# Regret Analysis of Distributed Online LQR Control for Unknown LTI Systems

Ting-Jui Chang and Shahin Shahrampour, *Senior Member*, *IEEE*

***Abstract*—Online optimization has recently opened avenues to study optimal control for time-varying cost functions that are unknown in advance. Inspired by this line of research, we study the distributed online linear quadratic regulator (LQR) problem for linear time-invariant (LTI) systems with unknown dynamics. Consider a multi-agent network where each agent is modeled as a LTI system. The network has a global *time-varying* quadratic cost, which may evolve adversarially and is only *partially* observed by each agent sequentially. The goal of the network is to collectively (i) estimate the unknown dynamics and (ii) compute local control sequences competitive to the best centralized policy in hindsight, which minimizes the sum of network costs over time. This problem is formulated as a *regret* minimization. We propose a distributed variant of the online LQR algorithm, where agents compute their system estimates during an exploration stage. Each agent then applies distributed online gradient descent on a semi-definite programming (SDP) whose feasible set is based on the agent system estimate. We prove that with high probability the regret bound of our proposed algorithm scales as $O(T^{2/3}\log T)$, implying the consensus of all agents over time. We also provide simulation results verifying our theoretical guarantee.**

## I. Introduction

In recent years, there has been a significant interest on problems arising at the interface of control and machine learning. Among classical control problems, LQR control [1]–[3] is a prominent point in case. LQR control centers around LTI systems, where the control-state pairs introduce a quadratic cost with *time-invariant* parameters. When the dynamics of the LTI system is known, for finite-horizon and infinite-horizon problems, the optimal controllers have closed-form solutions, which can be derived by solving the corresponding Riccati equations.

Despite the excellent insights on the LQR problem provided by the classical control theory, in practical problems we might encounter two challenges. (I) The environment could change in an unpredictable way, which makes the cost parameters *time-varying* and *unknown* in advance (e.g., in variable-supply electricity production and building climate control with time-varying energy costs [4]). (II) Furthermore, the dynamics of the LTI system may be *unknown*. The former challenge has motivated research at the interface of online optimization and control, where online LQR problem is cast as a *regret* minimization and the performance of an online algorithm is compared to that of the best fixed control policy in hindsight. The regret metric is particularly meaningful in the online setting, where the cost parameters are unknown in advance. The focus of online LQR is on the finite-time performance from a learning-theory perspective (see details of this literature in item 4 of Subsection I-A). The latter challenge is addressed via adaptive control in general. In this case, the learner must strike a balance between exploration (estimating the system dynamics) and exploitation (using the estimates to compete with the performance of the optimal controller) [5]–[8].

In this work, we consider the distributed online LQR problem for a network of LTI systems with *unknown* dynamics. Each system is represented by an agent in the network that has a global *time-varying* quadratic cost. The cost sequence may evolve adversarially and is only *partially* observed by each agent sequentially, i.e., the agents do not have the knowledge of local costs in advance. The term *adversarial* implies that there is no statistical/probabilistic assumption imposed on the cost sequence. The goal of each agent is to generate a control sequence (in an online fashion) that is competitive to that of the best centralized policy in hindsight, and the sub-optimality is formulated by the notion of *regret*. Specifically, for an online control problem with a finite time horizon $T$, a successful algorithm must attain a regret that is sub-linear in $T$, which implies that its time-averaged performance tends to that of the best policy in hindsight asymptotically. In practice, this setting can be applied for modeling the energy consumption in mobile sensor networks as described in Example 1. Our main contributions in addressing this problem are as follows:

1) We propose a decentralized algorithm with two phases. In the *exploration* phase, each agent first spends $T_0$ iterations to collect data for the system identification. Then, in the following $T_1$ iterations, all agents jointly compute system estimates by applying the **EXTRA** algorithm [9], which is an iterative decentralized optimization method. In the *exploitation* phase (of length $T - T_0 - T_1$), agents perform distributed online gradient descent on a SDP (whose feasible set is constructed by local system estimates) and extract the control policies accordingly.
2) The exploration and exploitation phases play conflicting roles in regret minimization. In particular, if $(T_0 + T_1)$ is larger, the SDP is built on finer system estimates and the incurred regret during the exploitation phase is lower. However, that also means that the regret during the exploration is larger. Therefore, we must strike a balance between exploration and exploitation. In the main theorem, we quantify the dependence of the regret bound on $T_0$ and $T_1$, and we show that with an optimal choice of $T_0$ and $T_1$, the regret is bounded by $O(T^{2/3}\log T)$, where $T$ is the total number of iterations. This implies that the agents reach consensus and collectively compete with the best fixed controller in hindsight.
3) Besides the exploration-exploitation trade-off, the main

T.J. Chang and S. Shahrampour are with the Department of Mechanical and Industrial Engineering, Northeastern University, Boston, MA 02115, USA. email:{chang.tin,s.shahrampour}@northeastern.edu.

This work is supported in part by NSF ECCS-2136206 Award.

technical challenge is that the decentralized system identification step results in different SDPs across agents. This implies that the feasible set of SDP varies from one agent to another, and we cannot directly use distributed online optimization results on a common feasible set. We draw upon techniques from alternating projections to tackle this problem.

4) We provide simulations verifying the sublinearity of regret and the consensus of all agents. We further illustrate the impact of network connectivity on the regret.

Our technical proofs are provided in the Appendix (Section VI of [10]).

### A. Related Literature

**(1) Distributed LQR Control:** Distributed LQR has been widely studied in the control literature. A number of works focus on multi-agent systems with known, identical decoupled dynamics. In [11], a distributed control design is proposed by solving a single LQR problem whose size scales with the maximum degree of the graph capturing the network. The authors of [12] derive the necessary condition for an optimal distributed controller design, resulting in a non-convex optimization problem. The work of [13] addresses a multi-agent network, where the dynamics of each agent is a single integrator. The authors of [13] show that the computation of the optimal controller requires the knowledge of the graph and the initial information of all agents. Given the difficulty of precisely solving the optimal distributed controller, Jiao et al. [14] provide the sufficient conditions to obtain sub-optimal controllers. All of the aforementioned works need global information such as network topology to compute the controllers. On the other hand, Jiao et al. [15] propose a decentralized method to compute the controllers and show that the system will reach consensus. For the case of unknown dynamics, Alemzadeh et al. [16] propose a distributed Q-learning algorithm for dynamically decoupled systems. There are other works focusing on distributed control without assuming identical decoupled sub-systems. Fattahi et al. [17] study distributed controllers for unknown and sparse LTI systems. Furieri et al. [18] address model-free methods for distributed LQR problems and provide sample-complexity bounds for problems with the local gradient dominance property (e.g., quadratically-invariant problems). The work of [19] investigates the convergence of distributed controllers to a global minimum for quadratically invariant problems with first-order methods.

**(2) System Identification of LTI Systems:** For solving LQR problems with unknown dynamics, we first need to learn the underlying system. To provide performance guarantees for the controller, it is important to explicitly quantify the uncertainty of the model estimate. The classical theory of system identification for LTI systems (e.g., [20]–[23]) characterizes the asymptotic properties of the estimators. On the contrary, recent results in statistical learning focus on finite-time guarantees. In [24], it is shown that for fully observable systems, a least-squares estimator can learn the underlying dynamics from multiple trajectories. These results are later extended to the estimation using a single trajectory [25], [26]. For partially observable systems, estimators with polynomial sample complexities are provided in the literature (e.g., [27]–[30]), and the work of [31] improves the sample complexity to poly-logarithmic.

**(3) Online LQR with Unknown Dynamics and Time-Invariant Costs:** There is a recent line of research dealing with LQR control problems with unknown dynamics. Several techniques are proposed using (i) gradient estimation (e.g., [32]–[35]), (ii) the estimation of dynamics matrices and derivation of the controller by considering the estimation uncertainty (e.g., [7], [8], [24], [36]–[38]), and (iii) wave-filtering [39], [40].

**(4) Online Control with Time-Varying Costs:** Recently, there has been a significant interest in studying linear dynamical systems with time-varying cost functions, where online learning techniques are applied. This literature investigates two scenarios: **I) Known Systems:** Cohen et al. [4] study the SDP relaxation for online LQR control and establish a regret bound of $O(\sqrt{T})$ for known LTI systems with time-varying quadratic costs. Agarwal et al. [41] propose the disturbance-action policy parameterization and reduce the online control problem to online convex optimization with memory. They show that for adversarial disturbances and arbitrary time-varying convex functions, the regret is $O(\sqrt{T})$. Agarwal et al. [42] consider the case of time-varying strongly-convex functions and improve the regret bound to $O(\text{poly}(\log T))$. Simchowitz et al. [43] further extend the $O(\text{poly}(\log T))$ regret bound to partially observable systems with semi-adversarial disturbances. Yu et al. [44] incorporate the idea of model predictive control into online LQR control with a time-invariant cost function and correct noise predictions. Zhang et al. [45] extend this idea to the setup where costs are time-varying and accurate disturbance predictions are not accessible. Both works provide dynamic regret bounds with a term shrinking exponentially with the prediction window. Our previous work [46] studies the distributed online LQR control with known dynamics and provides the regret bound of $O(\sqrt{T})$. **II) Unknown Systems:** For fully observable systems, Hazan et al. [47] derive the regret of $O(T^{2/3})$ for time-varying convex functions with adversarial noises. For partially observable systems, the work of [43] addresses the cases of (i) convex functions with adversarial noises and (ii) strongly-convex functions with semi-adversarial noises, and it provides regret bounds of $O(T^{2/3})$ and $O(\sqrt{T})$, respectively. Lale et al. [48] establish an $O(\text{poly}(\log T))$ regret bound for the case of stochastic perturbations, time-varying strongly-convex functions, and partially observable states.

Our work lies precisely at the interface of distributed LQR, online LQR and adaptive control, addressing distributed online LQR with unknown dynamics.

## II. Preliminaries and Problem Formulation

### A. Notation

| | |
|---|---|
| $[n]$ | The set $\{1, 2, \dots, n\}$ for any integer $n$ |
| $\mathrm{Tr}(\cdot)$ | The trace operator |
| $\|\cdot\|$ | Euclidean (spectral) norm of a vector (matrix) |
| $\|\cdot\|_F$ | Frobenius norm of a matrix |
| $\mathrm{E}[\cdot]$ | The expectation operator |
| $\Pi_{\mathcal{S}}[\cdot]$ | The operator for the projection to set $\mathcal{S}$ |
| $[\mathbf{A}]_{ij}$ | The entry in the $i$-th row and $j$-th column of $\mathbf{A}$ |
| $[\mathbf{A}]_{:,j}$ | The $j$-th column of $\mathbf{A}$ |
| $\mathbf{A} \bullet \mathbf{B}$ | $\mathrm{Tr}(\mathbf{A}^\top \mathbf{B})$ |
| $\mathbf{A} \succeq \mathbf{B}$ | $(\mathbf{A} - \mathbf{B})$ is positive semi-definite |
| $\mathbb{1}$ | The vector of all ones |
| $\mathbf{e}_i$ | The $i$-th basis vector |
| $\mathrm{vec}(\mathbf{A})$ | Vectorized version of the matrix $\mathbf{A}$ |

### B. Distributed Online LQR Control with Unknown Dynamics

We consider a multi-agent network of $m$ LTI systems, where the dynamics of agent $i$ is given as,

$$\mathbf{x}_{i,t+1} = \mathbf{A}\mathbf{x}_{i,t} + \mathbf{B}\mathbf{u}_{i,t} + \mathbf{w}_{i,t}, \quad i \in [m]$$

and $\mathbf{x}_{i,t} \in \mathrm{R}^d$ and $\mathbf{u}_{i,t} \in \mathrm{R}^k$ represent agent $i$ state and control (or action) at time $t$, respectively. Furthermore, $\mathbf{A} \in \mathrm{R}^{d\times d}$, $\mathbf{B} \in \mathrm{R}^{d\times k}$, and $\mathbf{w}_{i,t}$ is a Gaussian noise with zero mean and covariance $\mathbf{W} \succeq \sigma^2\mathbf{I}$. The system parameters $(\mathbf{A}, \mathbf{B})$ are *unknown* to all agents and need to be estimated. The noise sequence $\{\mathbf{w}_{i,t}\}$ is independent over time and agents. We also assume that $\|[\mathbf{A}\ \mathbf{B}]\|_F \le \vartheta$ and let $n := d + k$ for the presentation simplicity.

Departing from the classical LQR control, we consider the *online* distributed LQR problem, where the cost functions are *unknown* in advance. At round $t$, agent $i$ receives the state $\mathbf{x}_{i,t}$ and applies the action $\mathbf{u}_{i,t}$. Then, positive semi-definite cost matrices $\mathbf{Q}_{i,t}$ and $\mathbf{R}_{i,t}$ are revealed, and the agent incurs the cost $\mathbf{x}_{i,t}^\top \mathbf{Q}_{i,t}\mathbf{x}_{i,t} + \mathbf{u}_{i,t}^\top \mathbf{R}_{i,t}\mathbf{u}_{i,t}$. Throughout this paper, we assume that $\mathrm{Tr}(\mathbf{Q}_{i,t}), \mathrm{Tr}(\mathbf{R}_{i,t}) \le C$ for all $i, t$ and some $C > 0$. Agent $i$ follows a policy that selects the control $\mathbf{u}_{i,t}$ based on the observed cost matrices $\mathbf{Q}_{i,1}, \dots, \mathbf{Q}_{i,t-1}$ and $\mathbf{R}_{i,1}, \dots, \mathbf{R}_{i,t-1}$, as well as the information received from its *local* neighborhood. This policy is not driven based on individual costs. On the contrary, agents follow a *team* goal through minimizing a cost collectively as we describe next.

**Centralized Benchmark:** In order to gauge the performance of a distributed online LQR algorithm, we require a centralized benchmark. In this paper, we focus on the *finite-horizon* problem, where for a centralized policy $\pi$, the cost after $T$ steps is given as

$$J_T(\pi) = \mathrm{E}\left[\sum_{t=1}^{T} \mathbf{x}_t^{\pi\top} \mathbf{Q}_t \mathbf{x}_t^\pi + \mathbf{u}_t^{\pi\top} \mathbf{R}_t \mathbf{u}_t^\pi\right], \tag{1}$$

where $\mathbf{Q}_t = \sum_{i=1}^m \mathbf{Q}_{i,t}$ and $\mathbf{R}_t = \sum_{i=1}^m \mathbf{R}_{i,t}$, and the expectation is over the possible randomness of the policy as well as the noise. The superscript $\pi$ in $\mathbf{u}_t^\pi$ and $\mathbf{x}_t^\pi$ alludes that the state-control pairs are chosen by the policy $\pi$, given full access to cost matrices of all agents. Notice that in the *infinite-horizon* version of the problem with time-invariant cost matrices $(\mathbf{Q}, \mathbf{R})$, where the goal is to minimize $\lim_{T\to\infty} J_T(\pi)/T$, it is well-known that for a controllable LTI system $(\mathbf{A}, \mathbf{B})$, the optimal policy is given by the constant linear feedback, i.e., $\mathbf{u}_t^\pi = \mathbf{K}\mathbf{x}_t^\pi$ for a matrix $\mathbf{K} \in \mathrm{R}^{k\times d}$.

**Regret Definition:** The goal of a distributed online LQR algorithm $\mathcal{A}$ is to mimic the performance of an ideal centralized algorithm that solves (1). The main two challenges are (i) the online nature of the problem, where cost matrices become available sequentially, and (ii) the distributed setup, where agent $i$ only receives information about the sequence $\{\mathbf{Q}_{i,t}, \mathbf{R}_{i,t}\}$ while the network cost is based on $\{\mathbf{Q}_t, \mathbf{R}_t\}$. In this setting, each agent $j$ locally generates the control sequence $\{\mathbf{u}_{j,t}\}_{t=1}^T$, that is competitive to the best policy among a benchmark policy class $\Pi$. This can be formulated as minimizing the individual regret, which is defined as follows

$$\mathrm{Regret}_T^j(\mathcal{A}) := J_T^j(\mathcal{A}) - \min_{\pi\in\Pi} J_T(\pi), \tag{2}$$

for agent $j \in [m]$, where

$$J_T^j(\mathcal{A}) = \mathrm{E}\left[\sum_{t=1}^{T} \mathbf{x}_{j,t}^{\mathcal{A}\top} \mathbf{Q}_t \mathbf{x}_{j,t}^{\mathcal{A}} + \mathbf{u}_{j,t}^{\mathcal{A}\top} \mathbf{R}_t \mathbf{u}_{j,t}^{\mathcal{A}}\right]. \tag{3}$$

A successful distributed algorithm is one that keeps the regret sublinear with respect to $T$. Of course, this also depends on the choice of the benchmark policy class $\Pi$, which is assumed to be the set of strongly stable policies (to be defined precisely in Section II-C). Since the underlying dynamics is unknown, agents have to find a good trade-off between *exploration* (estimating the system parameters) and *exploitation* (keeping the regret sublinear).

**Network Structure:** The underlying network topology is captured by an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = [m]$ denotes the set of nodes (i.e., agents) and $\mathcal{E}$ represents the set of edges. If there is an edge between nodes $i$ and $j$, agent $i$ assigns a positive weight $[\mathbf{P}]_{ji}$ to the information received from agent $j$. If there is no edge between nodes $i$ and $j$, $[\mathbf{P}]_{ji}$ is equal to zero. The weighted adjacency matrix $\mathbf{P}$ is assumed to be symmetric and doubly stochastic, i.e., all elements of $\mathbf{P}$ are non-negative and $\sum_{i=1}^m [\mathbf{P}]_{ji} = \sum_{j=1}^m [\mathbf{P}]_{ji} = 1$. The network is further assumed to be connected, i.e., for any two agents $i, j \in [m]$, there is a (potentially multi-hop) path from $i$ to $j$. We also assume $\mathbf{P}$ has a positive diagonal. Then, there exists a geometric mixing bound for $\mathbf{P}$ [49], such that $\sum_{j=1}^m \left|[\mathbf{P}^k]_{ji} - 1/m\right| \le \sqrt{m}\beta^k$, $i \in [m]$, where $\beta$ is the second largest singular value of $\mathbf{P}$. Agents do not directly share their observed cost functions with each other, but they exchange local parameters used for constructing the controllers, which effectively captures the information of local costs in their neighborhood. The communication is consistent with the structure of $\mathbf{P}$. We elaborate on this in the algorithm description.

**Example 1.** *Our framework can be used for minimizing the energy consumption in mobile sensor networks (MSNs) [50] in time-varying settings. Consider a MSN where at time $t$ the total mobility cost (or budget) of sensors is modeled by matrices $(\mathbf{Q}_t, \mathbf{R}_t)$. Each agent $i$ has a local budget of $(\mathbf{Q}_{i,t}, \mathbf{R}_{i,t})$, but the team goal is to design actions that minimize the global network cost over time. Then, actions of*

*this MSN should be guided to minimize the global cost in* (1)*, though each sensor only has local information.*

### C. Strong Stability and Sequential Strong Stability

We consider the set of strongly stable linear (i.e., $\mathbf{u} = \mathbf{K}\mathbf{x}$) controllers as the benchmark policy class. Following [4], we define the notion of strong stability as follows.

**Definition 1.** *(Strong Stability) A linear policy $\mathbf{K}$ is $(\kappa, \gamma)$-strongly stable (for $\kappa > 0$ and $0 < \gamma \leq 1$) for the LTI system $(\mathbf{A}, \mathbf{B})$, if $\|\mathbf{K}\| \leq \kappa$, and there exist matrices $\mathbf{L}$ and $\mathbf{H}$ such that $\mathbf{A} + \mathbf{B}\mathbf{K} = \mathbf{H}\mathbf{L}\mathbf{H}^{-1}$, with $\|\mathbf{L}\| \leq 1 - \gamma$ and $\|\mathbf{H}\|\|\mathbf{H}^{-1}\| \leq \kappa$.*

Strong stability is a quantitative version of stability, in the sense that any stable policy is strongly stable for some $\kappa$ and $\gamma$, and vice versa [4]. A strongly stable policy ensures fast mixing and exponential convergence to a steady-state distribution. In particular, for the LTI system $\mathbf{x}_{t+1} = \mathbf{A}\mathbf{x}_t + \mathbf{B}\mathbf{u}_t + \mathbf{w}_t$, if a $(\kappa, \gamma)$-strongly stable policy $\mathbf{K}$ is applied ($\mathbf{u}_t = \mathbf{K}\mathbf{x}_t$), $\widehat{\mathbf{X}}_t$ (the state covariance matrix of $\mathbf{x}_t$) converges to $\mathbf{X}$ (the steady-state covariance matrix) with the following exponential rate

$$\|\widehat{\mathbf{X}}_t - \mathbf{X}\| \leq \kappa^2 e^{-2\gamma t} \|\widehat{\mathbf{X}}_0 - \mathbf{X}\|.$$

See Lemma 3.2 in [4] for details. The *sequential* nature of *online* LQR control requires another notion of strong stability, called *sequential strong stability* [4], defined as follows.

**Definition 2.** *(Sequential Strong Stability) A sequence of linear policies $\{\mathbf{K}_t\}_{t=1}^T$ is $(\kappa, \gamma)$-strongly stable if there exist matrices $\{\mathbf{H}_t\}_{t=1}^T$ and $\{\mathbf{L}_t\}_{t=1}^T$ such that $\mathbf{A} + \mathbf{B}\mathbf{K}_t = \mathbf{H}_t\mathbf{L}_t\mathbf{H}_t^{-1}$ for all $t$ with the following properties,*

1) $\|\mathbf{L}_t\| \leq 1 - \gamma$ *and* $\|\mathbf{K}_t\| \leq \kappa$.
2) $\|\mathbf{H}_t\| \leq \beta'$ *and* $\|\mathbf{H}_t^{-1}\| \leq 1/\alpha'$ *with* $\kappa = \beta'/\alpha'$.
3) $\|\mathbf{H}_{t+1}^{-1}\mathbf{H}_t\| \leq 1 + \gamma/2$.

Sequential strong stability generalizes strong stability to the time-varying scenario, where a sequence of policies $\{\mathbf{K}_t\}_{t=1}^T$ is used. The convergence of steady-state covariance matrices induced by $\{\mathbf{K}_t\}_{t=1}^T$ is characterized as follows.

**Lemma 1.** *(Lemma 3.5 in [4]) Suppose a time-varying policy ($\mathbf{u}_t = \mathbf{K}_t\mathbf{x}_t$) is applied. Denote the steady-state covariance matrix of $\mathbf{K}_t$ as $\mathbf{X}_t$. If $\{\mathbf{K}_t\}$ are $(\kappa, \gamma)$-sequentially strongly stable and $\|\mathbf{X}_t - \mathbf{X}_{t-1}\| \leq \eta$, $\widehat{\mathbf{X}}_t$ (the state covariance matrix of $\mathbf{x}_t$) converges to $\mathbf{X}_t$ as follows*

$$\|\widehat{\mathbf{X}}_{t+1} - \mathbf{X}_{t+1}\| \leq \kappa^2 e^{-\gamma t} \|\widehat{\mathbf{X}}_1 - \mathbf{X}_1\| + \frac{2\eta\kappa^2}{\gamma}.$$

### D. SDP Relaxation for LQR Control

For the following dynamical system

$$\mathbf{x}_{t+1} = \mathbf{A}\mathbf{x}_t + \mathbf{B}\mathbf{u}_t + \mathbf{w}_t, \quad \mathbf{w}_t \sim \mathcal{N}(0, \mathbf{W}),$$

the infinite-horizon version of (1), i.e., minimize $\lim_{T\to\infty} J_T(\pi)/T$, with fixed cost matrices $\mathbf{Q}$ and $\mathbf{R}$ can be relaxed via a SDP when the steady-state distribution exists. For $\nu > 0$, the SDP relaxation is formulated as [4]

$$\begin{aligned} \text{minimize} \quad & J(\Sigma) = \begin{pmatrix} \mathbf{Q} & 0 \\ 0 & \mathbf{R} \end{pmatrix} \bullet \Sigma \\ \text{subject to} \quad & \Sigma_{\mathbf{xx}} = [\mathbf{A}\ \mathbf{B}]\Sigma[\mathbf{A}\ \mathbf{B}]^\top + \mathbf{W}, \\ & \Sigma \succeq 0, \quad \mathrm{Tr}(\Sigma) \leq \nu, \end{aligned} \tag{4}$$

where

$$\Sigma = \begin{pmatrix} \Sigma_{\mathbf{xx}} & \Sigma_{\mathbf{xu}} \\ \Sigma_{\mathbf{ux}} & \Sigma_{\mathbf{uu}} \end{pmatrix}.$$

Recall that in the online LQR problem, we deal with time-varying cost matrices $(\mathbf{Q}_t, \mathbf{R}_t)$, and for any $t \in [T]$, the above SDP yields different solutions. In fact, for any feasible solution $\Sigma$ of the above SDP, a strongly stable controller $\mathbf{K} = \Sigma_{\mathbf{xu}}^\top \Sigma_{\mathbf{xx}}^{-1}$ can be extracted. The steady-state covariance matrix induced by this controller is also feasible for the SDP and its cost is at most that of $\Sigma$ (see Theorem 4.2 in [4]). Moreover, for any (slowly-varying) sequence of feasible solutions to the SDP, the induced controller sequence is sequentially strongly-stable.

### E. Challenges of Distributed Online LQR for Unknown Dynamical Systems

The works of [4] and [46] tackle the centralized and decentralized online LQR, respectively. To keep the regret sublinear, the key idea in online LQR is to construct sequentially strongly stable controllers using online gradient descent (projected to the feasible set of SDP in (4)). However, in our work, given that system parameters $(\mathbf{A}, \mathbf{B})$ are unknown, the agents must perform a system identification first. The system identification step results in two challenges: (i) an exploration-exploitation trade-off to keep the regret sublinear, and (ii) different SDPs across agents as a result of decentralized estimation. The latter is particularly challenging, because as we can see in (4), each agent will only have a local estimate of $(\mathbf{A}, \mathbf{B})$, so the SDPs will have different feasible sets across agents, and we cannot directly apply distributed online optimization results on a common feasible set (e.g., [51], [52]). In this work, we propose an algorithm (in the next section) for which we prove that an extracted controller based on precise enough system estimates $(\widehat{\mathbf{A}}, \widehat{\mathbf{B}})$ is strongly stable w.r.t. the system $(\mathbf{A}, \mathbf{B})$.

## III. Algorithm and Theoretical Results

We now develop the distributed online LQR algorithm for unknown systems and study its theoretical regret bound.

### A. Algorithm

Our proposed method is outlined in Algorithm 1. In the first $T_0 + 1$ iterations, we need to collect data for the system identification. Suppose that each agent has access to a controller $\mathbf{K}_0$, which is $(\kappa_0, \gamma_0)$-strongly stable w.r.t. the system $(\mathbf{A}, \mathbf{B})$. This controller can be different across agents, but for the presentation simplicity, we assume that agents use the same controller $\mathbf{K}_0$. The knowledge of such controller is a common assumption in centralized LQR (see e.g., [7], [8]). In this period, agent $i$ at time $t$ applies the control $\mathbf{u}_{i,t} \sim \mathcal{N}(\mathbf{K}_0\mathbf{x}_{i,t}, 2\sigma^2\kappa_0^2 \cdot \mathbf{I})$, which prevents the state $\mathbf{x}_{i,t}$

from going unbounded (lines 2-7). For the next $T_1$ iterations, all agents perform the system identification step by solving a distributed least-squares (LS) problem. In this step, the global LS problem is formed using the data collected by all agents, where the local cost of each agent is only based on its own collected data. Here, we can use any iterative distributed optimization algorithm to get precise enough system estimates. We employ the **EXTRA** algorithm [9] since it achieves a geometric rate for strongly convex problems, and it can be implemented in a decentralized fashion (lines 8-16), where at each iteration, agents exchange their local system estimates with their neighbors in consistent with the network topology and then update their estimates accordingly. After $T_0+T_1+1$ iterations, each agent $i$ at time $t$ runs a distributed online gradient descent on the SDP (4), where the local cost is defined w.r.t. matrices $\mathbf{Q}_{i,t}$ and $\mathbf{R}_{i,t}$, and the feasible set is defined w.r.t. system estimates $(\widehat{\mathbf{A}}_{i,t}, \widehat{\mathbf{B}}_{i,t})$. This provides an iterative update where agent $i$ forms $\Sigma_{i,t+1}$ using its local cost matrices as well as $\Sigma_{j,t}$ for any $j$ in the neighborhood of $i$. A control matrix $\mathbf{K}_{i,t}$ is then extracted from the update of $\Sigma_{i,t}$ and is used to determine the action. In particular, $\mathbf{u}_{i,t}$ is sampled from a Gaussian distribution $\mathcal{N}(\mathbf{K}_{i,t}\mathbf{x}_{i,t}, \mathbf{V}_{i,t})$, which entails $\mathrm{E}[\mathbf{u}_{i,t}|\mathcal{F}_t] = \mathbf{K}_{i,t}\mathbf{x}_{i,t}$, where $\mathcal{F}_t$ is the smallest $\sigma$-field containing the information about all agents up to time $t$ (lines 17-26). The choice of $\mathbf{V}_{i,t}$ in line 23 is due to a technical reason. It ensures the fast convergence of the covariance matrix of $\mathbf{x}_{i,t}$ to the steady-state covariance matrix, when applying $\mathbf{K}_{i,t}$ to the underlying system $(\mathbf{A}, \mathbf{B})$.

### *B. Theoretical Result: Regret Bound*

Before presenting our theoretical result, let us state all the assumptions we use in our analysis as follows.

**Assumption 1.** *The cost matrices satisfy $Tr(\mathbf{Q}_{i,t}) \le C$ and $Tr(\mathbf{R}_{i,t}) \le C$, $\forall i \in [m]$ and $\forall t \in [T]$, where $C$ is a constant.*

**Assumption 2.** *The system matrices $(\mathbf{A}, \mathbf{B})$ have bounded norms, i.e., $\|[\mathbf{A}\ \mathbf{B}]\|_F \le \vartheta$.*

**Assumption 3.** *The covariance matrix of the noise ($\mathbf{W}$) satisfies $\mathbf{W} \succeq \sigma^2\mathbf{I}$ and $Tr(\mathbf{W}) \le \lambda^2$.*

**Assumption 4.** *The network structure is captured by a connected undirected graph. The communication matrix $\mathbf{P}$ is symmetric and doubly stochastic with a positive diagonal.*

**Assumption 5.** *We assume the knowledge of one $(\kappa_0, \gamma_0)$-strongly stable controller $\mathbf{K}_0$ w.r.t. system matrices $(\mathbf{A}, \mathbf{B})$ before the learning process.*

We now present our main theoretical result. By applying Algorithm 1, we show that for a multi-agent network of unknown LTI systems (with a connected communication graph), the individual regret of an arbitrary agent is upper-bounded by $O(T^{2/3}\log T)$, which implies that the agents *collectively* perform as well as the best fixed controller in hindsight for large enough $T$.

**Theorem 2.** *Suppose Assumptions (1, 2, 3, 4, 5) hold. Given $\kappa \ge 1$ and $0 \le \gamma < 1$, set $\nu = 2\kappa^4\lambda^2/\gamma$ and step size $\eta = T^{-1/3}$. If we run Algorithm 1 (denoted by $\mathcal{A}$) with $T_0 = T^{2/3}\log(T/\delta)$ and $T_1 = \Theta(\log(T))$, then with probability $(1-\delta)$, the individual regret of agent $j$ with respect to any*

**Algorithm 1** Distributed Online LQR Control with Unknown Dynamics

1: **Require:** number of agents $m$, doubly stochastic matrix $\mathbf{P} \in \mathrm{R}^{m\times m}$, parameter $\nu$, step size $\eta$, a $(\kappa_0, \gamma_0)$-strongly stable controller $\mathbf{K}_0$ w.r.t. system matrices $(\mathbf{A}, \mathbf{B})$, covariance parameter of the noise $\sigma$, parameter $\vartheta$.
**Initialize:** $\mathbf{x}_{i,1} = 0, \forall i \in [m]$.
2: **for** $t = 1, 2, \ldots, T_0+1$ **do**
3: **for** $i = 1, 2, \ldots, m$ **do**
4: Receive $\mathbf{x}_{i,t}$
5: Perform action $\mathbf{u}_{i,t} \sim \mathcal{N}(\mathbf{K}_0\mathbf{x}_{i,t}, 2\sigma^2\kappa_0^2 \cdot \mathbf{I})$
6: **end for**
7: **end for**
8: After the first $(T_0+1)$ iterations, each agent $i$ uses the collected data to form the local function

$$f_i(\mathbf{A}, \mathbf{B}) := \sum_{t=1}^{T_0} \|[\mathbf{A}\mathbf{B}]\mathbf{z}_{i,t} - \mathbf{x}_{i,t+1}\|^2 + \frac{\sigma^2\vartheta^{-2}}{m}\|[\mathbf{A}\mathbf{B}]\|_F^2,$$

where $\mathbf{z}_{i,t} = [\mathbf{x}_{i,t}^\top\ \mathbf{u}_{i,t}^\top]^\top$.
9: Choose the step size $\alpha$ following the result in [9] and set $\tilde{\mathbf{P}} := \frac{\mathbf{I}+\mathbf{P}}{2}$. Denote by $\widehat{D}_i$ the agent $i$ vectorized system estimate $[\widehat{\mathbf{A}}_i\ \widehat{\mathbf{B}}_i]$. Apply **EXTRA** to solve the global LS problem $\sum_{i=1}^m f_i(\mathbf{A}, \mathbf{B})$ in a distributed fashion.
10: Randomly generate $\widehat{D}_i^0$ for all $i \in [m]$.
11: $\forall i,\ \widehat{D}_i^1 = \sum_{j=1}^m [\mathbf{P}]_{ji}\widehat{D}_j^0 - \alpha\nabla f_i(\widehat{D}_i^0)$.
12: **for** $k = 0, 1, \ldots, T_1-1$ **do**
13: **for** $i = 1, 2, \ldots, m$ **do**
14: $\widehat{D}_i^{k+2} = \sum_{j=1}^m 2[\tilde{\mathbf{P}}]_{ji}\widehat{D}_j^{k+1} - \sum_{j=1}^m [\tilde{\mathbf{P}}]_{ji}\widehat{D}_j^k - \alpha[\nabla f_i(\widehat{D}_i^{k+1}) - \nabla f_i(\widehat{D}_i^k)]$.
15: **end for**
16: **end for**
17: For all $i \in [m]$, transform the vectorized $\widehat{D}_i^{T_1+1}$ back to matrix form $[\widehat{\mathbf{A}}_{i,t}\ \widehat{\mathbf{B}}_{i,t}]$ for all $t \ge (T_0+1)+T_1$.
18: Let $T_s := (T_0+T_1+2)$.
19: Initialize $\Sigma_{i,T_s} = \Sigma_{T_s}$ for any $i \in [m]$.
20: **for** $t = T_s, \ldots, T$ **do**
21: **for** $i = 1, 2, \ldots, m$ **do**
22: Receive $\mathbf{x}_{i,t}$
23: Compute $\mathbf{K}_{i,t} = (\Sigma_{i,t})_{\mathbf{ux}}(\Sigma_{i,t})_{\mathbf{xx}}^{-1}$ and $\mathbf{V}_{i,t} = (\Sigma_{i,t})_{\mathbf{uu}} - \mathbf{K}_{i,t}(\Sigma_{i,t})_{\mathbf{xx}}\mathbf{K}_{i,t}^\top$
24: Perform $\mathbf{u}_{i,t} \sim \mathcal{N}(\mathbf{K}_{i,t}\mathbf{x}_{i,t}, \mathbf{V}_{i,t})$ and observe $\mathbf{Q}_{i,t}, \mathbf{R}_{i,t}$
25: $\Sigma_{i,t+1} = \Pi_{\mathcal{S}_{t+1}^i}\left[\sum_{j=1}^m [\mathbf{P}]_{ji}\Sigma_{j,t} - \eta\begin{pmatrix}\mathbf{Q}_{i,t} & 0\\ 0 & \mathbf{R}_{i,t}\end{pmatrix}\right]$, where

$$\mathcal{S}_{t+1}^i := \left\{\Sigma \in \mathrm{R}^{n\times n} \middle| \begin{array}{l}\Sigma \succeq 0,\quad \mathrm{Tr}(\Sigma) \le \nu,\\ \Sigma_{\mathbf{xx}} = \widehat{\mathbf{C}}_{i,t+1}\Sigma\widehat{\mathbf{C}}_{i,t+1}^\top + \mathbf{W}\end{array}\right\},$$

and $\widehat{\mathbf{C}}_{i,t+1} = [\widehat{\mathbf{A}}_{i,t+1}\ \widehat{\mathbf{B}}_{i,t+1}]$.
26: **end for**
27: **end for**

*$(\kappa, \gamma)$-strongly stable controller $\mathbf{K}^s$ is bounded as follows*

$$\begin{aligned} \textit{Regret}_T^j(\mathcal{A}) &= J_T^j(\mathcal{A}) - J_T(\mathbf{K}^s) \\ &= O\Big(\sqrt{m}\big(\frac{n\kappa^{24}}{\gamma^6} + \frac{m\kappa^{12}}{(1-\beta)\gamma^3}\big)T^{2/3}\log(T/\delta)\Big), \end{aligned}$$

*for large enough $T$.*

From Theorem 2, we can see that the regret bound depends on the stability properties of the benchmark controller $\mathbf{K}^s$. In particular, a smaller $\kappa$ or a larger $\gamma$ entail a tighter regret bound. In fact, we prove that the local controller $\mathbf{K}_{i,t}$ (generated by Algorithm 1) is $(\frac{1}{\sqrt{2}\bar{\gamma}}, \frac{\bar{\gamma}}{2})$-strongly stable w.r.t. $(\mathbf{A}, \mathbf{B})$, where $\bar{\gamma} = O(\gamma/\kappa^4)$ (see Section VI-D in [10]). This implies that a smaller $\kappa$ or a larger $\gamma$ will also make the decentralized controller "more" strongly stable w.r.t. $(\mathbf{A}, \mathbf{B})$. As for the effect of the network topology, the dependence of $(1-\beta)^{-1}$ implies that when the network is well-connected (i.e., $\beta$ is smaller), the resulting bound is tighter. A smaller $\beta$ allows the Markov chain $\mathbf{P}$ to mix faster, which intuitively results in faster information propagation over the network of agents, and later in Section IV we verify this dependency by trying networks of different topologies. The exact expressions of the regret upper bound and the lower bound of $T$ are provided in the Appendix of [10], and we highlight the key technical challenges in Section III-C.

**Remark 1.** *For online LQR control with known dynamics, [4] and [46] prove regret bounds of $O(\sqrt{T})$ for centralized and distributed cases, respectively. However, in this work, since the system is unknown, agents need to compute system estimates first. This brings forward an exploration cost that increases the order of regret. In other words, agents objective is still to minimize the regret, but if they do not collect enough data, the estimation error propagates into the exploitation phase, yielding a larger regret (in terms of order).*

**Remark 2.** *For online control with unknown dynamics, both [47] and [43] consider the setup where costs are time-varying convex functions with adversarial noises, and they derive the regret bounds of $O(T^{2/3})$ for fully observable systems and partially observable systems, respectively. In this work, we consider the distributed variant of online LQR control with stochastic noises and unknown dynamics. Our regret bound of $O(T^{2/3}\log T)$ is consistent with previous results on centralized problems in the convex setting (disregarding the log factor).*

### C. Key Technical Challenges in the Proof

The regret can be decomposed into three terms, where each term must be small enough to bound the regret. In [4], two of these terms are bounded using the properties of strong stability and sequential strong stability, and one term is bounded using the standard regret bound for online gradient descent. In our setup, since $(\mathbf{A}, \mathbf{B})$ is *unknown*, the agents cannot work with the ideal feasible set in (4), and as evident from line 25 of the algorithm, $\mathcal{S}_{t+1}^i$ is constructed only based on agent $i$ system estimate. This brings forward two challenges. (i) Agent $i$ constructs the controllers based on iterates $\{\Sigma_{i,t}\}$ that are not necessarily in the feasible set of (4), so we need to establish the stability properties of these controllers. (ii) The feasible sets are different across agents (i.e., $\mathcal{S}_{t+1}^i \neq \mathcal{S}_{t+1}^j$ for $i \neq j$), so we cannot directly apply distributed online optimization results on a common feasible set.

To tackle the first challenge, we first derive the bound on the precision of each agent system estimate based on the **EXTRA** algorithm (see Lemma 6 in our arXiv version [10]) and combine that with statistical properties of centralized LS estimation using results of [8]. We then establish that if each agent system estimate is close enough to the true system, a strongly stable policy w.r.t. the system estimate is also strongly stable w.r.t. the true system (see Lemma 3 and Lemma 4 in our arXiv version [10]). To address the second challenge, we use alternating projections to prove that a point in the feasible set of one agent is close enough to its projection to the feasible set of another agent, when system estimates of these two agents are close. Then, in Theorem 9 of [10], we show the contribution of distributed online optimization to the regret. We finally put together these results to prove our regret bound.

## IV. Numerical Experiments

We now provide numerical simulations verifying the theoretical guarantee of our algorithm.

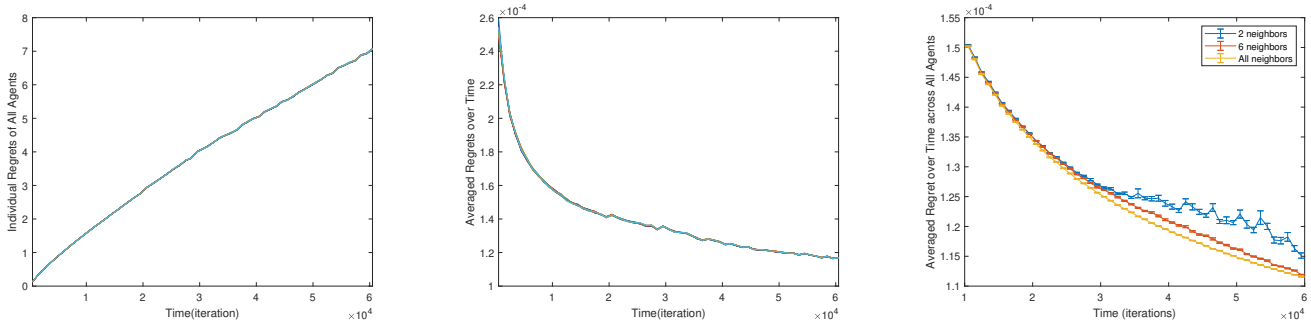
**Experiment Setup:** We first consider a network of $m = 20$ agents, captured by a cyclic graph, where each agent has a self-weight of 0.6 and assigns the weight 0.2 to each of its two neighbors. The (hyper)-parameters are set as follows: $d = k = 3$, $\kappa = 1.5$, $\gamma = 0.4$, $C = 300$. We let matrices $\mathbf{A} = (1-2\gamma)\mathbf{I}$ and $\mathbf{B} = (\gamma/\kappa)\mathbf{I}$ to ensure the existence of a $(\kappa, \gamma)$-strongly stable controller. For time-varying cost matrices, we set $\mathbf{Q}_{i,t}$ (respectively, $\mathbf{R}_{i,t}$) as a diagonal matrix where each diagonal term is sampled from the uniform distribution over $[0, C/d]$ (respectively, $[0, C/k]$), so that $\text{Tr}(\mathbf{Q}_{i,t}), \text{Tr}(\mathbf{R}_{i,t}) \leq C$. The disturbance $\mathbf{w}_{i,t}$ is sampled from a standard Gaussian distribution, and thus $\lambda^2 = d = 3$ and $\sigma^2 = 1$.

**Simulation:** We simulate Algorithm 1 for $T = 1K, 2K, 3K, \ldots, 60K$. For the benchmark, we set $\mathbf{K}^s$ as $(-\kappa)10^{-2}\mathbf{I}$ which is $(\kappa, \gamma)$-strongly stable w.r.t. $\mathbf{A}, \mathbf{B}$, and the resulting cumulative cost is small enough to be the benchmark. For the projection on the feasible set, we apply Dykstra's projection algorithm. Due to floating-point computations, $\mathbf{V}_{i,t}$ for action-sampling may not be positive semi-definite (PSD). Therefore, we address it by adding to $\mathbf{V}_{i,t}$ a small term $((1e-15)\mathbf{I})$ to keep it PSD. The entire process is repeated for 50 Monte-Carlo simulations, and in the figures we present the averaged plots.

| Iterations | 20K | 30K | 40K | 50K | 60K |
|---|---|---|---|---|---|
| Averaged Regret | 1.424 | 1.34 | 1.248 | 1.201 | 1.168 |
| Standard Error | 0.0087 | 0.0097 | 0.0052 | 0.0032 | 0.0037 |

TABLE I: The mean and standard error of the averaged regret over time and agents ($\times 10^{-4}$).

**Performance:** I) Sublinearity of Regret: To verify the result of Theorem 2, in Fig. 1b, we present the averaged regret over




(a) The plot of individual regrets of all agents over time.

(b) The temporal average of regret converges to zero for all agents.

(c) The averaged regrets over time for different networks: more connectivity results in smaller regret.

Fig. 1: The individual regrets of all agents are shown to be sublinear.

time (i.e., individual regret divided by $T$), which is clearly decreasing over time. In Table I, we tabulate the averaged regrets (over time and agents) as well as their standard errors computed from 50 trials for $T = 20K, 30K, 40K, 50K, 60K$. We can see that 50 trials is enough to obtain a small standard error. II) Impact of Network Topology: To study the impact of network topology, we use three different networks: a cyclic graph with 2 neighbors (Net A), a cyclic graph with 6 neighbors (Net B) and a complete graph (Net C) where every entry of $\mathbf{P}$ is $\frac{1}{20}$. From Fig. 1c, we can see that the regret increases when $\beta$ is smaller (Net A $>$ Net B $>$ Net C). This result is consistent with the impact of $\beta$ shown in Theorem 2.

## V. Conclusion

In this paper, we considered the distributed online LQR problem with unknown LTI systems and time-varying quadratic cost functions. We developed a fully decentralized algorithm to estimate the unknown system and minimize the finite-horizon cost, which can be cast as a regret minimization. We proved that the individual regret, which is the performance of the control sequence of any agent compared to the best (linear and strongly stable) controller in hindsight, is upper bounded by $O(T^{2/3}\log T)$. Future directions include analyzing the *dynamic* regret defined w.r.t. the optimal (instantaneous) control policy in hindsight, investigating *coupled* time-varying cost functions, and analyzing the adversarial noise setup.

## References


[1] B. D. O. Anderson, J. B. Moore, and B. P. Molinari, "Linear optimal control," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. SMC-2, no. 4, pp. 559–559, 1972.

[2] D. P. Bertsekas, "Dynamic programming and optimal control," 1995.

[3] K. Zhou, J. C. Doyle, K. Glover *et al.*, *Robust and optimal control*. Prentice hall New Jersey, 1996, vol. 40.

[4] A. Cohen, A. Hasidim, T. Koren, N. Lazic, Y. Mansour, and K. Talwar, "Online linear quadratic control," in *International Conference on Machine Learning (ICML)*, 2018, pp. 1029–1038.

[5] Y. Abbasi-Yadkori and C. Szepesvári, "Regret bounds for the adaptive control of linear quadratic systems," in *Annual Conference on Learning Theory (COLT)*. JMLR Workshop and Conference Proceedings, 2011, pp. 1–26.

[6] M. Ibrahimi, A. Javanmard, and B. V. Roy, "Efficient reinforcement learning for high dimensional linear quadratic systems," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2012, pp. 2636–2644.

[7] S. Dean, H. Mania, N. Matni, B. Recht, and S. Tu, "Regret bounds for robust adaptive control of the linear quadratic regulator," in *International Conference on Neural Information Processing Systems (NeurIPS)*, 2018, pp. 4192–4201.

[8] A. Cohen, T. Koren, and Y. Mansour, "Learning linear-quadratic regulators efficiently with only $\sqrt{T}$ regret," in *International Conference on Machine Learning (ICML)*. PMLR, 2019, pp. 1300–1309.

[9] W. Shi, Q. Ling, G. Wu, and W. Yin, "Extra: An exact first-order algorithm for decentralized consensus optimization," *SIAM Journal on Optimization*, vol. 25, no. 2, pp. 944–966, 2015.

[10] T.-J. Chang and S. Shahrampour, "Regret analysis of distributed online LQR control for unknown LTI systems," *arXiv preprint arXiv:2105.07310*, 2021.

[11] 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.

[12] A. Mosebach and J. Lunze, "Synchronization of autonomous agents by an optimal networked controller," in *European Control Conference (ECC)*, 2014, pp. 208–213.

[13] Y. Cao and W. Ren, "Optimal linear-consensus algorithms: An LQR perspective," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 40, no. 3, pp. 819–830, 2010.

[14] J. Jiao, H. L. Trentelman, and M. K. Camlibel, "A suboptimality approach to distributed linear quadratic optimal control," *IEEE Transactions on Automatic Control*, vol. 65, no. 3, pp. 1218–1225, 2020.

[15] ——, "Distributed linear quadratic optimal control: Compute locally and act globally," *IEEE Control Systems Letters*, vol. 4, no. 1, pp. 67–72, 2020.

[16] S. Alemzadeh and M. Mesbahi, "Distributed q-learning for dynamically decoupled systems," in *American Control Conference (ACC)*, 2019, pp. 772–777.

[17] S. Fattahi, N. Matni, and S. Sojoudi, "Efficient learning of distributed linear-quadratic control policies," *SIAM Journal on Control and Optimization*, vol. 58, no. 5, pp. 2927–2951, 2020.

[18] L. Furieri, Y. Zheng, and M. Kamgarpour, "Learning the globally optimal distributed LQ regulator," in *Learning for Dynamics and Control (L4DC)*, 2020, pp. 287–297.

[19] L. Furieri and M. Kamgarpour, "First order methods for globally optimal distributed controllers beyond quadratic invariance," in *American Control Conference (ACC)*, 2020, pp. 4588–4593.

[20] K. J. Åström and P. Eykhoff, "System identification—a survey," *Automatica*, vol. 7, no. 2, pp. 123–162, 1971.

[21] L. Ljung, "System identification," *Wiley encyclopedia of electrical and electronics engineering*, pp. 1–19, 1999.

[22] H.-F. Chen and L. Guo, *Identification and stochastic adaptive control*. Springer Science & Business Media, 2012.

[23] G. C. Goodwin, "Dynamic system identification: experiment design and data analysis," *Mathematics in science and engineering*, vol. 136, 1977.

[24] 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.

[25] M. Simchowitz, H. Mania, S. Tu, M. I. Jordan, and B. Recht, "Learning without mixing: Towards a sharp analysis of linear system identification," in *Conference On Learning Theory (COLT)*. PMLR, 2018, pp. 439–473.

[26] T. Sarkar and A. Rakhlin, "Near optimal finite time identification of arbitrary linear dynamical systems," in *International Conference on Machine Learning (ICML)*. PMLR, 2019, pp. 5610–5618.

[27] S. Oymak and N. Ozay, "Non-asymptotic identification of LTI systems from a single trajectory," in *American control conference (ACC)*, 2019, pp. 5655–5661.

[28] T. Sarkar, A. Rakhlin, and M. A. Dahleh, "Finite time LTI system identification," *Journal of Machine Learning Research*, vol. 22, pp. 1–61, 2021.

[29] A. Tsiamis and G. J. Pappas, "Finite sample analysis of stochastic system identification," in *IEEE Conference on Decision and Control (CDC)*, 2019, pp. 3648–3654.

[30] M. Simchowitz, R. Boczar, and B. Recht, "Learning linear dynamical systems with semi-parametric least squares," in *Conference on Learning Theory (COLT)*. PMLR, 2019, pp. 2714–2802.

[31] S. Fattahi, "Learning partially observed linear dynamical systems from logarithmic number of samples," in *Learning for Dynamics and Control (L4DC)*. PMLR, 2021, pp. 60–72.

[32] M. Fazel, R. Ge, S. Kakade, and M. Mesbahi, "Global convergence of policy gradient methods for the linear quadratic regulator," in *International Conference on Machine Learning (ICML)*, 2018, pp. 1467–1476.

[33] D. Malik, A. Pananjady, K. Bhatia, K. Khamaru, P. Bartlett, and M. Wainwright, "Derivative-free methods for policy optimization: Guarantees for linear quadratic systems," in *International Conference on Artificial Intelligence and Statistics (AISTATS)*. PMLR, 2019, pp. 2916–2925.

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

[35] H. Mohammadi, M. Soltanolkotabi, and M. R. Jovanovic, "Random search for learning the linear quadratic regulator," in *American Control Conference (ACC)*, 2020, pp. 4798–4803.

[36] A. Cassel, A. Cohen, and T. Koren, "Logarithmic regret for learning linear quadratic regulators efficiently," in *International Conference on Machine Learning (ICML)*. PMLR, 2020, pp. 1328–1337.

[37] M. Simchowitz and D. Foster, "Naive exploration is optimal for online LQR," in *International Conference on Machine Learning (ICML)*. PMLR, 2020, pp. 8937–8948.

[38] S. Lale, K. Azizzadenesheli, B. Hassibi, and A. Anandkumar, "Reinforcement learning with fast stabilization in linear dynamical systems," in *International Conference on Artificial Intelligence and Statistics (AISTATS)*. PMLR, 2022, pp. 5354–5390.

[39] E. Hazan, K. Singh, and C. Zhang, "Learning linear dynamical systems via spectral filtering," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2017, pp. 6702–6712.

[40] S. Arora, E. Hazan, H. Lee, K. Singh, C. Zhang, and Y. Zhang, "Towards provable control for unknown linear dynamical systems," 2018.

[41] N. Agarwal, B. Bullins, E. Hazan, S. M. Kakade, and K. Singh, "Online control with adversarial disturbances," in *International Conference on Machine Learning (ICML)*, 2019, pp. 154–165.

[42] N. Agarwal, E. Hazan, and K. Singh, "Logarithmic regret for online control," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2019, pp. 10 175–10 184.

[43] M. Simchowitz, K. Singh, and E. Hazan, "Improper learning for non-stochastic control," in *Conference on Learning Theory (COLT)*. PMLR, 2020, pp. 3320–3436.

[44] C. Yu, G. Shi, S.-J. Chung, Y. Yue, and A. Wierman, "The power of predictions in online control," *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 33, 2020.

[45] R. Zhang, Y. Li, and N. Li, "On the regret analysis of online LQR control with predictions," in *American Control Conference (ACC)*, 2021, pp. 697–703.

[46] T.-J. Chang and S. Shahrampour, "Distributed online linear quadratic control for linear time-invariant systems," in *American Control Conference (ACC)*, 2021, pp. 923–928.

[47] E. Hazan, S. Kakade, and K. Singh, "The nonstochastic control problem," in *Algorithmic Learning Theory (ALT)*, 2020, pp. 408–421.

[48] S. Lale, K. Azizzadenesheli, B. Hassibi, and A. Anandkumar, "Logarithmic regret bound in partially observable linear dynamical systems," *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 33, pp. 20 876–20 888, 2020.

[49] J. S. Liu, *Monte Carlo strategies in scientific computing*. Springer Science & Business Media, 2008.

[50] G. Guo, Y. Zhao, and G. Yang, "Cooperation of multiple mobile sensors with minimum energy cost for mobility and communication," *Information Sciences*, vol. 254, pp. 69–82, 2014.

[51] F. Yan, S. Sundaram, S. Vishwanathan, and Y. Qi, "Distributed autonomous online learning: Regrets and intrinsic privacy-preserving properties," *IEEE Transactions on Knowledge and Data Engineering*, vol. 25, no. 11, pp. 2483–2493, 2012.

[52] S. Shahrampour and A. Jadbabaie, "Distributed online optimization in dynamic environments using mirror descent," *IEEE Transactions on Automatic Control*, vol. 63, no. 3, pp. 714–725, 2018.