SPE (Algorithm 2) subroutine by inputting (sub-)graph $\mathcal{G}$, $\mathcal{G}_{d,\text{learn}}$, the mapping $\text{policy}(\mathcal{V}_{\mathcal{G}_d})$, and the previous estimate of $\widetilde{\mathbf{H}}_{k-1}$. As will be discussed in Section I-A, $\widetilde{\mathbf{H}}_k$ contains the required information to determine the two control components K_k and L_k from data. We extract this square matrix through a recursive update on the vector θ_{k-1}, derived from half-vectorization of $\widetilde{\mathbf{H}}_{k-1}$, solving RLS for the linear equation $\mathcal{R}(\tilde{\mathbf{x}}_t, \tilde{\mathbf{u}}_t) = \zeta_t^\intercal \theta_{k-1}$, where $\mathcal{R}(\tilde{\mathbf{x}}_t, \tilde{\mathbf{u}}_t)$ denotes the local cost and $\zeta_t \in \mathbb{R}^p$ contains the data measurements. We use subscript k for policy update and t for data collection. The adaptive nature of the algorithm involves the exploration signal $\boldsymbol{e}_t$ to be augmented to the policy vector in order to provide persistence of excitation.

3) Persistence of Excitation and Convergence of SPE: In our setup, $\boldsymbol{e}_t$ is sampled from a normal distribution $\boldsymbol{e} \sim \mathcal{N}(\mathbf{0}, \Sigma)$ where the choice of the variance $\Sigma \succ 0$ is problem-specific. In practice, excitation of the input is a subtle task and has

Algorithm 1 Data-driven Structured Policy Iteration (D2SPI)

1: **Initialization** $(t \leftarrow 1,\ k \leftarrow 1)$
2: Choose $\mathcal{G}_d \subseteq \mathcal{G}$ with $d = d_{\max}(\mathcal{G}) + 1$
3: Obtain Q_1, Q_2, R, and set $Q_d \leftarrow Q_1 + dQ_2$
4: Get $\tilde{\mathbf{x}}_1 \in \mathbb{R}^{dn}$ and $\widetilde{\mathbf{H}}_0 \in \mathbb{R}^{p\times p}$, $p = d(n+m)$
5: Set $\mathcal{P}_0 \leftarrow \beta \mathrm{I}_{p(p+1)/2}$ for large $\beta > 0$ and fix Σ
6: Set K_1 stabilizing (1), $L_1 = \mathbf{0}$, $\Delta K_1 = K_1$
7: Turn on temporary links in $\mathcal{G}_d$ and set $\tau_1 \leftarrow 0$
8: **While** (K_k, L_k) **has not converged, do** ("learning phase")
9: Set $\mathrm{Policy}_k(\mathcal{V}_\mathcal{G})$ such that for each $i \in \mathcal{V}_\mathcal{G}$,

$$\begin{aligned} \boldsymbol{u}_i &\leftarrow \Delta K_k \boldsymbol{x}_i + L_k \textstyle\sum_{j\in\mathcal{N}_i} \frac{\tau_k}{d-1}\boldsymbol{x}_j, && \text{if } i \in \mathcal{V}_{\mathcal{G}\setminus\mathcal{G}_d} \\ \boldsymbol{u}_i &\leftarrow \Delta K_k \boldsymbol{x}_i + L_k \textstyle\sum_{j\in\mathcal{V}_{\mathcal{G}_d}} \boldsymbol{x}_j, && \text{if } i \in \mathcal{V}_{\mathcal{G}_d} \end{aligned}$$

10: Evaluate $\widetilde{\mathbf{H}}_k$ from Algorithm 2
$\widetilde{\mathbf{H}}_k \leftarrow \mathrm{SPE}\big(\mathcal{G},\ \mathcal{G}_{d,\mathrm{learn}},\ \mathrm{Policy}_k(\mathcal{V}_\mathcal{G}),\ \widetilde{\mathbf{H}}_{k-1}, \mathcal{P}_0\big)$
11: Recover X_1, X_2, Y_1, Y_2, Z_1 and Z_2 from $\widetilde{\mathbf{H}}_k$

$$\begin{aligned} X_1 &\leftarrow \widetilde{\mathbf{H}}_k[1:n,\ 1:n] \\ Y_1 &\leftarrow \widetilde{\mathbf{H}}_k[dn+1:dn+m,\ dn+1:dn+m] \\ Z_1 &\leftarrow \widetilde{\mathbf{H}}_k[dn+1:dn+m,\ 1:n] \\ X_2 &\leftarrow \widetilde{\mathbf{H}}_k[1:n,\ n+1:2n] \\ Y_2 &\leftarrow \widetilde{\mathbf{H}}_k[dn+1:dn+m,\ dn+m+1:dn+2m+1] \\ Z_2 &\leftarrow \widetilde{\mathbf{H}}_k[dn+1:dn+m,\ n+1:2n] \\ \Delta X \leftarrow X_1 - X_2,&\quad \Delta Y \leftarrow Y_1 - Y_2,\quad \Delta Z \leftarrow Z_1 - Z_2. \end{aligned}$$

12: Update the control components

$$\begin{aligned} F^{-1} &\leftarrow Y_1 - (d-1)Y_2\big(Y_1 + (d-2)Y_2\big)^{-1}Y_2 \\ G &\leftarrow \big(Y_1 + (d-1)Y_2\big)^{-1} Y_2 \left(Y_1 - Y_2\right)^{-1} \\ K_{k+1} &\leftarrow -FZ_1 + (d-1)GZ_2, \\ L_{k+1} &\leftarrow -FZ_2 + GZ_1 + (d-2)GZ_2, \\ \Delta K_{k+1} &\leftarrow K_{k+1} - L_{k+1} \end{aligned}$$

13: Obtain the stability margin

$$\begin{aligned} \Xi_{k+1} \leftarrow& \Delta X - Q_d + \Delta K_{k+1}^\intercal \Delta Z \\ &+ \Delta Z^\intercal \Delta K_{k+1} + \Delta K_{k+1}^\intercal (\Delta Y - R)\Delta K_{k+1} \\ \gamma_{k+1} \leftarrow& \lambda_{\min}\Big(\Delta K_{k+1}^\intercal R \Delta K_{k+1} + Q_d\Big) \\ &/\lambda_{\max}\Big(\Xi_{k+1} + L_{k+1}^\intercal(\Delta Y - R)L_{k+1}\Big) \\ \tau_{k+1} \leftarrow& \sqrt{\gamma_{k+1}^2/(1+\gamma_{k+1})} \end{aligned}$$

14: Go to Line 8 and set $k \leftarrow k+1$
15: Switch OFF the temporary links and retrieve $\mathcal{G}_d$
16: Set $\mathrm{Policy}_k(\mathcal{V}_\mathcal{G})$ such that for each $i \in \mathcal{V}_\mathcal{G}$,
$\boldsymbol{u}_i \leftarrow \Delta K_k \boldsymbol{x}_i + \frac{\tau_k}{d-1} L_k \sum_{j\in\mathcal{N}_i} \boldsymbol{x}_j$

been realized in a variety of forms such as random noise [19], sinusoidal signals [53], and exponentially decaying noise [54]. We denote by $\mathcal{P}$ the projection factor that is reset to $\mathcal{P}_0 \succ 0$ for each iteration. Convergence of SPE—guaranteed based on the persistence of excitation condition—is followed by the update of $\widetilde{\mathbf{H}}_k$ that encodes the necessary information to obtain K_k and L_k.

4) Learning Control Components using Patterned Monoid: Learning the control components K_k and L_k is achieved by first recovering the block matrices X_1, X_2, Y_1, Y_2, Z_1, and Z_2 from $\widetilde{\mathbf{H}}_k$ that are further utilized to form intermediate variables F and G. Matrix inversions on line 12 of Algorithm 1 will be justified in Section IV Lemma 2. Such recovery of meaningful blocks from $\widetilde{\mathbf{H}}_k$ is due to the specific structure resulting from adding extra links to $\mathcal{G}_d$ that is captured systematically by the Patterned monoid; this point will be discussed further subsequently in Proposition 2.

5) Learning Gain Margin and the Distributed Feedback Design: Each iteration loop is completed by updating the parameters γ_k and τ_k, that prove instrumental in the stability analysis of the proposed distributed controller for the entire network. Finally, with the convergence of D2SPI, $\mathcal{G}_d$ is retrieved by removing the temporary links and the structured policy is extended to the entire graph $\mathcal{G}$ in Line 16.

Let us point out a few remarks on the computational complexity of the proposed algorithm. First, note that the inverse operations on line 12 occur on matrices of size $m \times m$, and hence computationally inexpensive. Furthermore, the complexity of finding extreme singular values—as on line 13 in Algorithm 1—is known to be $\mathcal{O}(n^2)$ [55]. Hence, the computational complexity of D2SPI is mainly due to the SPE recursion that is equivalent to the complexity of RLS for the number of unknown system parameters in $\mathcal{G}_d$, i.e., the computational cost is $\mathcal{O}\left(d^2(n+m)^2\right)$ [56]. This implies that the computational complexity of the algorithm is *fixed* for any number of agents N, as long as the maximum degree of the graph retains its order.

Remark 2: Adding temporary links within the subgraph $\mathcal{G}_d$ is an effective means of learning optimal K_k and L_k for the subgraph $\mathcal{G}_{d,\mathrm{learn}}$ by utilizing dynamical interdependencies among the agents. Although initializing K_k such that (1) is Schur stable is a standard assumption in data-driven control, obtaining this initial gain for an unknown system is nontrivial. While we invoke this assumption in this work, the interested reader is referred to [40], [57] for more recent works pertaining to this assumption and related system theoretic intricacies [51].

IV. Convergence and Stability Analyses

In this section, we provide convergence and stability analyses for the D2SPI algorithm. First, we study the structure and stability margins of each local controller and proceed to establish stability properties of the proposed controller for the entire network throughout the learning process. Lastly, we show the convergence of D2SPI to a stabilizing suboptimal distributed controller followed by the derivation of a suboptimality bound characterized by the problem parameters. For clarity, we defer some of the subtleties of the analysis and detailed proofs to Appendix I.

First, let us demonstrate how a specific structure and stability of the controller for the subgraph $\mathcal{G}_{d,\mathrm{learn}}$, when properly initialized, can be preserved throughout the D2SPI algorithm.

Proposition 2: Let $\widetilde{\mathbf{K}}_k := \mathrm{I}_d \otimes \left(K_k - L_k\right) + \mathbb{1}\mathbb{1}^\intercal \otimes L_k$ for all $k \geq 1$, with K_k and L_k as in Algorithm 1. Under Assumption 1 and throughout the learning phase (for all $k \geq 1$), $\widetilde{\mathbf{K}}_k$ is stabilizing for the system in $\mathcal{G}_{d,\mathrm{learn}}$ and $\mathrm{Policy}_k(\mathcal{V}_{\mathcal{G}_{d,\mathrm{learn}}})|_t = \widetilde{\mathbf{K}}_k\, \mathrm{State}(\mathcal{V}_{\mathcal{G}_{d,\mathrm{learn}}})|_t$, for all t. Furthermore, $\Delta K_k := K_k - L_k$ stabilizes the dynamics of a single agent, i.e., $A + B\Delta K_k$ is Schur stable.

Note that Proposition 2 proves the existence of a stabilizing controller ΔK_k and its corresponding cost-to-go matrix ΔP_k.

In the sequel, our goal is to design a distributed suboptimal controller for the entire networked system on $\mathcal{G}$ based on the components that shape ΔK_k. This extension is built upon the stability margin derived next.

Proposition 3: At each iteration $k \geq 1$, let K_k, L_k and τ_k be obtained via Algorithm 1. Then, $A+B(K_k-\alpha L_k)$ is Schur stable for all α satisfying $|\alpha - 1| \leq \tau_k$.

The stability margin τ_k in Proposition 3 is upper-bounded by the stability margin of the pair $(A+B(K_k-L_k), B)$. This implies that if the original closed-loop system for an agent does not have a favorable stability margin, then τ_k can be small–reducing the influence of the agent's neighbors on its policy (Line 16 of Algorithm 1). Nonetheless, Proposition 3 provides model-free stability gain margins τ_k at each iteration of the algorithm for the dynamics of a single agent in $\mathcal{G}$. In our analysis, we take advantage of these margins to characterize stability guarantees for the controller proposed during the learning phase. This is captured in the following result.

Theorem 1: Suppose K_k, L_k and τ_k are defined as in Algorithm 1. Then, under Assumption 1, the control policy $\text{Policy}_k(\mathcal{V}_\mathcal{G})$ designed during the learning phase (line 9) stabilizes the network $\mathcal{G}$ at each iteration of the learning phase and for any choice of $\mathcal{V}_{\mathcal{G}_d}$.

Theorem 1 establishes that the proposed feedback mechanism stabilizes the entire network, facilitating control of agents *outside* of $\mathcal{G}_d$, during the learning phase. In the meantime, the practicality and suboptimality of the algorithm depend on its convergence addressed next.

Theorem 2: Under Assumption 1 and (long enough) finite termination of Algorithm 2, Algorithm 1 converges, i.e., $K_k \to K^*$, $L_k \to L^*$, and $\tau_k \to \tau^*$ as $k \to \infty$, where $\widetilde{\mathbf{K}}^* = \mathrm{I}_d \otimes \big(K^* - L^*\big) + \mathbb{1}\mathbb{1}^\intercal \otimes L^*$ is the optimal solution to the infinite-horizon state-feedback LQR problem with system parameters $\big(\widetilde{\mathbf{A}}, \widetilde{\mathbf{B}}, \widetilde{\mathbf{Q}}, \widetilde{\mathbf{R}}\big)$ defined as $\widetilde{\mathbf{A}} = \mathrm{I}_d \otimes A$, $\widetilde{\mathbf{B}} = \mathrm{I}_d \otimes B$, $\widetilde{\mathbf{Q}} = \mathrm{I}_d \otimes (Q_1 + dQ_2) - \mathbb{1}\mathbb{1}^\intercal \otimes Q_2$, and $\widetilde{\mathbf{R}} = \mathrm{I}_d \otimes R$.

Finally, we note that as the temporary links introduced during the learning phase are removed, the structure of the agents' interaction is once again the original network $\mathcal{G}$. As such, it is vital to provide stability guarantees after Algorithm 1 terminates and components of the control design have converged. This issue is addressed in the following corollary whose proof is similar to that of Theorem 1 and thus omitted.

Corollary 1: Suppose that K^*, L^*, γ^*, and τ^* are given as in Theorem 2 under a convergent Algorithm 1. Then $\text{Policy}(\mathcal{V}_\mathcal{G})$ (defined on Line 16), stabilizes the entire networked system in (2).

We conclude this section by exploring the suboptimality of the proposed policy. Given the problem parameters, let $\hat{\mathbf{K}}^*_{\text{struc}}$ denote the globally optimal distributed solution for the structured LQR problem in (3) with the associated cost matrix $\hat{\mathbf{P}}^*_{\text{struc}}$. Given any other stabilizing structured policy $\hat{\mathbf{K}}$ associated with cost matrix $\hat{\mathbf{P}}$, we define the *optimality gap* as $\text{gap}(\hat{\mathbf{K}}) := \text{tr}[\hat{\mathbf{P}} - \hat{\mathbf{P}}^*_{\text{struc}}]$. The following theorem provides an upperbound on the optimality gap of structured policy learned by D2SPI based on the problem parameters. In particular, when the system is "contractible," the derived upperbound depends on the difference of the distributed controller with that of *unstructured* optimal LQR controller.

Algorithm 2 Subgraph Policy Evaluation (SPE)

1: **Input:** Graph $\mathcal{G}$, subgraph $\mathcal{G}' \subseteq \mathcal{G}$, $\text{Policy}(\mathcal{V}_\mathcal{G})$, $\mathbf{H}$, $\mathcal{P}$
2: **Output:** Updated cost matrix $\mathbf{H}^+$ associated with $\mathcal{G}'$
3: **While** $\mathbf{H}$ **has not converged, do**
4: Set $\tilde{\mathbf{x}}_t \leftarrow \text{State}(\mathcal{V}_{\mathcal{G}'})|_t$ and $\tilde{\mathbf{u}}_t \leftarrow \text{Policy}(\mathcal{V}_{\mathcal{G}'})|_t$
5: Choose $\boldsymbol{e}_t \sim \mathcal{N}(\mathbf{0}, \Sigma)$ and update $\text{Policy}(\mathcal{V}_{\mathcal{G}'})|_t$ as $\tilde{\mathbf{u}}_t \leftarrow \tilde{\mathbf{u}}_t + \boldsymbol{e}_t$ for all $i \in \mathcal{V}_{\mathcal{G}'}$
6: Run $\mathcal{G}$ under policy $\text{Policy}(\mathcal{V}_\mathcal{G})$
7: Collect $\text{State}(\mathcal{V}_{\mathcal{G}'})|_{t+1}$ only from $\mathcal{G}'$ and set $\tilde{\mathbf{x}}_{t+1} \leftarrow \text{State}(\mathcal{V}_{\mathcal{G}'})|_{t+1}$, $\tilde{\mathbf{u}}_{t+1} \leftarrow \text{Policy}(\mathcal{V}_{\mathcal{G}'})|_{t+1}$
8: Set $\phi_t \leftarrow [\tilde{\mathbf{x}}_t^\intercal \ \tilde{\mathbf{u}}_t^\intercal]^\intercal - [\tilde{\mathbf{x}}_{t+1}^\intercal \ \tilde{\mathbf{u}}_{t+1}^\intercal]^\intercal$
9: Compute $\zeta_t = \text{vech}(\phi_t \phi_t^\intercal)$ and $\mathcal{R}(\tilde{\mathbf{x}}_t, \tilde{\mathbf{u}}_t) = \tilde{\mathbf{x}}_t^\intercal \big(\mathrm{I} \otimes Q_d - \mathbb{1}\mathbb{1}^\intercal \otimes Q_2\big)\tilde{\mathbf{x}}_t + \tilde{\mathbf{u}}_t^\intercal \big(\mathrm{I} \otimes R\big)\tilde{\mathbf{u}}_t$
10: Set $\theta \leftarrow \text{vech}(\mathbf{H})$ and update $\theta \leftarrow \theta + \mathcal{P}\zeta_t \big(\mathcal{R}(\tilde{\mathbf{x}}_t, \tilde{\mathbf{u}}_t) - \zeta_t^\intercal \theta\big) / (1 + \zeta_t^\intercal \mathcal{P} \zeta_t)$, $\mathcal{P} \leftarrow \mathcal{P} - \mathcal{P}\zeta_t\zeta_t^\intercal \mathcal{P} / (1 + \zeta_t^\intercal \mathcal{P} \zeta_t)$
11: Find $\mathbf{H}^+ = \text{vech}^{-1}(\theta)$, update $\mathbf{H} \leftarrow \mathbf{H}^+$ and $t \leftarrow t+1$

Theorem 3: Let $\hat{\mathbf{K}}^*$ be the structured policy learned by Algorithm 1 at convergence, corresponding to the cost matrix $\hat{\mathbf{P}}^*$. Moreover, let $\hat{\mathbf{K}}_{\text{lqr}}$ denote the optimal (unstructured) solution to the infinite-horizon state-feedback LQR problem with parameters $(\hat{\mathbf{A}}, \hat{\mathbf{B}}, \hat{\mathbf{Q}}, \hat{\mathbf{R}})$ with the cost matrix $\hat{\mathbf{P}}_{\text{lqr}}$. If $\hat{\mathbf{A}}_{\hat{\mathbf{K}}_{\text{lqr}}} = \hat{\mathbf{A}} + \hat{\mathbf{B}}\hat{\mathbf{K}}_{\text{lqr}}$ is contractible then $0 \leq \text{gap}(\hat{\mathbf{K}}^*) \leq \text{tr}(\mathbf{M})/[1 - \sigma^2_{\max}(\hat{\mathbf{A}}_{\hat{\mathbf{K}}_{\text{lqr}}})]$, where $\mathbf{M} := (\hat{\mathbf{R}} + \hat{\mathbf{B}}^\intercal \hat{\mathbf{P}}^* \hat{\mathbf{B}})(\hat{\mathbf{K}}^* \hat{\mathbf{K}}^{*\intercal} - \hat{\mathbf{K}}_{\text{lqr}} \hat{\mathbf{K}}_{\text{lqr}}^\intercal) + 2\hat{\mathbf{A}}^\intercal \hat{\mathbf{P}}^* \hat{\mathbf{B}}(\hat{\mathbf{K}}^* - \hat{\mathbf{K}}_{\text{lqr}})$.

Remark 3: First, recall how the converged policy by Theorem 2 is related to the optimal Linear Quadratic Regulator (LQR) policy on the fully connected subgraph $\mathcal{G}_{d,\text{learn}}$. Second, the optimality gap is characterized by $\text{tr}(\mathbf{M})$ which is essentially proportional to the difference $\hat{\mathbf{K}}^* - \hat{\mathbf{K}}_{\text{lqr}}$; i.e., how close our designed structured policy is to the unstructured LQR that directly depends on the connectivity of the graph topology. Third, the contractibility of the pair (A, B) is more restrictive condition than regularizability of the system [40], a notion that has recently been employed in iterative data-guided control methods [58], [59]. Contractibility also facilitates the validity of assuming access to the initial stabilizing controller.

V. Simulation Examples

In this section, we examine the performance and convergence of D2SPI. In order to assess the suboptimality of the synthesized controllers, we report the trace of cost matrices, $\text{tr}(\hat{\mathbf{P}}_k)$, associated with the proposed distributed controller learned by D2SPI at iteration k. As the optimal distributed design is unknown, we compare these results against the optimal cost for the unconstrained LQR problem, $\text{tr}(\hat{\mathbf{P}}_{\text{LQR}})$, obtained via the solution of the Algebraic Riccati Equation with parameters $(\hat{\mathbf{A}}, \hat{\mathbf{B}}, \hat{\mathbf{Q}}, \hat{\mathbf{R}})$. Note that this is an infeasible solution to the problem in (3); nevertheless, it provides a theoretical lowerbound to evaluate the performance of any feasible solution—including the optimal one.[5]

A. Convergence–Randomly Selected Parameters

In the first example, we sample continuous-time system parameters of a single agent (A, B) from a zero-mean normal

[5]All the simulations were run on a 3.2 GHz Quad-Core Intel Core i5 CPU and in MATLAB. The scripts are publicly available at https://github.com/shahriarta/D2SPI.

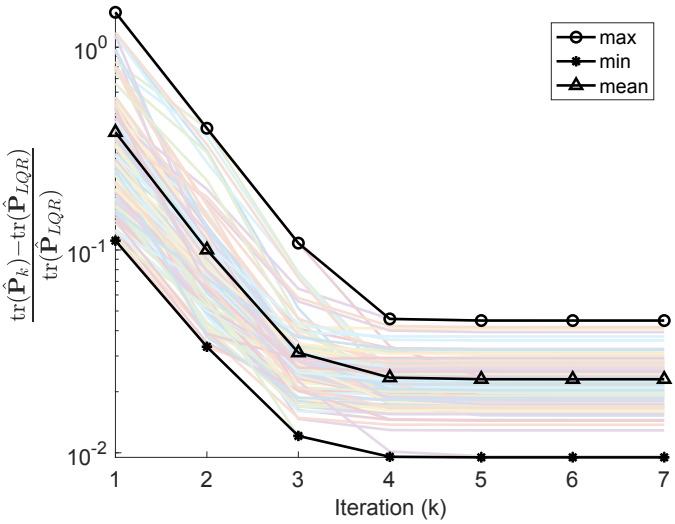


Fig. 2: Convergence of Algorithm 1 for 100 randomly sampled system parameters (A, B).

distribution with unit covariance, such that $A \in \mathbb{R}^{5\times5}$ and $B \in \mathbb{R}^{5\times3}$. We then consider a path-graph of 10 agents and demonstrate how Algorithm 1 converges for this network using different instances of the system parameters. The continuous-time system dynamics of a single agent is discretized with a sampling rate of $\Delta T = 0.1s$. We set $Q_1 = 0.2\mathrm{I}_5$, $Q_2 = \mathrm{I}_5$ and $R = \mathrm{I}_3$ in (3). We assume a random exploration signal sampled from a normal distribution $e_t \sim \mathcal{N}(0, \sigma^2)$, where the variance σ^2 is chosen accordingly for different input channels.

Figure 2 shows the performance of the synthesized controller by illustrating the normalized suboptimality error for the entire network with respect to the (infeasible) LQR controller. This figure depicts the simulation results for 100 random system parameters and shows the progress of the proposed method at each iteration in Algorithm 1. The actual simulations are plotted in faded color, whereas their statistics are plotted in solid black. Note that almost all realizations of the network have converged after 5 iterations of the learning phase, due to its fast quadratic convergence rate. Assessing the suboptimality of the proposed controller compared against the (infeasible) centralized LQR controller, reveals an average improvement by a factor of 200.

B. Network of homogeneous plants

In this example, we apply D2SPI to two other simulation scenarios involving homogeneous networks of agents with unknown and unstructured model uncertainties. In particular, we use the dynamics of plants with continuous-time system parameters (A, B) (as reported in Appendix F of [60]), in conjunction with random d-regular graph topologies of different sizes. We then examine the efficacy of D2SPI by illustrating the cost associated with the proposed distributed controller as a function of nodes in the graph. The rest of the problem parameters are chosen identical to the setup in Section V-A. Uncertainty in the model are introduced in this example as follows. Each agent follows an unknown LTI dynamics similar to (1) with A replaced by $A + \Delta A$, where entries of ΔA are sampled from a normal distribution $\mathcal{N}(0, 0.05)$. Assuming that one has access to a stabilizing controller K_1 for the system with nominal parameters A and B, we set the initial stabilizing controller to be the LQR optimal controller with parameters (A, B, Q_1, R).

Figure 3 shows the results of the second simulation example, illustrating how the cost of the proposed controller changes with respect to the number of nodes in a path graph with different number of nodes. Figure 3a compares the cost associated with our design $\hat{\mathbf{P}}_\infty$ against the cost of the initial controller $\hat{\mathbf{P}}_1$, and the (infeasible) LQR controller $\hat{\mathbf{P}}_{\mathrm{LQR}}$. Figure 3b illustrates the evolution of the normalized suboptimality of our proposed algorithm (with respect to the infeasible LQR controller) as a function of number of nodes in the corresponding graphs. Figure 3c and Figure 3d show similar results for the random 3-regular graph topologies with even number of nodes.

As it can be validated from Figure 3, the cost associated with our final proposed distributed controller has significantly improved the optimality of the initial controller. In particular, the normalized suboptimality errors of our final design are less than 6% and 2.4% for path-graph and random 3-regular graph topologies, respectively. Furthermore, this normalized error generally decrease as the number of nodes in the corresponding graphs increase.

VI. Concluding Remarks

In this paper, we have proposed the D2SPI algorithm as an efficient model-free distributed control synthesis process for potentially high-dimensional network of homogeneous linear systems. D2SPI is built upon a construction referred to as Patterned monoid, that facilitates exploiting network symmetries for policy synthesis consistent with the underlying network. Such symmetries allow a data collection procedure during the learning phase (with temporary additional links) for a smaller portion of the network. Using data collection on this smaller subnetwork, we are then able to synthesize a distributed feedback mechanism for the entire system–even during the learning phase. Moreover, D2SPI builds upon parameter estimation techniques that represent an end-to-end policy prediction directly from the observed data. Extension of the setup proposed for D2SPI involves heterogeneous networked systems, that is currently being examined as part of our future work.

Appendix I
Analysis and Proofs

In this section, we provide the building blocks needed for the proofs and analysis of our algorithm. We first provide some insights on how the setup is connected to the classic model-based LQR machinery and some previously established results that we leverage from the literature. The main proofs then follows.

A. Underlying Model of the Subsystem $\mathcal{G}_\mathrm{d}$

The configuration of the synthesis problem in D2SPI intertwines an online recursion on the subsystem corresponding to $\mathcal{G}_d$ and the original system $\mathcal{G}$. In particular, during the learning phase, considering the same cost structure and

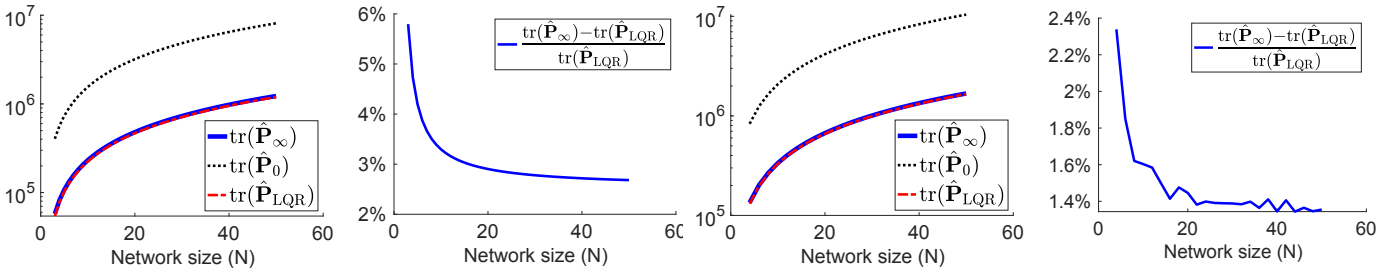


Fig. 3: Suboptimality of the distributed controller learned by Algorithm 1 for a network of N homogeneous plants with path-graphs structures in (a) and (b), and with random 3-regular graphs structures in (c) and (d).

problem parameters as in (3)—but for the completed subgraph $\mathcal{G}_{d,\mathrm{learn}}$— results in the $(\widetilde{\mathbf{A}}, \widetilde{\mathbf{B}}, \widetilde{\mathbf{Q}}, \widetilde{\mathbf{R}})$ parameters as defined in Theorem 2. Then, similarly from (1) the dynamics of the subgraph $\mathcal{G}_{d,\mathrm{learn}}$ assumes the form,

$$\tilde{\mathbf{x}}_{t+1} = \widetilde{\mathbf{A}}\tilde{\mathbf{x}}_t + \widetilde{\mathbf{B}}\tilde{\mathbf{u}}_t, \tag{4}$$

where $\tilde{\mathbf{x}}_t$ and $\tilde{\mathbf{u}}_t$ are formed from concatenation of state and control signals in $\mathcal{G}_{d,\mathrm{learn}}$—recall that $\tilde{\mathbf{u}}_t$ is also denoted by $\mathrm{Policy}(\mathcal{V}_{\mathcal{G}_{d,\mathrm{learn}}})|_t$ in the algorithm to emphasize the temporal implementation of a specific policy in Algorithm 2. From the Bellman equation [61] for the LQR problem with these parameters, the cost matrix $\widetilde{\mathbf{P}}_k$ of $\mathcal{G}_{d,\mathrm{learn}}$ is correlated with the one-step LQR cost as,

$$\tilde{\mathbf{x}}_t^\intercal \widetilde{\mathbf{P}}_k \tilde{\mathbf{x}}_t = \mathcal{R}(\tilde{\mathbf{x}}_t, \tilde{\mathbf{u}}_t) + \tilde{\mathbf{x}}_{t+1}^\intercal \widetilde{\mathbf{P}}_k \tilde{\mathbf{x}}_{t+1}, \tag{5}$$

where $\mathcal{R}(\tilde{\mathbf{x}}_t, \tilde{\mathbf{u}}_t) = \tilde{\mathbf{x}}_t^\intercal \widetilde{\mathbf{Q}} \tilde{\mathbf{x}}_t + \tilde{\mathbf{u}}_t^\intercal \widetilde{\mathbf{R}} \tilde{\mathbf{u}}_t$ and $\widetilde{\mathbf{P}}_k$ satisfies the Lyapunov equation,

$$\widetilde{\mathbf{P}}_k = \big(\widetilde{\mathbf{A}} + \widetilde{\mathbf{B}}\widetilde{\mathbf{K}}_k\big)^\intercal \widetilde{\mathbf{P}}_k \big(\widetilde{\mathbf{A}} + \widetilde{\mathbf{B}}\widetilde{\mathbf{K}}_k\big) + \widetilde{\mathbf{Q}} + \widetilde{\mathbf{K}}_k^\intercal \widetilde{\mathbf{R}} \widetilde{\mathbf{K}}_k, \tag{6}$$

and $\widetilde{\mathbf{K}}_k$ is the controller policy at iteration k. The dynamic programming solution to the LQR problem suggests a linear feedback form $\tilde{\mathbf{u}}_t = \widetilde{\mathbf{K}}_k \tilde{\mathbf{x}}_t$ for the subsystem $\mathcal{G}_{d,\mathrm{learn}}$ at each iteration. Combining (4) and (5) with some rearrangements result in

$$\begin{aligned}\tilde{\mathbf{x}}_t^\intercal \widetilde{\mathbf{P}}_k \tilde{\mathbf{x}}_t &= \tilde{\mathbf{z}}_t^\intercal \begin{bmatrix} \widetilde{\mathbf{Q}} + \widetilde{\mathbf{A}}^\intercal \widetilde{\mathbf{P}}_k \widetilde{\mathbf{A}} & \widetilde{\mathbf{A}}^\intercal \widetilde{\mathbf{P}}_k \widetilde{\mathbf{B}} \\ \widetilde{\mathbf{B}}^\intercal \widetilde{\mathbf{P}}_k \widetilde{\mathbf{A}} & \widetilde{\mathbf{R}} + \widetilde{\mathbf{B}}^\intercal \widetilde{\mathbf{P}}_k \widetilde{\mathbf{B}} \end{bmatrix} \tilde{\mathbf{z}}_t \\ &=: \tilde{\mathbf{z}}_t^\intercal \begin{bmatrix} [\widetilde{\mathbf{H}}_k]_{11} & [\widetilde{\mathbf{H}}_k]_{12} \\ [\widetilde{\mathbf{H}}_k]_{21} & [\widetilde{\mathbf{H}}_k]_{22} \end{bmatrix} \tilde{\mathbf{z}}_t = \tilde{\mathbf{z}}_t^\intercal \widetilde{\mathbf{H}}_k \tilde{\mathbf{z}}_t, \end{aligned} \tag{7}$$

where $\tilde{\mathbf{z}}_t = [\tilde{\mathbf{x}}_t^\intercal \ \tilde{\mathbf{u}}_t^\intercal]^\intercal$. Then, the following policy update (due to Hewer) is guaranteed to converge to the optimal LQR policy under controllability assumption [21]:

$$\widetilde{\mathbf{K}}_{k+1} = -\left(\widetilde{\mathbf{R}} + \widetilde{\mathbf{B}}^\intercal \widetilde{\mathbf{P}}_k \widetilde{\mathbf{B}}\right)^{-1} \widetilde{\mathbf{B}}^\intercal \widetilde{\mathbf{P}}_k \widetilde{\mathbf{A}} = -[\widetilde{\mathbf{H}}_k]_{22}^{-1} [\widetilde{\mathbf{H}}_k]_{21}, \tag{8}$$

which is also reconstructed by information in $\widetilde{\mathbf{H}}_k$. Furthermore, the cost matrix in (6) can also be reconstructed by the same information:

$$\widetilde{\mathbf{P}}_k = [\widetilde{\mathbf{H}}_k]_{11} + [\widetilde{\mathbf{H}}_k]_{12}\widetilde{\mathbf{K}}_k + \widetilde{\mathbf{K}}_k^\intercal [\widetilde{\mathbf{H}}_k]_{21} + \widetilde{\mathbf{K}}_k^\intercal [\widetilde{\mathbf{H}}_k]_{22} \widetilde{\mathbf{K}}_k. \tag{9}$$

Hence, $\widetilde{\mathbf{H}}_k$ provides the required information to perform both policy update (8) and policy evaluation steps (9) in a policy iteration algorithm. We will see that because of the particular structure of our setup, $\widetilde{\mathbf{H}}_k$ enjoys a special block pattern captured by the proposed Patterned monoid, justifying the recovery of the block matrices X_1, X_2, Y_1, Y_2, Z_1, and Z_2 from $\widetilde{\mathbf{H}}_k$ in line 11 of D2SPI. D2SPI leverages this idea to implicitly learn $\widetilde{\mathbf{H}}_k$ from data (by adapting the idea of [19]) and exploit these matrix blocks in order to find an efficient suboptimal solution to the main distributed problem in (3).

Here, in addition to the policy update from data as in (8), we show that the same information can be used to also learn a gain margin directly from data (see Proposition 3). This gain margin is then used to guarantee the stability of the entire network (see Theorem 1).

Finally, for technical reasons, recall that the infinite-horizon state-feedback LQR problem with parameters (A, B, Q, R) can be cast as the minimization of

$$f_\Sigma(K) \coloneqq \mathrm{tr}\,[P_K \Sigma] \tag{10}$$

over the static stabilizing policy K, for some initial state distribution with covariance $\Sigma \succ 0$, where P_K is cost matrix associated with K satisfying the following Lyapunov equation [16], [17], [62]:

$$P_K = (A + BK)^\intercal P_K (A + BK) + Q + K^\intercal R K.$$

Herein, we set $\Sigma = \mathrm{I}$ and consider $f_{\mathrm{I}}(K)$.

B. Technical Observations and Main Proofs

In the remainder of this section, we first restate some well-known technical facts to make the paper self-contained, and then propose a few additional algebraic facts for our analysis whose proofs are deferred to Appendix II. We then continue with the proof of the main results.

Lemma 1: The following relations hold:

1) ([63]) When $X \succ 0$,

$$M^\intercal X N + N^\intercal X M \succeq -(a M^\intercal X M + \frac{1}{a} N^\intercal X N),$$

$$M^\intercal X N + N^\intercal X M \preceq a M^\intercal X M + \frac{1}{a} N^\intercal X N,$$

where $M, N \in \mathbb{R}^{n \times m}$ with $n \geq m$ and $a > 0$.

2) ([64, Lyapunov Equation]) Suppose that $A \in \mathbb{R}^{n \times n}$ has spectral radius less than 1, i.e., $\rho(A) < 1$. Then $A^\intercal X A + Q -$

$X = 0$ has a unique solution, $X = \sum_{j=0}^{\infty}(A^\intercal)^j Q A^j$. In this case, if $Q \succ 0$, then $X \succ 0$.

3) ([63, Block matrix inverse forluma (0.8.5.6)]) The following identity holds for matrices A, B, C, and D with compatible dimensions,

$$\begin{bmatrix} A & B \\ C & D \end{bmatrix}^{-1} = \begin{bmatrix} H^{-1} & -H^{-1}BD^{-1} \\ -D^{-1}CH^{-1} & D^{-1} + D^{-1}CH^{-1}BD^{-1} \end{bmatrix},$$

where D and $H = A - BD^{-1}C$ are invertible.

4) (Matrix Inversion Lemma [65]) The following identity holds,

$$(A + UCV)^{-1} = A^{-1} - A^{-1}U(C^{-1} + VA^{-1}U)^{-1}VA^{-1},$$

for matrices A, U, C, and V with compatible dimensions where A, C, and $A + UCV$ are invertible.

Finally, we provide the main technical lemma that streamlines the properties of the Patterned monoid under algebraic manipulation which will be frequently used in the proofs of Proposition 2 and Theorem 2.

Lemma 2: Suppose $\mathbf{N}_r \in \mathrm{PM}(r \times n, \mathbb{R})$ for some n and $r \geq 2$ such that $\mathbf{N}_r = \mathrm{I}_r \otimes \big(A - B\big) + \mathbb{1}_r \mathbb{1}_r^\intercal \otimes B$, for some $A \in \mathrm{GL}(n, \mathbb{R}) \cap \mathbb{S}^{n\times n}$ and $B \in \mathbb{S}^{n\times n}$. Then the following hold:

1) $\det(\mathbf{N}_r) = \det(A - B)^{r-1} \det(A + (r-1)B)$.

2) If $\mathbf{N}_r \succ 0$, then we have $A + (\ell - 1)B \succ 0$ for all $\ell = 0, 1, \cdots, r$. Furthermore, $A - \ell B \left(A + (\ell - 1)B\right)^{-1} B$ is invertible for $\ell = 1, 2, \cdots, r-1$.

3) If $\mathbf{N}_r \succ 0$, then $\mathbf{N}_r^{-1} \in \mathrm{PM}(r \times n, \mathbb{R})$; in fact,

$$\mathbf{N}_r^{-1} = \mathrm{I}_r \otimes \big(F_r + G_r\big) - \mathbb{1}_r \mathbb{1}_r^\intercal \otimes G_r,$$

with F_r and G_r defined as,

$$F_r = \big(A - (r-1)B\big(A + (r-2)B\big)^{-1}B\big)^{-1},$$
$$G_r = \big(A + (r-1)B\big)^{-1} B \big(A - B\big)^{-1}.$$

4) If also $\mathbf{M}_r \in \mathrm{L}(r \times n, \mathbb{R})$, i.e. $\mathbf{M}_r = \mathrm{I}_r \otimes \big(C - D\big) + \mathbb{1}_r \mathbb{1}_r^\intercal \otimes D$ for some $C, D \in \mathbb{R}^{n\times n}$ then,

$$\mathbf{N}_r \mathbf{M}_r = \mathrm{I}_r \otimes \big(A - B\big)\big(C - D\big) + \mathbb{1}_r \mathbb{1}_r^\intercal \otimes \Big(B\big(C - D\big) + \big(A - B\big)D + rBD\Big).$$

1) Proof of Proposition 1: For any Schur stable matrix A, and any symmetric positive definite matrix Q there exists a unique positive definite solution P to the discrete Lyapunov equation described by $P = \sum_{j=0}^{\infty}(A^\intercal)^j Q A^j$ (Lemma 1.2). Note, that $\mathrm{PM}(r \times n, \mathbb{R}) \subset \mathbb{S}^{rn\times rn} \cap \mathrm{L}(r \times n, \mathbb{R})$ by construction. Therefore, since $\mathrm{PM}(r \times n, \mathbb{R})$ is a monoid and $\mathrm{L}(r \times n, \mathbb{R})$ closed under matrix multiplication by Lemma 2, each summand falls in $\mathrm{L}(r \times n, \mathbb{R})$ whenever $Q \in \mathrm{PM}(r \times n, \mathbb{R})$. Also, as the infinite sum preserves the structure, $P \in \mathrm{L}(r \times n, \mathbb{R})$. To show that $P \in \mathrm{PM}(r \times n, \mathbb{R})$ it now suffices to note that $P \succ 0$ whenever $Q \succ 0$ because then every principle submatrix of P has to be positive definite and thus invertible. Conversely, if $P \in \mathrm{PM}(r \times n, \mathbb{R})$, then $Q = P - A^\intercal P A^\intercal$ also must lie in $\mathrm{PM}(r \times n, \mathbb{R})$ as $Q \succ 0$. This completes the proof. □

2) Proof of Proposition 2: At iteration k of the learning phase in Algorithm 1, the first claim is a direct consequence of the structure of $\mathcal{G}_{d,\mathrm{learn}}$ during the learning phase, where $\mathcal{G}_{d,\mathrm{learn}} = \mathcal{K}(\mathcal{G}_d)$ and hence $\boldsymbol{u}_i = (K_k - L_k)\boldsymbol{x}_i + L_k \sum_{j\in\mathcal{V}_{\mathcal{G}_{d,\mathrm{learn}}}} \boldsymbol{x}_j$ for all $i \in \mathcal{V}_{\mathcal{G}_{d,\mathrm{learn}}}$, which, in turn, results in $\tilde{\mathbf{u}}_t = \widetilde{\mathbf{K}}_k \tilde{\mathbf{x}}_t$ with $\widetilde{\mathbf{K}}_k$ as claimed. The stability of the policy $\widetilde{\mathbf{K}}_k$ for the pair $(\widetilde{\mathbf{A}}, \widetilde{\mathbf{B}})$ throughout the learning phase is then argued in Section I-B.5 under Assumption 1.

Next, the cost matrix $\widetilde{\mathbf{P}}_k$ associated with $\widetilde{\mathbf{K}}_k$ satisfies the Lyapunov equation (6). We can verify that $\widetilde{\mathbf{A}} + \widetilde{\mathbf{B}}\widetilde{\mathbf{K}}_k \in \mathrm{L}(d \times n, \mathbb{R})$ which is also Schur stable, and $\widetilde{\mathbf{Q}} + \widetilde{\mathbf{K}}_k^\intercal \widetilde{\mathbf{R}} \widetilde{\mathbf{K}}_k \in \mathrm{PM}(d \times n, \mathbb{R})$. Thus, by Proposition 1, we conclude that $\widetilde{\mathbf{P}}_k \in \mathrm{PM}(d \times n, \mathbb{R})$. So, let

$$\widetilde{\mathbf{P}}_k = \mathrm{I}_d \otimes \big(P_1 - P_2\big) + \mathbb{1}\mathbb{1}^\intercal \otimes P_2, \tag{11}$$

for some P_1, P_2. Note that $\widetilde{\mathbf{K}}_k$ is stabilizing and $\widetilde{\mathbf{Q}} + \widetilde{\mathbf{K}}_k^\intercal \widetilde{\mathbf{R}} \widetilde{\mathbf{K}}_k \succ 0$, therefore $\widetilde{\mathbf{P}}_k \succ 0$ from (6). But then, by Lemma 2.Item 2 and the structure of $\widetilde{\mathbf{P}}_k$ from (11), we claim that $\Delta P_k := P_1 - P_2 \succ 0$. Next, one can also verify that

$$\begin{aligned} &\big(\widetilde{\mathbf{A}} + \widetilde{\mathbf{B}}\widetilde{\mathbf{K}}_k\big)^\intercal \widetilde{\mathbf{P}}_k \big(\widetilde{\mathbf{A}} + \widetilde{\mathbf{B}}\widetilde{\mathbf{K}}_k\big) \\ &\qquad = \mathrm{I}_d \otimes \big(A_{\Delta K_k}^\intercal (\Delta P_k) A_{\Delta K_k}\big) + \mathbb{1}\mathbb{1}^\intercal \otimes (\star), \\ &\widetilde{\mathbf{Q}} + \widetilde{\mathbf{K}}_k^\intercal \widetilde{\mathbf{R}} \widetilde{\mathbf{K}}_k \\ &\qquad = \mathrm{I}_d \otimes \big(Q_d + (\Delta K_k)^\intercal R (\Delta K_k)\big) + \mathbb{1}\mathbb{1}^\intercal \otimes (\star), \end{aligned}$$

where $\Delta K_k = K_k - L_k$, $A_{\Delta K_k} = A + B(\Delta K_k)$, $Q_d = Q_1 + dQ_2$, and $(\star)$ is hiding extra terms. But then, by (6) and (11) we obtain that ΔP_k must satisfy:

$$\Delta P_k = A_{\Delta K_k}^\intercal (\Delta P_k) A_{\Delta K_k} + Q_d + \Delta K_k^\intercal R \Delta K_k, \tag{12}$$

which itself is a Lyapunov equation. Finally, since $Q_d + \Delta K_k^\intercal R \Delta K_k \succ 0$ and $\Delta P_k \succ 0$, by Lyapunov Stability Criterion we conclude that $A_{\Delta K_k}$ is Schur stable. This completes the proof. □

3) Proof of Proposition 3: Define the Lyapunov candidate function $V_k(\boldsymbol{x}_t) = \boldsymbol{x}_t^\intercal \Delta P_k \boldsymbol{x}_t$ with $\Delta P_k \succ 0$ as in (12) and where $\boldsymbol{x}_t$ contains the states of the closed-loop system $\boldsymbol{x}_{t+1} = \big(A + B(K_k - \alpha L_k)\big)\boldsymbol{x}_t$ for some scalar α. We show that for the given choice of α, V_k is decreasing. We define

$$\Delta V_k(\boldsymbol{x}_t) \coloneqq V_k(\boldsymbol{x}_{t+1}) - V_k(\boldsymbol{x}_t) = \boldsymbol{x}_t^\intercal \Gamma_k \boldsymbol{x}_t,$$

where

$$\Gamma_k = (A_{\Delta K_k} + (1-\alpha)L_k)^\intercal \Delta P_k (A_{\Delta K_k} + (1-\alpha)L_k) - \Delta P_k.$$

Suppose $\alpha < 1$, then from (12),

$$\begin{aligned} \Gamma_k &= -(Q_d + \Delta K_k^\intercal R \Delta K_k) + (1-\alpha)^2 L_k^\intercal B^\intercal \Delta P_k B L_k \\ &\quad + (1-\alpha)\left(A_{\Delta K_k}^\intercal \Delta P_k B L_k + L_k^\intercal B^\intercal \Delta P_k A_{\Delta K_k}\right) \\ &\preceq -(Q_d + \Delta K_k^\intercal R \Delta K_k) + (1-\alpha)^2 L_k^\intercal B^\intercal \Delta P_k B L_k \\ &\quad + (1-\alpha)\left(a A_{\Delta K_k}^\intercal \Delta P_k A_{\Delta K_k} + (1/a) L_k^\intercal B^\intercal \Delta P_k B L_k\right) \\ &= -\big(Q_d + \Delta K_k^\intercal R \Delta K_k\big) + (1-\alpha) a A_{\Delta K_k}^\intercal \Delta P_k A_{\Delta K_k} \\ &\quad + (1-\alpha)\left(1/a + 1 - \alpha\right) L_k^\intercal B^\intercal \Delta P_k B L_k \end{aligned}$$

where the inequality holds for any $a > 0$ due to Lemma 1. Let $\beta = (1-\alpha)/2$ and choose $a = \beta + \sqrt{\beta^2+1}$. Then, $(1/a + 1 - \alpha) = a$ and thus

$$\lambda_{\max}(\Gamma_k) \le -\lambda_{\min}\big[Q_d + \Delta K_k^\intercal R \Delta K_k\big] + (1-\alpha)a\lambda_{\max}\big[A_{\Delta K_k}^\intercal \Delta P_k A_{\Delta K_k} + L_k^\intercal B^\intercal \Delta P_k B L_k\big]$$

Now, using the parameters (13) constructing the blocks of (7), we obtain

$$\begin{aligned} L_k^\intercal B^\intercal \Delta P_k B L_k &= L_k^\intercal (\Delta Y - R) L_k, \quad \text{and} \\ A_{\Delta K_k}^\intercal \Delta P_k A_{\Delta K_k} &= \Delta X - Q_d + \Delta K_k^\intercal \Delta Z + \Delta Z^\intercal \Delta K_k \\ &\quad + \Delta K_k^\intercal (\Delta Y - R) \Delta K_k =: \Xi_k. \end{aligned}$$

Thus, the latter bound can be obtained completely from data as

$$\lambda_{\max}(\Gamma_k) \le \left((1-\alpha)a - \gamma_k\right) \lambda_{\max}\left[\Xi_k + L_k^\intercal (\Delta Y - R) L_k\right]$$

with

$$\gamma_k \coloneqq \lambda_{\min}\big[Q_d + \Delta K_k^\intercal R \Delta K_k\big] / \lambda_{\max}\left[\Xi_k + L_k^\intercal (\Delta Y - R) L_k\right],$$

which coincides with updates in Line 13 of Algorithm 1.

Finally, from the hypothesis $1 - \tau_k < \alpha < 1$ with $\tau_k = \sqrt{\gamma_k^2/(1+\gamma_k)}$, we obtain that $\gamma_k^2 - 4\beta^2\gamma_k - 4\beta^2 > 0$. But, since $\gamma_k > 0$, this second-order term in γ_k is positive only if

$$\gamma_k > 2\beta^2 + 2\beta\sqrt{\beta^2+1} = 2\beta(\beta + \sqrt{\beta^2+1}) = (1-\alpha)a.$$

Therefore, $\Delta V_k(\boldsymbol{x}_t) < 0$ for $1 - \tau_k < \alpha < 1$. Similar reasoning for $1 < \alpha < 1 + \tau_k$ also shows that $\Delta V_k(\boldsymbol{x}_t) < 0$ which completes the proof. □

4) Proof of Theorem 1: From Definition 1 (according to a consistent choice of labeling of the nodes so that the last d nodes are chosen as $\mathcal{G}_d$) and Proposition 2, the feedback policy in line 9 of Algorithm 1 can be cast in the compact form,

$$\begin{aligned} \text{Policy}_k(\mathcal{V}_\mathcal{G}) &= \begin{bmatrix} \text{Policy}_k(\mathcal{V}_{\mathcal{G}\setminus\mathcal{G}_{d,\text{learn}}}) \\ \text{Policy}_k(\mathcal{V}_{\mathcal{G}_{d,\text{learn}}}) \end{bmatrix} \\ &= \begin{bmatrix} \hat{\mathbf{K}}_{\mathcal{G}\setminus\mathcal{G}_d} & \hat{\mathbf{L}}_{\mathcal{G}\setminus\mathcal{G}_d} \\ \mathbf{0} & \widetilde{\mathbf{K}}_k \end{bmatrix} \begin{bmatrix} \text{State}_k(\mathcal{V}_{\mathcal{G}\setminus\mathcal{G}_{d,\text{learn}}}) \\ \text{State}_k(\mathcal{V}_{\mathcal{G}_{d,\text{learn}}}) \end{bmatrix} \\ &=: \hat{\mathbf{K}}_k \text{State}_k(\mathcal{G}), \end{aligned}$$

where $\hat{\mathbf{L}}_{\mathcal{G}\setminus\mathcal{G}_d} \coloneqq \frac{\tau_k}{d-1}[\mathcal{A}_\mathcal{G}]_{12} \otimes L_k$, $\widetilde{\mathbf{K}}_k \coloneqq \mathrm{I}_d \otimes (K_k - L_k) + \mathbb{1}_d\mathbb{1}_d^\intercal \otimes L_k$, $\hat{\mathbf{K}}_{\mathcal{G}\setminus\mathcal{G}_d} \coloneqq \mathrm{I}_{N-d} \otimes K_k - \left(\mathrm{I}_{N-d} - \frac{\tau_k}{d-1}\mathcal{A}_{\mathcal{G}\setminus\mathcal{G}_d}\right) \otimes L_k$, $\mathcal{A}_\mathcal{G}$ denotes the adjacency matrix of $\mathcal{G}$, and $[\mathcal{A}_\mathcal{G}]_{12}$ is its submatrix capturing the interconnection of $\mathcal{G} \setminus \mathcal{G}_d$ and $\mathcal{G}_d$:

$$\mathcal{A}_\mathcal{G} = \begin{bmatrix} \mathcal{A}_{\mathcal{G}\setminus\mathcal{G}_d} & [\mathcal{A}_\mathcal{G}]_{12} \\ * & \mathcal{A}_{\mathcal{G}_d} \end{bmatrix}.$$

Note that the structure of $\hat{\mathbf{K}}_k$ emanates from the fact that the information exchange is unidirectional during the learning phase. Now consider the closed-loop system of $\mathcal{G}$,

$$\begin{aligned} &\hat{\mathbf{A}}_\mathcal{G}|_{cl} \coloneqq \hat{\mathbf{A}} + \hat{\mathbf{B}}\hat{\mathbf{K}}_k \\ &= \begin{bmatrix} \mathrm{I}_{N-d} \otimes A + (\mathrm{I}_{N-d} \otimes B)\hat{\mathbf{K}}_{\mathcal{G}\setminus\mathcal{G}_d} & (\mathrm{I}_{N-d} \otimes B)\hat{\mathbf{L}}_{\mathcal{G}\setminus\mathcal{G}_d} \\ \mathbf{0} & \widetilde{\mathbf{A}}_{\widetilde{\mathbf{K}}_k} \end{bmatrix}, \end{aligned}$$

where $\widetilde{\mathbf{A}}_{\widetilde{\mathbf{K}}_k} \coloneqq \widetilde{\mathbf{A}} + \widetilde{\mathbf{B}}\widetilde{\mathbf{K}}_k$ is the closed-loop system of $\mathcal{G}_d$. Define $S = \mathrm{I}_{N-d} - \frac{\tau_k}{d-1}\mathcal{A}_{\mathcal{G}\setminus\mathcal{G}_d}$ and let J be the Jordan form of S according to the similarity transformation $\mathcal{T} \in \mathbb{R}^{(N-d)\times(N-d)}$ such that $\mathcal{T}S\mathcal{T}^{-1} = J$. Now consider the following similarity transformation of $\hat{\mathbf{A}}_\mathcal{G}|_{cl}$,

$$\begin{aligned} &\begin{bmatrix} \mathcal{T} \otimes \mathrm{I}_n & \mathbf{0} \\ \mathbf{0} & \mathrm{I}_d \otimes \mathrm{I}_n \end{bmatrix} \left(\hat{\mathbf{A}}_\mathcal{G}|_{cl}\right) \begin{bmatrix} \mathcal{T} \otimes \mathrm{I}_n & \mathbf{0} \\ \mathbf{0} & \mathrm{I}_d \otimes \mathrm{I}_n \end{bmatrix}^{-1} \\ &= \begin{bmatrix} \mathrm{I}_{N-d} \otimes \big(A + BK_k\big) - J \otimes BL_k & * \\ \mathbf{0} & \widetilde{\mathbf{A}}_{\widetilde{\mathbf{K}}_k} \end{bmatrix}. \end{aligned}$$

Note that $\mathrm{I}_{N-d} \otimes \big(A + BK_k\big) - J \otimes BL_k$ is a block upper triangular matrix whose diagonal blocks are equal to $A + B(K_k - \lambda_i(S)L_k)$ for $i = 1, \cdots, N-d$. We already know from Proposition 2 that $\widetilde{\mathbf{A}}_{\widetilde{\mathbf{K}}_k}$ is Schur stable. Hence, in order to show that $\hat{\mathbf{A}}_\mathcal{G}|_{cl}$ is Schur stable, it suffices to show that $\rho\left(A + B(K - \lambda_i(S)L)\right) < 1$ for $i = 1, \cdots, N-d$. Recall that $|\lambda_i(\mathcal{A}_\mathcal{G})| \le d_{\max}$ [66], thus by definition of S and the fact that $d_{\max} = d-1$, we conclude that $|\lambda_i(S) - 1| \le \tau_k$. The claim now follows directly from Proposition 3. □

5) Proof of Theorem 2: At iteration k of the learning phase in Algorithm 1, consider $\widetilde{\mathbf{H}}_k$ and its corresponding blocks as defined in (7). First, we consider the structure of the stabilizing feedback policy $\widetilde{\mathbf{K}}_k$ as shown in Proposition 2, together with that of system parameters $(\widetilde{\mathbf{A}}, \widetilde{\mathbf{B}}, \widetilde{\mathbf{Q}}, \widetilde{\mathbf{R}})$, and apply Lemma 2 and Proposition 1 to conclude that $[\widetilde{\mathbf{H}}_k]_{11}, [\widetilde{\mathbf{H}}_k]_{22} \in \mathrm{PM}(d \times n, \mathbb{R})$ and $[\widetilde{\mathbf{H}}_k]_{21} \in \mathrm{L}(d \times n, \mathbb{R})$. Thus, we get

$$\begin{aligned} [\widetilde{\mathbf{H}}_k]_{11} &= \mathrm{I}_d \otimes \big(X_1 - X_2\big) + \mathbb{1}\mathbb{1}^\intercal \otimes X_2, \\ [\widetilde{\mathbf{H}}_k]_{22} &= \mathrm{I}_d \otimes \big(Y_1 - Y_2\big) + \mathbb{1}\mathbb{1}^\intercal \otimes Y_2, \\ [\widetilde{\mathbf{H}}_k]_{21} &= \mathrm{I}_d \otimes \big(Z_1 - Z_2\big) + \mathbb{1}\mathbb{1}^\intercal \otimes Z_2, \end{aligned}$$

which coincides with the recovery of X_i, Y_i and Z_i for $i = 1, 2$ in Line 11 of Algorithm 1. We can also unravel the structure of block matrices constructing $\widetilde{\mathbf{H}}_k$ and obtain that

$$\begin{aligned} X_1 &= Q_1 + (d-1)Q_2 + A^\intercal P_1 A, & Y_2 &= B^\intercal P_2 B, \\ X_2 &= -Q_2 + A^\intercal P_2 A, & Z_1 &= B^\intercal P_1 A, \\ Y_1 &= R + B^\intercal P_1 B, & Z_2 &= B^\intercal P_2 A, \end{aligned} \tag{13}$$

with P_1 and P_2 as in (11). Similarly, by Lemma 2, we obtain

$$[\widetilde{\mathbf{H}}_k]_{22}^{-1} = \mathrm{I}_d \otimes (F - G) + \mathbb{1}\mathbb{1}^\intercal \otimes G,$$

where (by Lemma 2 Item 3) F and G must satisfy

$$\begin{aligned} F^{-1} &= Y_1 - (d-1)Y_2\big(Y_1 + (d-2)Y_2\big)^{-1}Y_2, \\ G &= \big(Y_1 + (d-1)Y_2\big)^{-1} Y_2 \big(Y_1 - Y_2\big)^{-1}, \end{aligned}$$

which coincides with the definitions in Line 12 of Algorithm 1. Finally, by definition of K_{k+1} and L_{k+1} in Line 12 and Lemma 2 Item 4, one can verify that $\widetilde{\mathbf{K}}_{k+1} = -[\widetilde{\mathbf{H}}_k]_{22}^{-1}[\widetilde{\mathbf{H}}_k]_{21}$, which coincides with the policy iteration in the Hewer's algorithm [21] for the system in $\mathcal{G}_{d,\text{learn}}$ (see also Section I-A). Note that by assumption the pair (A, B) is controllable, so is the system $(\widetilde{\mathbf{A}}, \widetilde{\mathbf{B}})$ in $\mathcal{G}_{d,\text{learn}}$. Therefore, these

updates are guaranteed to remain stabilizing and converge to the claimed optimal LQR policy $\widetilde{\mathbf{K}}^*$ provided that we have access to the true parameters $\widetilde{\mathbf{H}}_k$.

Next, consider LQR cost $f_{\mathrm{I}}(\widetilde{\mathbf{K}})$ as in (10) but for the problem parameters $(\widetilde{\mathbf{A}}, \widetilde{\mathbf{B}}, \widetilde{\mathbf{Q}}, \widetilde{\mathbf{R}})$. For completing the proof of convergence, it is left to argue that there exists a large enough integer C such that, at each iteration k of the learning phase, the recursive least square in Algorithm 2 provides a more accurate estimation, denoted by θ_k°, of the true parameters $\theta_k = \mathrm{vech}(\widetilde{\mathbf{H}}_k)$, and the LQR cost $f_{\mathrm{I}}(\widetilde{\mathbf{K}}) = \mathrm{tr}(\widetilde{\mathbf{P}}_{\widetilde{\mathbf{K}}})$ decreases. This claim essentially follows by [19, Theorem 1] which we try to summarize for completeness. For that, at iteration k, let us denote the policy obtained using the estimated parameters by $\widetilde{\mathbf{K}}_k^\circ$ which in turn estimates the true policy $\widetilde{\mathbf{K}}_k$. We then define a "Lyapunov" function candidate

$$s_k := f_{\mathrm{I}}(\widetilde{\mathbf{K}}_{k-1}^\circ) + \big\|\theta_{k-2} - \theta_{k-2}^\circ\big\|.$$

Following the same induction reasoning as that of [19, Theorem 1] and under persistently exciting input, there exists an integer C such that

$$s_{k+1} = s_k - \varepsilon_1(C)\big\|\theta_{k-1} - \theta_{k-1}^\circ\big\| - \varepsilon_2(C)\big\|\widetilde{\mathbf{K}}_k - \widetilde{\mathbf{K}}_{k-1}^\circ\big\|^2,$$

for some positive constants $\varepsilon_1(C)$ and $\varepsilon_1(C)$ that are independent of k. But then, $s_{k+1} \le s_k$ and

$$\varepsilon_1(C)\sum_{k=2}^{\infty}\big\|\theta_{k-1} - \theta_{k-1}^\circ\big\| \le s_1;\ \ \varepsilon_2(C)\sum_{k=2}^{\infty}\big\|\widetilde{\mathbf{K}}_k - \widetilde{\mathbf{K}}_{k-1}^\circ\big\|^2 \le s_1.$$

Also, s_1 is bounded as $\widetilde{\mathbf{K}}_1$ stabilizes $(\widetilde{\mathbf{A}}, \widetilde{\mathbf{B}})$. This guarantees that, first, $\widetilde{\mathbf{K}}_k^\circ$ remains stabilizing as $f_{\mathrm{I}}(\widetilde{\mathbf{K}}_{k-1}^\circ) \le s_0$ and $\widetilde{\mathbf{Q}} \succ 0$; second, the estimates θ_{k-1}° become more accurate; and third, $\widetilde{\mathbf{K}}_k^\circ \to \widetilde{\mathbf{K}}^*$ as $\widetilde{\mathbf{K}}_{k+1} \to \widetilde{\mathbf{K}}^*$ at $k \to \infty$ [21]. This completes the proof. □

6) Proof of Theorem 3: Consider the LQR cost $f_{\mathrm{I}}(\hat{\mathbf{K}})$ as in (10) but for the problem parameters $(\hat{\mathbf{A}}, \hat{\mathbf{B}}, \hat{\mathbf{Q}}, \hat{\mathbf{R}})$. In the "unstructured" case (i.e. ignoring the constraint $K \in \mathcal{U}_{m,n}^N(\mathcal{G})$), we know that the optimal LQR cost matrix for the entire networked system satisfies [67],

$$\hat{\mathbf{P}}_{\mathrm{lqr}}^* = \hat{\mathbf{A}}_{\hat{\mathbf{K}}_{\mathrm{lqr}}}^\intercal \hat{\mathbf{P}}_{\mathrm{lqr}} \hat{\mathbf{A}}_{\hat{\mathbf{K}}_{\mathrm{lqr}}} + \hat{\mathbf{K}}_{\mathrm{lqr}}^\intercal \hat{\mathbf{R}} \hat{\mathbf{K}}_{\mathrm{lqr}} + \hat{\mathbf{Q}}, \tag{14}$$

where $\hat{\mathbf{K}}_{\mathrm{lqr}} = \arg\min_K f_{\mathrm{I}}(K)$. Moreover, the cost matrix $\hat{\mathbf{P}}^*$– associated with the structured policy $\hat{\mathbf{K}}^*$ learned by Algorithm 1–satisfies

$$\hat{\mathbf{P}}^* = \hat{\mathbf{A}}_{\hat{\mathbf{K}}^*}^\intercal \hat{\mathbf{P}}^* \hat{\mathbf{A}}_{\hat{\mathbf{K}}^*} + \hat{\mathbf{K}}^{*\intercal} \hat{\mathbf{R}} \hat{\mathbf{K}}^* + \hat{\mathbf{Q}}, \tag{15}$$

where $\hat{\mathbf{A}}_{\hat{\mathbf{K}}^*} = \hat{\mathbf{A}} + \hat{\mathbf{B}}\hat{\mathbf{K}}^*$ and $\hat{\mathbf{K}}^* \in \mathcal{U}_{m,n}^N(\mathcal{G})$. Finally, let $\hat{\mathbf{K}}_{\mathrm{struc}}^* \in \arg\min_{K \in \mathcal{U}_{m,n}^N(\mathcal{G})} f_{\mathrm{I}}(K)$ denote a "structured" stabilizing optimal LQR policy which is associated with the cost matrix $\hat{\mathbf{P}}_{\mathrm{struc}}^*$. We know such a policy exists since the smooth cost is lower-bounded and $\hat{\mathbf{K}}_1 = I_N \otimes K_1 \in \mathcal{U}_{m,n}^N(\mathcal{G})$ is a feasible point of this optimization—as K_1 is assumed to be stabilizing for the single pair (A, B). Therefore,

$$\mathrm{tr}\left[\hat{\mathbf{P}}_{\mathrm{lqr}}^*\right] = f_{\mathrm{I}}\left(\hat{\mathbf{K}}_{\mathrm{lqr}}^*\right) \le f_{\mathrm{I}}\left(\hat{\mathbf{K}}_{\mathrm{struc}}^*\right) \le f_{\mathrm{I}}\left(\hat{\mathbf{K}}^*\right) = \mathrm{tr}\left[\hat{\mathbf{P}}^*\right],$$

where the last inequality above follows by the fact that $\hat{\mathbf{K}}^*$ is a feasible solution to the structured problem by construction, *i.e.*, $\hat{\mathbf{K}}^* \in \mathcal{U}_{m,n}^N(\mathcal{G})$. Therefore, $0 \le \mathrm{gap}(\hat{\mathbf{K}}^*) \le \mathrm{tr}\left[\hat{\mathbf{P}}^* - \hat{\mathbf{P}}_{\mathrm{lqr}}^*\right]$. But then, one can obtain from (14), (15), and some algebraic manipulation,

$$\hat{\mathbf{P}}^* - \hat{\mathbf{P}}_{\mathrm{lqr}}^* = \mathbf{A}_{\hat{\mathbf{K}}_{\mathrm{lqr}}^*}^\intercal \Big(\hat{\mathbf{P}}^* - \hat{\mathbf{P}}_{\mathrm{lqr}}\Big)\mathbf{A}_{\hat{\mathbf{K}}_{\mathrm{lqr}}} + \mathbf{M}',$$

where

$$\mathbf{M}' = \mathbf{A}_{\hat{\mathbf{K}}^*}^\intercal \hat{\mathbf{P}}^* \mathbf{A}_{\hat{\mathbf{K}}^*} - \mathbf{A}_{\hat{\mathbf{K}}_{\mathrm{lqr}}^*}^\intercal \hat{\mathbf{P}}^* \mathbf{A}_{\hat{\mathbf{K}}_{\mathrm{lqr}}^*} + \hat{\mathbf{K}}^{*\intercal}\hat{\mathbf{R}}\hat{\mathbf{K}}^* - \hat{\mathbf{K}}_{\mathrm{lqr}}^\intercal \hat{\mathbf{R}} \hat{\mathbf{K}}_{\mathrm{lqr}}.$$

Since $\hat{\mathbf{P}}^* - \hat{\mathbf{P}}_{\mathrm{lqr}}^* \succ 0$ and $\hat{\mathbf{A}}_{\hat{\mathbf{K}}_{\mathrm{lqr}}}$ is contractible by the hypothesis, from the first part and Theorem 1 in [68] we obtain,

$$\mathrm{gap}(\hat{\mathbf{K}}^*) \le \frac{\mathrm{tr}(\mathbf{M}')}{1 - \sigma_{\max}^2\big(\hat{\mathbf{A}}_{\hat{\mathbf{K}}_{\mathrm{lqr}}}\big)} = \frac{\mathrm{tr}(\mathbf{M})}{1 - \sigma_{\max}^2\big(\hat{\mathbf{A}}_{\hat{\mathbf{K}}_{\mathrm{lqr}}}\big)},$$

where the last equality follows by the cyclic permutation property of trace and definition of $\mathbf{A}_{\hat{\mathbf{K}}^*}$. □

Appendix II
Proof of Lemma 2

First, we show that the following algebraic identities hold which will be used in the proof of Lemma 2.

Lemma 3: Suppose A and B are symmetric matrices such that A, $A-B$, and $A+(n-1)B$ are all invertible for some integer n. Then the following relations hold:

1) $\big(A+nB\big)\big(A+(n-1)B\big)^{-1}\big(A-B\big) = A - nB\big(A+(n-1)B\big)^{-1}B$.

2) $\big(A+nB\big)\big(A+(n-1)B\big)^{-1}\big(A-B\big) = \big(A-B\big)\big(A+(n-1)B\big)^{-1}\big(A+nB\big)$.

3) $\big(A+nB\big)\big(A-B\big)^{-1}B = B\big(A-B\big)^{-1}\big(A+nB\big)$.

Proof: These claims follow by the algebraic manipulations below: First,

$$\begin{aligned}
&\big(A+nB\big)\big(A+(n-1)B\big)^{-1}\big(A-B\big)\\
=&\Big(\big(A+(n-1)B\big)+B\Big)\big(A+(n-1)B\big)^{-1}\big(A-B\big)\\
=&\Big(\mathrm{I}+B\big(A+(n-1)B\big)^{-1}\Big)\big(A-B\big)\\
=&A-B+B\big(A+(n-1)B\big)^{-1}A-B\big(A+(n-1)B\big)^{-1}B\\
=&A+B\Big(\big(A+(n-1)B\big)^{-1}A-\mathrm{I}\Big)-B\big(A+(n-1)B\big)^{-1}B\\
=&A+B\big(A+(n-1)B\big)^{-1}\Big(A-\big(A+(n-1)B\big)\Big)\\
&-B\big(A+(n-1)B\big)^{-1}B\\
=&A-(n-1)B\big(A+(n-1)B\big)^{-1}B-B\big(A+(n-1)B\big)^{-1}B\\
=&A-nB\big(A+(n-1)B\big)^{-1}B.
\end{aligned}$$

Second,

$$\begin{aligned}
&\big(A+nB\big)\big(A+(n-1)B\big)^{-1}\big(A-B\big)\\
=&-n\big(A-B\big)\big(A+(n-1)B\big)^{-1}\big(A-B\big)\\
&+(n+1)A\big(A+(n-1)B\big)^{-1}\big(A-B\big)\\
=&-n\big(A-B\big)\big(A+(n-1)B\big)^{-1}\big(A-B\big)\\
&+(n+1)\big(\mathrm{I}+(n-1)BA^{-1}\big)^{-1}\big(A-B\big)\\
=&-n\big(A-B\big)\big(A+(n-1)B\big)^{-1}\big(A-B\big)\\
&+(n+1)\big(A-B\big)\Big(\big(\mathrm{I}+(n-1)BA^{-1}\big)\big(A-B\big)\Big)^{-1}\big(A-B\big)
\end{aligned}$$

$$
\begin{aligned}
&= -n(A-B)\big(A+(n-1)B\big)^{-1}(A-B)\\
&\quad +(n+1)(A-B)\Big(\big(A-B\big)\big(\mathrm{I}+(n-1)A^{-1}B\big)\Big)^{-1}(A-B)\\
&= -n(A-B)\big(A+(n-1)B\big)^{-1}(A-B)\\
&\quad +(n+1)(A-B)\big(A+(n-1)B\big)^{-1}A\\
&=(A-B)\big(A+(n-1)B\big)^{-1}(A+nB).
\end{aligned}
$$

Finally,

$$
\begin{aligned}
&(A+nB)(A-B)^{-1}B\\
&\quad=\big(A-B+(n+1)B\big)(A-B)^{-1}B\\
&\quad=\big(\mathrm{I}+(n+1)B(A-B)^{-1}\big)B=B\big(\mathrm{I}+(n+1)(A-B)^{-1}B\big)\\
&\quad=B(A-B)^{-1}(A+nB).
\end{aligned}
$$

□

Proof of Lemma 2: Part 1) We prove the claim by induction. First, note that both $\mathbf{N}_r$ and its principle submatrix A are invertible. For $r=2$, by Schur complement of $\mathbf{N}_2$, we get

$$
\begin{aligned}
\det(\mathbf{N}_2) &= \det(A)\det(A-BA^{-1}B)\\
&= \det(A)\det(\mathrm{I}-BA^{-1})\det(\mathrm{I}+BA^{-1})\det(A)\\
&= \det(A-B)\det(A+B).
\end{aligned}
$$

Now, suppose the claim holds for $r=p$. Then, for $r=p+1$, similarly by Schur complement we get

$$
\begin{aligned}
&\det(\mathbf{N}_{p+1}) = \det(A)\det\big(\mathbf{N}_p-\mathbb{1}\mathbb{1}^\intercal\otimes BA^{-1}B\big)\\
&= \det(A)\det(A-B)^{p-1}\\
&\qquad\cdot\det\big(A-BA^{-1}B+(p-1)(B-BA^{-1}B)\big),\\
&= \det(A)\det(A-B)^{p-1}\\
&\qquad\cdot\det\big(A-B+pBA^{-1}(A-B)\big),\\
&= \det(A)\det(A-B)^{p}\det\big(I+pBA^{-1}\big),\\
&= \det(A-B)^{p}\det(A+pB),
\end{aligned}
$$

where the second equality follows by applying the induction hypothesis to $\mathbf{N}_p-\mathbb{1}\mathbb{1}^\intercal\otimes BA^{-1}B$ and some algebraic manipulation. This completes the proof.

Part 2) From item 1 of the this lemma,

$$
\begin{aligned}
&\det(\mathbf{N}_r-\lambda\mathrm{I}_r\otimes\mathrm{I})\\
&\qquad=\det(A-\lambda\mathrm{I}-B)^{r-1}\det(A-\lambda\mathrm{I}+(r-1)B),
\end{aligned}
$$

implying that the spectrum of $\mathbf{N}_r$ coincides with that of $A-B$ and $A+(r-1)B$—modulo algebraic multiplicities. Hence, $\mathbf{N}_r\succ 0$ results in $A-B\succ 0$ and $A+(r-1)B\succ 0$. Furthermore, $\mathbf{N}_r\succ 0$ if and only if its principal submatrices are positive definite. So, by applying the latter result to principal submatrices, we claim that $A+\ell B\succ 0$ for $\ell=0,\cdots,r-2$. Lastly, from item 1 of Lemma 3, for $\ell=1,\cdots,r-1$ we have

$$
\begin{aligned}
&A-\ell B\big(A+(\ell-1)B\big)^{-1}B\\
&\quad=\big(A+\ell B\big)\big(A+(\ell-1)B\big)^{-1}(A-B),
\end{aligned}
$$

which, by the first part of this claim, is invertible as a multiplication of invertible matrices.

3) Since $\mathbf{N}_r$ and A are invertible, the Schur complement $\mathbf{N}_{r-1}-L_{r-1}A^{-1}L_{r-1}^\intercal$ is also invertible where $L_{r-1}=\mathbb{1}_{r-1}\otimes B$. We prove the claim by induction on r. For $r=2$, by [63, Block matrix inverse forluma (0.8.5.6)],

$$
\mathbf{N}_2^{-1}=\begin{bmatrix} H^{-1} & -H^{-1}BA^{-1}\\ -A^{-1}BH^{-1} & A^{-1}+A^{-1}BH^{-1}BA^{-1}\end{bmatrix},
$$

where $H=A-BA^{-1}B$ is the Schur complement of A. By [63, Woodbury inversion formula (0.7.4.1)], $H^{-1}=A^{-1}+A^{-1}BH^{-1}BA^{-1}$, establishing the recurrence of diagonal blocks. Also, $\mathbf{N}_2$ is symmetric, so is $\mathbf{N}_2^{-1}$ and thus establishing that $\mathbf{N}_2\in\mathrm{PM}(2\times n,\mathbb{R})$. Now, from item 1 in Lemma 3 with $n=1$, we get that

$$
\begin{aligned}
A^{-1}BH^{-1} &= A^{-1}B\big(A-B\big)^{-1}A\big(A+B\big)^{-1}\\
&= A^{-1}B\big(A+B\big)^{-1}A\big(A-B\big)^{-1}\\
&= A^{-1}B\big(\mathrm{I}+A^{-1}B\big)^{-1}\big(A-B\big)^{-1}\\
&= \big(\mathrm{I}-(\mathrm{I}+A^{-1}B)^{-1}\big)\big(A-B\big)^{-1}=G_2,
\end{aligned}
$$

where we also used $\big(A-B\big)^{-1}A\big(A+B\big)^{-1}=\big(A+B\big)^{-1}A\big(A-B\big)^{-1}$ derived from Lemma 3 item 2. Hence,

$$
\mathbf{N}_2^{-1}=\mathrm{I}_2\otimes\big(H^{-1}+H^{-1}BA^{-1}\big)-\mathbb{1}_2\mathbb{1}_2^\intercal\otimes\big(H^{-1}BA^{-1}\big).
$$

Assume that the claim holds for $r=p$. To extend the result to $r=p+1$, again by [63, Block matrix inverse forluma (0.8.5.6)] and [63, Woodbury inversion formula (0.7.4.1)],

$$
\mathbf{N}_{p+1}^{-1}=\begin{bmatrix} A & L_p^\intercal\\ L_p & \mathbf{N}_p\end{bmatrix}^{-1}=\begin{bmatrix} P^{-1} & -P^{-1}L_p^\intercal\mathbf{N}_p^{-1}\\ -\mathbf{N}_p^{-1}L_pP^{-1} & \big(\mathbf{N}_p-L_pA^{-1}L_p\big)^{-1}\end{bmatrix},
$$

where $P=A-L_p^\intercal\mathbf{N}_p^{-1}L_p$ and $L_p=\mathbb{1}_p\otimes B$. Let $\mathbf{N}_p^{-1}=\mathrm{I}_p\otimes\big(F_p+G_p\big)-\mathbb{1}_p\mathbb{1}_p^\intercal\otimes G_p$ where,

$$
\begin{aligned}
F_p &= \Big(A-(p-1)B\big(A+(p-2)B\big)^{-1}B\Big)^{-1},\\
G_p &= \big(A+(p-1)B\big)^{-1}B\big(A-B\big)^{-1},
\end{aligned}
$$

where the inversions are valid from item 2 of the current Lemma. By simplification we get $P=A-pB\big(F_p-(p-1)G_p\big)B$ and from Lemma 3 items 1 and 2 for $n=p-1$,

$$
\begin{aligned}
F_p-(p-1)G_p &= \Big(A-(p-1)B\big(A+(p-2)B\big)^{-1}B\Big)^{-1}\\
&\quad-(p-1)\big(A+(p-1)B\big)^{-1}B\big(A-B\big)^{-1}\\
&=\big(A+(p-1)B\big)^{-1}\big(A+(p-2)B\big)\big(A-B\big)^{-1}\\
&\quad-(p-1)\big(A+(p-1)B\big)^{-1}B\big(A-B\big)^{-1}\\
&=\big(A+(p-1)B\big)^{-1},
\end{aligned}
$$

where the first term in the right hand side of the first equation undergoes Item 1 and Item 2 in Lemma 3, consecutively. And, the last equality follows by a direct simplification. The latter established equality results in $P=A-pB\big(A+(p-1)B\big)^{-1}B$. Next, considering the off-diagonal blocks of $\mathbf{N}_{p+1}^{-1}$, with some simplification, each block of $P^{-1}L_p^\intercal\mathbf{N}_p^{-1}$ is equivalent to $P^{-1}B\big(F_p-(p-1)G_p\big)$ and using the previous reasoning and Lemma 3 it can be simplified to,

$$
\begin{aligned}
&P^{-1}B\big(F_p-(p-1)G_p\big)\\
&=P^{-1}B\big(A+(p-1)B\big)^{-1}\\
&=\big(A+pB\big)^{-1}\big(A+(p-1)B\big)\big(A-B\big)^{-1}B\big(A+(p-1)B\big)^{-1}\\
&=\big(A+pB\big)^{-1}B\big(A-B\big)^{-1}.
\end{aligned}
$$

Similarly, each block of $\mathbf{N}_p^{-1}L_pP^{-1}$ is also equal to $\big(A+(p-1)B\big)^{-1}B\big(A-B\big)^{-1}$. Therefore, it only remains to show that the blocks of $\big(\mathbf{N}_p-L_pA^{-1}L_p\big)^{-1}$ are consistent with the desired pattern in $\mathbf{N}_{p+1}^{-1}$. Note that $\mathbf{N}_p-L_pA^{-1}L_p=\mathrm{I}\otimes\big(A-B\big)+\mathbb{1}\mathbb{1}^\intercal\otimes\big(B-BA^{-1}B\big)$. Hence, with some algebraic rigor we can show that each diagonal term of $(\mathbf{N}_p-L_pA^{-1}L_p)^{-1}$ is $\big(A+pB\big)^{-1}\big(A+(p-1)B\big)\big(A-B\big)^{-1}$ and each off-diagonal becomes $-\big(A+pB\big)B\big(A-B\big)^{-1}$. Hence,

$$\mathbf{N}_{p+1}^{-1}=\mathrm{I}\otimes\big(F_{p+1}+G_{p+1}\big)-\mathbb{1}\mathbb{1}^\intercal G_{p+1},$$

with F_{p+1} and G_{p+1} defined as,

$$\begin{aligned}F_{p+1}&=\big(A+pB\big)^{-1}\big(A+(p-1)B\big)\big(A-B\big)^{-1}\\G_{p+1}&=\big(A+pB\big)B\big(A-B\big)^{-1}.\end{aligned}$$

4)With direct multiplication and using the mixed-product property of Kronecker products,

$$\begin{aligned}\mathbf{N}_r\mathbf{M}_r=&\mathrm{I}_r\otimes\big(A-B\big)\big(C-D\big)+\mathbb{1}_r\mathbb{1}_r^\intercal\otimes B\big(C-D\big)\\&+\mathbb{1}_r\mathbb{1}_r^\intercal\otimes\big(A-B\big)D+r\mathbb{1}_r\mathbb{1}_r^\intercal\otimes BD\\=&\mathrm{I}_r\otimes\big(A-B\big)\big(C-D\big)\\&+\mathbb{1}_r\mathbb{1}_r^\intercal\otimes\Big(B\big(C-D\big)+\big(A-B\big)D+rBD\Big).\square\end{aligned}$$

REFERENCES

[1] S. Talebi, S. Alemzadeh, and M. Mesbahi, "Distributed model-free policy iteration for networks of homogeneous systems," in *Conference on Decision and Control*, pp. 6970–6975, 2021.

[2] D. A. Bini, B. Iannazzo, and B. Meini, *Numerical Solution of Algebraic Riccati Equations*. Society for Industrial and Applied Mathematics, 2011.

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

[4] C. H. Papadimitriou and J. Tsitsiklis, "Intractable problems in control theory," *SIAM Journal on Control and Optimization*, vol. 24, no. 4, pp. 639–654, 1986.

[5] D. M. Stipanović, G. Inalhan, R. Teo, and C. J. Tomlin, "Decentralized overlapping control of a formation of unmanned aerial vehicles," *Automatica*, vol. 40, no. 8, pp. 1285–1296, 2004.

[6] F. Borrelli, T. Keviczky, G. J. Balas, G. Stewart, K. Fregene, and D. Godbole, "Hybrid decentralized control of large scale systems," in *International Workshop on Hybrid Systems: Computation and Control*, pp. 168–183, Springer, 2005.

[7] F. Borrelli and T. Keviczky, "Distributed LQR design for identical dynamically decoupled systems," *IEEE Transactions on Automatic Control*, vol. 53, no. 8, pp. 1901–1912, 2008.

[8] P. Deshpande, P. P. Menon, C. Edwards, and I. Postlethwaite, "A distributed control law with guaranteed LQR cost for identical dynamically coupled linear systems," in *American Control Conference (ACC)*, pp. 5342–5347, IEEE, 2011.

[9] W. Wang, F. Zhang, and C. Han, "Distributed linear quadratic regulator control for discrete-time multi-agent systems," *IET Control Theory & Applications*, vol. 11, no. 14, pp. 2279–2287, 2017.

[10] P. Massioni and M. Verhaegen, "Distributed control for identical dynamically coupled systems: A decomposition approach," *IEEE Transactions on Automatic Control*, vol. 54, no. 1, pp. 124–135, 2009.

[11] J. Engwerda, *LQ Dynamic Optimization and Differential Games*. John Wiley & Sons, 2005.

[12] V. Blondel and J. N. Tsitsiklis, "NP-hardness of some linear control design problems," *SIAM Journal on Control and Optimization*, vol. 35, no. 6, pp. 2118–2127, 1997.

[13] V. Gupta, B. Hassibi, and R. M. Murray, "A sub-optimal algorithm to synthesize control laws for a network of dynamic agents," *International Journal of Control*, vol. 78, no. 16, pp. 1302–1313, 2005.

[14] M. Rotkowitz and S. Lall, "A characterization of convex problems in decentralized control," *IEEE Transactions on Automatic Control*, vol. 50, no. 12, pp. 1984–1996, 2005.

[15] B. Bamieh, F. Paganini, and M. A. Dahleh, "Distributed control of spatially invariant systems," *IEEE Transactions on Automatic Control*, vol. 47, no. 7, pp. 1091–1107, 2002.

[16] K. Mårtensson and A. Rantzer, "Gradient methods for iterative distributed control synthesis," in *Conference on Decision and Control held jointly with 2009 28th Chinese Control Conference*, pp. 549–554, IEEE, 2009.

[17] J. Bu, A. Mesbahi, M. Fazel, and M. Mesbahi, "LQR through the lens of first order methods: Discrete-time case," *arXiv preprint arXiv:1907.08921*, 2019.

[18] J. Bu, A. Mesbahi, and M. Mesbahi, "On topological properties of the set of stabilizing feedback gain," *IEEE Transactions on Automatic Control*, vol. 66, no. 2, pp. 730–744, 2021.

[19] S. J. Bradtke, B. E. Ydstie, and A. G. Barto, "Adaptive linear quadratic control using policy iteration," in *1994 American Control Conference (ACC)*, vol. 3, pp. 3475–3475, 1994.

[20] S. J. Bradtke, *Incremental Dynamic Programming for Online Adaptive Optimal Control*. PhD thesis, University of Massachusetts Amherst, 1994.

[21] G. Hewer, "An iterative technique for the computation of the steady state gains for the discrete optimal regulator," *IEEE Transactions on Automatic Control*, vol. 16, no. 4, pp. 382–384, 1971.

[22] D. McFadden, "On the controllability of decentralized macroeconomic systems: The assignment problem," in *Mathematical Systems Theory and Economics I/II*, pp. 221–239, Springer, 1969.

[23] S. H. Wang and E. Davison, "On the stabilization of decentralized control systems," *IEEE Transactions on Automatic Control*, vol. 18, no. 5, pp. 473–478, 1973.

[24] N. Sandell, P. Varaiya, M. Athans, and M. Safonov, "Survey of decentralized control methods for large scale systems," *IEEE Transactions on Automatic Control*, vol. 23, no. 2, pp. 108–128, 1978.

[25] P. Ioannou, "Decentralized adaptive control of interconnected systems," *IEEE Transactions on Automatic Control*, vol. 31, no. 4, pp. 291–298, 1986.

[26] J. A. Fax and R. M. Murray, "Information flow and cooperative control of vehicle formations," *IEEE Transactions on Automatic Control*, vol. 49, no. 9, pp. 1465–1476, 2004.

[27] P. Deshpande, P. Menon, C. Edwards, and I. Postlethwaite, "Sub-optimal distributed control law with $\mathcal{H}_2$ performance for identical dynamically coupled linear systems," *IET Control Theory & Applications*, vol. 6, no. 16, pp. 2509–2517, 2012.

[28] N. Motee and A. Jadbabaie, "Optimal control of spatially distributed systems," *IEEE Transactions on Automatic Control*, vol. 53, no. 7, pp. 1616–1629, 2008.

[29] A. Chapman, A. D. González, M. Mesbahi, L. Dueñas-Osorio, and R. M. D'Souza, "Data-guided control: Clustering, graph products, and decentralized control," in *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*, pp. 493–498, 2017.

[30] S. Alemzadeh and M. Mesbahi, "Influence models on layered uncertain networks: A guaranteed-cost design perspective," in *2018 IEEE Conference on Decision and Control (CDC)*, pp. 5251–5256, 2018.

[31] Y. R. Stürz, A. Eichler, and R. S. Smith, "Distributed control design for heterogeneous interconnected systems," *IEEE Transactions on Automatic Control*, vol. 66, no. 11, pp. 5112–5127, 2020.

[32] S. Vasisht, *Data-guided Estimation and Tracking Methods for Unmanned Aerial Vehicles*. PhD thesis, University of Washington, 2019.

[33] P. Dorato, V. Cerone, and C. Abdallah, *Linear-Quadratic Control: An Introduction*. Simon & Schuster, Inc., 1994.

[34] P. P. Khargonekar, I. R. Petersen, and K. Zhou, "Robust stabilization of uncertain linear systems: Quadratic stabilizability and H^∞ control theory," *IEEE Transactions on Automatic control*, vol. 35, no. 3, pp. 356–361, 1990.

[35] Z. Li, Z. Duan, L. Xie, and X. Liu, "Distributed robust control of linear multi-agent systems with parameter uncertainties," *International Journal of Control*, vol. 85, no. 8, pp. 1039–1050, 2012.

[36] L. Ljung, *System Identification: Theory for the User*. Pearson Education, 1998.

[37] 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 Transactions on Automatic Control*, 2020.

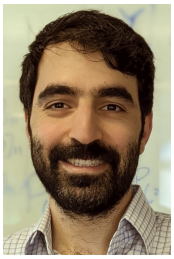
[38] A. Alaeddini, S. Alemzadeh, A. Mesbahit, and M. Mesbahi, "Linear model regression on time-series data: Non-asymptotic error bounds and applications," in *2018 IEEE Conference on Decision and Control (CDC)*, pp. 2259–2264, 2018.

[39] S. Dean, H. Mania, N. Matni, B. Recht, and S. Tu, "On the sample complexity of the linear quadratic regulator," *Foundations of Computational Mathematics*, pp. 1–47, 2019.

[40] S. Talebi, S. Alemzadeh, N. Rahimi, and M. Mesbahi, "Online regulation of unstable linear systems from a single trajectory," in *2020 59th IEEE Conference on Decision and Control (CDC)*, pp. 4784–4789, 2020.
[41] S. Oymak and N. Ozay, "Non-asymptotic identification of LTI systems from a single trajectory," in *American Control Conference*, pp. 5655–5661, IEEE, 2019.
[42] F. L. Lewis, D. Vrabie, and K. G. Vamvoudakis, "Reinforcement learning and feedback control: Using natural decision methods to design optimal adaptive controllers," *IEEE Control Systems Magazine*, vol. 32, no. 6, pp. 76–105, 2012.
[43] Y. Luo, Z. Yang, Z. Wang, and M. Kolar, "Natural actor-critic converges globally for hierarchical linear quadratic regulator," *arXiv preprint arXiv:1912.06875*, 2019.
[44] S. Alemzadeh and M. Mesbahi, "Distributed Q-learning for dynamically decoupled systems," in *2019 American Control Conference (ACC)*, pp. 772–777, 2019.
[45] T.-J. Chang and S. Shahrampour, "Distributed online linear quadratic control for linear time-invariant systems," in *American Control Conference*, pp. 923–928, IEEE, 2021.
[46] J. Li, H. Modares, T. Chai, F. L. Lewis, and L. Xie, "Off-policy reinforcement learning for synchronization in multiagent graphical games," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 28, no. 10, pp. 2434–2445, 2017.
[47] S. Talebi, S. Alemzadeh, L. J. Ratliff, and M. Mesbahi, "Distributed learning in network games: a dual averaging approach," in *Conference on Decision and Control*, pp. 5544–5549, IEEE, 2019.
[48] A. Nowé, P. Vrancx, and Y.-M. De Hauwere, "Game theory and multi-agent reinforcement learning," in *Reinforcement Learning*, pp. 441–470, Springer, 2012.
[49] G. C. Goodwin and K. S. Sin, *Adaptive Filtering Prediction and Control*. Englewood Cliffs, N.J.: Prentice-Hall, 1984.
[50] J. C. Willems, P. Rapisarda, I. Markovsky, and B. L. De Moor, "A note on persistency of excitation," *Systems & Control Letters*, vol. 54, no. 4, pp. 325–329, 2005.
[51] Y. Yu, S. Talebi, H. J. van Waarde, U. Topcu, M. Mesbahi, and B. Açıkmeşe, "On controllability and persistency of excitation in data-driven control: Extensions of Willems' fundamental lemma," in *Conference on Decision and Control*, pp. 6485–6490, 2021.
[52] S. Talebi, S. Alemzadeh, N. Rahimi, and M. Mesbahi, "On regularizability and its application to online control of unstable LTI systems," *IEEE Transactions on Automatic Control*, vol. 67, no. 12, pp. 6413–6428, 2022.
[53] Y. Jiang and Z.-P. Jiang, "Computational adaptive optimal control for continuous-time linear systems with completely unknown dynamics," *Automatica*, vol. 48, no. 10, pp. 2699–2704, 2012.
[54] F. L. Lewis and K. G. Vamvoudakis, "Reinforcement learning for partially observable dynamic processes: Adaptive dynamic programming using measured output data," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 41, no. 1, pp. 14–25, 2010.
[55] P. Comon and G. H. Golub, "Tracking a few extreme singular values and vectors in signal processing," *Proceedings of the IEEE*, vol. 78, no. 8, pp. 1327–1343, 1990.
[56] S. Haykin, *Adaptive Filter Theory*. Prentice Hall, 2002.
[57] X. Chen and E. Hazan, "Black-box control for linear dynamical systems," in *Proceedings of Thirty Fourth Conference on Learning Theory* (M. Belkin and S. Kpotufe, eds.), vol. 134 of *Proceedings of Machine Learning Research*, pp. 1114–1143, PMLR, 15–19 Aug 2021.
[58] Y. Abbasi-Yadkori and C. Szepesvári, "Regret bounds for the adaptive control of linear quadratic systems," in *Proceedings of the 24th Annual Conference on Learning Theory* (S. M. Kakade and U. von Luxburg, eds.), vol. 19 of *Proceedings of Machine Learning Research*, (Budapest, Hungary), pp. 1–26, PMLR, 09–11 Jun 2011.
[59] N. Agarwal, B. Bullins, E. Hazan, S. Kakade, and K. Singh, "Online control with adversarial disturbances," in *International Conference on Machine Learning*, pp. 111–119, 2019.
[60] Y. S. Hung and A. MacFarlane, *Multivariable Feedback: A Quasi-Classical Approach*. Springer-Verlag New York, Inc., 1982.
[61] R. Bellman, "Dynamic programming," *Science*, vol. 153, no. 3731, pp. 34–37, 1966.
[62] W. Levine, T. Johnson, and M. Athans, "Optimal limited state variable feedback controllers for linear systems," *IEEE Transactions on Automatic Control*, vol. 16, no. 6, pp. 785–793, 1971.
[63] R. A. Horn and C. R. Johnson, *Matrix Analysis*. Cambridge University Press, 2012.
[64] P. Lancaster, "Explicit solutions of linear matrix equations," *SIAM review*, vol. 12, no. 4, pp. 544–566, 1970.
[65] M. A. Woodbury, "Inverting modified matrices," *Statistical Research Group Memorandum Report*, vol. 42, no. 106, p. 336, 1950.
[66] B. Bollobás, *Modern Graph Theory*, vol. 184. Springer Science & Business Media, 2013.
[67] G. C. Goodwin, S. F. Graebe, and M. E. Salgado, *Control System Design*. Upper Saddle River, NJ: Prentice Hall, 2001.
[68] T. Mori, N. Fukuma, and M. Kuwahara, "On the discrete Lyapunov matrix equation," *IEEE Transactions on Automatic Control*, vol. 27, no. 2, pp. 463–464, 1982.

Siavash Alemzadeh (S'17) received his B.S. in Mechanical Engineering from Sharif University of Technology, Tehran, Iran in 2014. He received his M.S. in Mechanical Engineering in 2015 and his Ph.D. from William E. Boeing department of Aeronautics and Astronautics in 2020 from the University of Washington, Seattle, USA. He is currently a data and applied scientist at Microsoft.

His research interests are reinforcement learning, data-driven control of distributed systems, and networked dynamical systems and their applications in transportation, infrastructure networks, and robotics.



Shahriar Talebi (Member, IEEE) is currently a postdoctoral fellow at Harvard University with the School of Engineering and Applied Sciences (SEAS) and the Department of Statistics. He received the Ph.D. degree in aeronautics and astronautics, specializing in control theory, and the M.Sc. degree in Mathematics, focusing on differential geometry, both from the University of Washington, Seattle, WA, USA, in 2023. He also received the B.Sc. degree from the Sharif University of Technology, Tehran, Iran, in 2014, and the M.Sc. degree from the University of Central Florida, Orlando, FL, in 2017, both in electrical engineering. Dr. Talebi was the recipient of the 2022 Excellence in Teaching Award at UW. He is also a recipient of a number of scholarships, including the William E. Boeing Endowed Fellowship, Paul A. Carlstedt Endowment, and Latvian Arctic Pilot-A. Vagners Memorial Scholarship (UW, 2018–2019), as well as the Frank Hubbard Engineering Scholarship (UCF, 2017).

His research interests include control theory, differential geometry, learning for control, networked dynamical systems, and game theory.

Mehran Mesbahi (F'15) received his Ph.D. from the University of Southern California, Los Angeles, in 1996. He was a member of the Guidance, Navigation, and Analysis group at Jet Propulsion Laboratory from 1996-2000 and an Assistant Professor of Aerospace Engineering and Mechanics at the University of Minnesota from 2000-2002. He is currently a Professor of Aeronautics and Astronautics, Adjunct Professor of Electrical and Computer Engineering and Mathematics at the University of Washington, Executive Director of the Joint Center for Aerospace Technology Innovation, and member of the Washington State Academy of Sciences. He was the recipient of NSF CAREER Award, NASA Space Act Award, UW Distinguished Teaching Award, and UW College of Engineering Innovator Award for Teaching.

His research interests are distributed and networked aerospace systems, systems and control theory, and learning.