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 multilagent network where each agent is modeled as an 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 1) estimate the unknown dynamics and 2) 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 semidefinite programming whose feasible set is based on the agent system estimator. 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.

Index Terms—Distributed control, online optimization, robust control, regret minimization, system identification.

I. INTRODUCTION

In recent years, there has been a significant interest in problems arising at the interface of control and machine learning. Among classical control problems, linear quadratic regulator (LQR) control [1], [2], [3] is a prominent point in case. LQR control centers around linear time-invariant (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. 1) 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]). 2) 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 an 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 Section 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], [6], [7], [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 the 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 suboptimality 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 sublinear in $T$, which implies that its time-averaged performance tends to be that of the best policy in hindsight asymptotically. In practice, this setting can be applied for modeling the energy consumption in mobile sensor networks (MSNs) 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 semidefinite programming (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 tradeoff, the main 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 the 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 [10, Sec. VI].

A. Related Literature

1) Distributed LQR Control: Distributed LQR has been widely studied in the control literature. A number of works focus on multiagent 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. Mosebach and Lunze [12] derive the necessary condition for an optimal distributed controller design, resulting in a nonconvex optimization problem. Cao and Ren [13] address a multiagent network, where the dynamics of each agent is a single integrator. Cao and Ren [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] propose sufficient conditions to obtain suboptimal 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 subsystems. 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). Furieri and Kamgarpour [19] investigate 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], [21], [22], [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 (LS) 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], [28], [29], [30]), and Fattahi [31] improves the sample complexity to polylogarithmic.

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 the following:

1) gradient estimation (e.g., [32], [33], [34], [35]);

2) the estimation of dynamics matrices and derivation of the controller by considering the estimation uncertainty (e.g., [7], [8], [24], [36],[37], [38]);

3) 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 the following scenarios.

1) 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 semideviatorial 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 $O(\sqrt{T})$.

2) 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, Simchowitz et al. [43] address the cases of a) convex functions with adversarial noises and b) strongly-convex functions with semideviatorial 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

| Symbol | Description |
|--------|-------------|
| $\mathbb{R}^n$ | Set \{1, 2, ..., n\} for any integer n. |
| $\text{Tr}(\cdot)$ | Trace operator. |
| $\|\cdot\|_2$ | Euclidean (spectral) norm of a vector (matrix). |
| $\|\cdot\|_F$ | Frobenius norm of a matrix. |
| $\mathbb{E}$ | Expectation operator. |
| $\Pi_\mathcal{S}$ | Operator for the projection to set $\mathcal{S}$. |
| $A_{ij}$ | Entry in the $i$-th row and $j$-th column of $A$. |
| $A_i$ | $i$-th column of $A$. |
| $A \cdot B$ | $\text{Tr}(A^T B)$. |
| $A \succeq B$ | $(A - B)$ is positive semi-definite. |
| $\mathbb{1}$ | Vector of all ones. |
| $e_i$ | $i$-th basis vector. |
| $\text{vec}(A)$ | Vectorized version of the matrix $A$. |
B. Distributed Online LQR Control With Unknown Dynamics

We consider a multiagent network of $m$ LTI systems, where the dynamics of agent $i$ is given as
\[ x_{i,t+1} = Ax_{i,t} + Bu_{i,t} + w_{i,t}, \quad i \in [m] \]
and $x_{i,t} \in \mathbb{R}^d$ and $u_{i,t} \in \mathbb{R}^k$ represent agent $i$ state and control (or action) at time $t$, respectively. Furthermore, $A \in \mathbb{R}^{d \times d}$, $B \in \mathbb{R}^{d \times k}$, and $w_{i,t}$ is Gaussian noise with zero mean and covariance $W \succeq \sigma^2 I$. The system parameters $(A, B)$ are unknown to all agents and need to be estimated. The noise sequence $(w_{i,t})$ is independent over time and agents. We also assume that $\| [A B]\| \leq \theta$ 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 $x_{i,t}$ and applies the action $u_{i,t}$. Then, positive semidefinite cost matrices $Q_{i,t}$ and $R_{i,t}$ are revealed, and the agent incurs the cost $x_{i,t}^\top Q_{i,t} x_{i,t} + u_{i,t}^\top R_{i,t} u_{i,t}$. Throughout this article, we assume that $\text{Tr}(Q_{i,t})$, $\text{Tr}(R_{i,t}) \leq C$ for all $i$, and some $C > 0$. Agent $i$ follows a policy that selects the control $u_{i,t}$ based on the observed cost matrices $Q_{i,t}$, $R_{i,t}$, and $R_{i,1}, \ldots, R_{i,t-1}$, 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 article, we focus on the finite-horizon problem, where for a centralized policy $\pi$, the cost after $T$ steps is given as
\[ J_T(\pi) = \mathbb{E} \left[ \sum_{t=1}^{T} x_{i,t}^\top Q_{i,t} x_{i,t} + u_{i,t}^\top R_{i,t} u_{i,t} \right], \tag{1} \]
where $Q_{i} = \sum_{t=1}^{m} Q_{i,t}$ and $R_{i} = \sum_{t=1}^{m} R_{i,t}$, and the expectation is over the possible randomness of the policy as well as the noise. The superscript $\pi$ in $u_{\pi,t}$ and $x_{\pi,t}$ 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 $(Q, R)$, where the goal is to minimize $\lim_{T \to \infty} J_T(\pi)/T$, it is well known that for a controllable LTI system $(A, B)$, the optimal policy is given by the constant linear feedback, i.e., $u_{\pi,t} = K x_{\pi,t}$ for a matrix $K \in \mathbb{R}^{k \times d}$.

Regret Definition: The goal of a distributed online LQR algorithm $A$ is to mimic the performance of an ideal centralized algorithm that solves (1). The main two challenges are 1) the online nature of the problem, where cost matrices become available sequentially, and 2) the distributed setup, where agent $i$ only receives information about the sequence $(Q_{i,t}, R_{i,t})$ while the network cost is based on $(Q, R)$. In this setting, each agent $j$ locally generates the control sequence $(u_{j,t})_{t=1}^{T}$, which 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:
\[ \text{Regret}_j(A) := J_{T}^j(A) - \min_{\pi \in \Pi} J_T(\pi) \tag{2} \]
for agent $j \in [m]$, where
\[ J_{T}^j(A) = \mathbb{E} \left[ \sum_{t=1}^{T} x_{j,t}^\top Q_{j,t} x_{j,t} + u_{j,t}^\top R_{j,t} u_{j,t} \right]. \tag{3} \]
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 tradeoff between exploration (estimating the system parameters) and exploitation (keeping the regret sublinear).

Network Structure: The underlying network topology is captured by an undirected graph $G = (\mathcal{V}, E)$, where $\mathcal{V} = [m]$ denotes the set of nodes (i.e., agents) and $E$ represents the set of edges. If there is an edge between nodes $i$ and $j$, agent $i$ assigns a positive weight $|P_{ij}|$ to the information received from agent $j$. If there is no edge between nodes $i$ and $j$, $|P_{ij}|$ is equal to zero. The weighted adjacency matrix $P$ is assumed to be symmetric and doubly stochastic, i.e., all elements of $P$ are nonnegative and $\sum_{i=1}^{m} |P_{ij}| = \sum_{j=1}^{m} |P_{ij}| = 1$. The network is further assumed to be connected, i.e., for any two agents $i, j \in [m]$, there is a (potentially multihop) path from $i$ to $j$. We also assume $P$ has a positive diagonal. Then, there exists a geometric mixing bound for $P$ [49], such that $\sum_{j=1}^{m} |P_{ij}| = 1/m \leq \sqrt{m} \beta$, $i \in [m]$, where $\beta$ is the second largest singular value of $P$. 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 $P$. We elaborate on this in the algorithm description.

Example 1: Our framework can be used for minimizing the energy consumption in MSNs [50] in time-varying settings. Consider an MSN where, at time $t$, the total mobility cost (or budget) of sensors is modeled by matrices $(Q_t, R_t)$. Each agent $i$ has a local budget of $(Q_{i,t}, R_{i,t})$, 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), although each sensor only has local information.

C. Strong Stability and Sequential Strong Stability

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

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

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 $x_{i+1} = Ax_i + Bu_i + w_i$, if a $(\kappa, \gamma)$-strongly stable policy $K$ is applied $(u_i = K x_i)$, $X_i$ (the state covariance matrix of $x_i$) converges to $X$ (the steady-state covariance matrix) with the following exponential rate:
\[ \| \hat{X}_i - X \| \leq \kappa^2 e^{-2\gamma t} \| \hat{X}_0 - X \|. \]

See [4, Lemma 3.2] 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 $(K_{t})_{t=1}^{T}$ is $(\kappa, \gamma)$-strongly stable if there exist matrices $(H_t)_{t=1}^{T}$ and $(L_t)_{t=1}^{T}$ such that $A + BK_t = HL_tH_t^{-1}$ for all $t$ with the following properties.
1) $|L_t| \leq 1 - \gamma$ and $|K_t| \leq \kappa$.
2) $|H_t| \leq \beta'$ and $|H_t^{-1}| \leq 1/\alpha'$ with $\kappa = \beta'/\alpha'$.
3) $|H_{t+1}^{-1}H_t| \leq 1 + \gamma/2$.

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

\[ \text{Lemma 1: (Lemma 3.5 in [4]) Suppose a time-varying policy } (u_t = K_t x_t) \text{ is applied. Denote the steady-state covariance matrix of } K_t \text{ as } X_t. \text{ If } \{K_t\} \text{ are } (\kappa, \gamma)-sequentially strongly stable and } \|X_t - X_{t-1}\| \leq \tau, X_t \text{ (the steady-state covariance matrix of } x_t) \text{ converges to } X_t \text{ as follows:} \]

\[ \|X_{t+1} - X_t\| \leq \kappa^2 e^{-\gamma t} \|X_1 - X_0\| + \frac{2\eta \kappa^2}{\gamma}. \]

\[ \text{D. SDP Relaxation for LQR Control} \]

For the following dynamical system:

\[ x_{t+1} = Ax_t + Bu_t + w_t, \quad w_t \sim \mathcal{N}(0, W) \]

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

\[ \begin{align*}
    \text{minimize} & \quad J(\Sigma) = \begin{pmatrix} Q & 0 \\ 0 & R \end{pmatrix} \Sigma \begin{pmatrix} Q & 0 \\ 0 & R \end{pmatrix}^T \\
    \text{subject to} & \quad \Sigma_{xx} = [A \ B]\Sigma[A \ B]^T + W \\
    & \quad \Sigma \succeq 0, \quad \text{Tr}(\Sigma) \leq \nu
\end{align*} \]

where

\[ \Sigma = \begin{pmatrix} \Sigma_{xx} & \Sigma_{xu} \\ \Sigma_{ux} & \Sigma_{uu} \end{pmatrix}. \]

Recall that in the online LQR problem, we deal with time-varying cost matrices \( (Q_t, R_t) \), 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 \( K = \Sigma_{xx}^{-1} \Sigma_{xu}^{-1} \) 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 [4, Th. 4.2]). 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

Cohen et al. [4] and Chang and Shahrampour [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 \( (A, B) \) are unknown, the agents must perform a system identification first. The system identification step results in two challenges: 1) an exploration-exploitation tradeoff to keep the regret sublinear, and 2) 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 \( (A, B) \), 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 \( (A, B) \) is strongly stable w.r.t. the system \( (A, B) \).

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 \( K_0 \), which is \( (\kappa_0, \gamma_0) \)-strongly stable w.r.t. the system \( (A, B) \). This controller can be different across agents but for the presentation simplicity, we assume that agents use the same controller \( K_0 \). The knowledge of such a controller is a common assumption in centralized LQR (see, e.g., [7], [8]). In this period, agent \( i \) at time \( t \) applies the control \( u_{i,t} \sim \mathcal{N}(K_0 x_{i,t}, 2\sigma^2 \gamma_0 I) \), which prevents the state \( x_{i,t} \) from going unbounded (lines 2–7). For the next \( T_1 \) iterations, all agents perform the system identification step by solving a distributed 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 \( Q_{i,t} \) and \( R_{i,t} \), and the feasible set is defined w.r.t. system estimates \( (\hat{A}_{i,t}, \hat{B}_{i,t}) \). 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 \( K_{i,t} \) is then extracted from the update of \( \Sigma_{i,t} \) and is used to determine the action. In particular, \( u_{i,t} \) is sampled from a Gaussian distribution \( \mathcal{N}(K_{i,t} x_{i,t}, \mathcal{V}_{i,t}) \), which entails \( \mathbb{E}[u_{i,t}|F_t] = K_{i,t} x_{i,t} \), where \( F_t \) is the smallest \( \sigma \)-field containing the information about all agents up to time \( t \) (lines 17–26). The choice of \( \mathcal{V}_{i,t} \) in line 23 is due to a technical reason. It ensures the fast convergence of the covariance matrix of \( x_{i,t} \) to the steady-state covariance matrix, when applying \( K_{i,t} \) to the underlying system \( (A, B) \).

B. Theoretical Result: Regret Bound

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

\[ \text{Assumption 1: The cost matrices satisfy } \text{Tr}(Q_{i,t}) \leq C \text{ and } \text{Tr}(R_{i,t}) \leq C, \forall i \in [m] \text{ and } \forall t \in [T], \] where \( C \) is a constant.

\[ \text{Assumption 2: The system matrices } (A, B) \text{ have bounded norms, i.e., } ||[A \ B]||_F \leq \theta. \]

\[ \text{Assumption 3: The covariance matrix of the noise } (W) \text{ satisfies } W \succeq \sigma^2 I \text{ and } \text{Tr}(W) \leq \lambda^2. \]

\[ \text{Assumption 4: The network structure is captured by a connected undirected graph. The communication matrix } P \text{ is symmetric and doubly stochastic with a positive diagonal.} \]

\[ \text{Assumption 5: We assume the knowledge of one } (\kappa_0, \gamma_0) \text{-strongly stable controller } K_0 \text{ w.r.t. system matrices } (A, B) \text{ before the learning process.} \]

We now present our main theoretical result. By applying Algorithm 1, we show that for a multiagent 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
Algorithm 1: Distributed Online LQR Control With Unknown Dynamics.

1: \textbf{Require:} number of agents \(m\), doubly stochastic matrix \(P \in \mathbb{R}^{n \times m}\), parameter \(\nu\), step size \(\eta\), a \((\kappa_0, \gamma_0)\)-strongly stable controller \(K_0\) w.r.t. system matrices \((A, B)\), covariance parameter of the noise \(\sigma_x\), parameter \(\vartheta\).

2: \textbf{Initialize:} \(x_{i,0} = 0 \forall i \in [m]\).

3: for \(t = 1, 2, \ldots, T_0 + 1\) do

4: \hspace{1em} Receive \(x_{i,t}\).

5: \hspace{1em} Perform action \(u_{i,t} \sim \mathcal{N}(K_0x_{i,t}, 2\sigma^2\kappa^2_0\cdot I)\)

6: \textbf{end for}

7: \textbf{end for}

8: After the first \((T_0 + 1)\) iterations, each agent \(i\) uses the collected data to form the local function

\[f_i(A, B) := \sum_{t=1}^{T_0} \left\|\left[A|B\right]z_{i,t} - x_{i,t+1}\right\|^2 + \frac{\sigma^2\beta^2}{m} \left\|\left[A|B\right]\right\|^2\]

where \(z_{i,t} = [x_{i,t}^\top u_{i,t}^\top]^\top\).

9: Choose the step size \(\alpha\) following the result in [9] and set \(P := \frac{1}{\alpha}P\). Denote by \(\hat{D}\), the agent \(i\) vectorized system estimate \([\hat{A}_i, \hat{B}_i]\). Apply EXTRA to solve the global LS problem \(\sum_{i=1}^{m} f_i(A, B)\) in a distributed fashion.

10: Randomly generate \(D_i^0\) for all \(i \in [m]\).

11: \forall i, \(D_i^1 = \sum_{j=1}^{m} [P_ij]D_j^0 - \alpha V f_i(D_i^0)\).

12: for \(k = 0, 1, \ldots, T_1 - 1\) do

13: \hspace{1em} for \(i = 1, 2, \ldots, m\) do

14: \hspace{2em} \(D_i^{k+2} = \sum_{j=1}^{m} [P_i]_{j,i} D_j^{k+1} - \sum_{j=1}^{m} [P_i]_{j,i} D_j^{k} - \alpha (\nabla f_i(D_i^{k+1}) - \nabla f_i(D_i^{k}))\).

15: \hspace{1em} \textbf{end for}

16: \textbf{end for}

17: For all \(i \in [m]\), transform the vectorized \(\hat{D}_i^{k+1}\) back to matrix form \([\hat{A}_{i,k}, \hat{B}_{i,k}]\) for all \(t \geq (T_0 + 1) + T_1\).

18: Let \(T_x := (T_0 + T_1 + 2)\).

19: Initialize \(\Sigma_{x,T_0} = \Sigma_{x_T}\) for each \(i \in [m]\).

20: for \(t = T_0, \ldots, T_d\) do

21: \hspace{1em} for \(i = 1, 2, \ldots, m\) do

22: \hspace{2em} Receive \(x_{i,t}\).

23: \hspace{2em} Compute \(K_{i,t} = (\Sigma_{x,t})_{xx}(\Sigma_{x,t})_{xx}^{-1}\) and \(V_{i,t} = (\Sigma_{x,t})_{uu} - K_{i,t}(\Sigma_{x,t})_{xx}K_{i,t}^\top\).

24: \hspace{2em} Perform \(u_{i,t} \sim \mathcal{N}(K_{i,t}x_{i,t}, V_{i,t})\) and observe \(Q_{i,t}, R_{i,t}\).

25: \hspace{2em} \(\Sigma_{x,t+1} = \Pi_{S_{t+1}} \left[\sum_{j=1}^{m} [P]_{j,i} \Sigma_{i,j,t} - \eta \left( \begin{array}{cc} Q_{i,t} & 0 \\ 0 & R_{i,t} \end{array} \right) \right], \)

where \(S_{t+1} := \left\{ \Sigma \in \mathbb{R}^{n \times n} : \Sigma_{xx} \succeq 0, \text{Tr}(\Sigma) \leq \nu, \Sigma_{xx} = \hat{C}_{i,t+1}\Sigma_{i,t+1} + W \right\}\), and \(\hat{C}_{i,t+1} = [\hat{A}_{i,t+1} \hat{B}_{i,t+1}]\).

26: \hspace{1em} \textbf{end for}

27: \textbf{end for}

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 \(\kappa \geq 1\) and \(0 \leq \gamma < 1\), set \(\nu = 2\kappa^2\gamma^2/\gamma\) and step size \(\eta = T^{-1/3}\). If we run Algorithm 1 (denoted by \(A\)) 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 \((\kappa, \gamma)\)-strongly stable controller \(K^*\) is bounded as follows:

\[\text{Regret}_{j}(A) = J_I^j(A) - J_I^j(K^*) = O \left( \sqrt{m} \left( n\kappa^2 \frac{4}{\gamma^2} + \frac{m\kappa^2}{(1 - \beta)^3} \right) T^{2/3} \log(T/\delta) \right)\]

for large enough \(T\).

From Theorem 2, we can see that the regret bound depends on the stability properties of the benchmark controller \(K^\circ\). In particular, a smaller \(\kappa\) or a larger \(\gamma\) entails a tighter regret bound. In fact, we prove that the local controller \(K_{i,t}\) (generated by Algorithm 1) is \((\frac{\kappa}{\kappa - \frac{1}{2}})\)-strongly stable w.r.t. \((A, B)\), where \(\gamma = O(\kappa^2)\) (see [10, Sec. VI-D]). This implies that a smaller \(\kappa\) or a larger \(\gamma\) will also make the decentralized controller “more” strongly stable w.r.t. \((A, B)\). As 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 \(P\) to mix faster, which intuitively results in faster information propagation over the network of agents, and later in Section IV, we verify this dependence by trying networks of different topologies. The exact expressions of the regret upper bound and the lower bound of \(T\) are provided in [10, Appendix], and we highlight the key technical challenges in Section III-C.

Remark 1: For online LQR control with known dynamics, Cohen et al. [4] and Chang and Shahrampour [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, the authors in [43] and [47] 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 \((A, B)\) is unknown, the agents cannot work with the ideal feasible set in (4), and as evident from line 25 of the algorithm, \(S_{t+1}\) is constructed only based on agent \(i\) system estimate. This brings forward two challenges. 1) 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. 2) The feasible sets are different across agents (i.e., \(S_{i+1} \neq S_{j+1}\) 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 [10, Lemma 6]) and combine that with statistical properties of centralized LS estimation using results in [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 [10, Lemmas 3 and 4]). 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 [10, Th. 9], 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 of 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 $A = (1 - 2\gamma)I$ and $B = (\gamma/\kappa)I$ to ensure the existence of a $(\kappa, \gamma)$-strongly stable controller. For time-varying cost matrices, we set $Q_{i,t}$ (respectively, $R_{i,t}$) as a diagonal matrix where each diagonal term is sampled from the uniform distribution over $[0, C/d]$ (respectively, $[0, C/k]$), so that $\text{Tr}(Q_{i,t})$, $\text{Tr}(R_{i,t}) \leq C$. 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 $K^*$ as $(-\kappa)10^{-2}I$, which is $(\kappa, \gamma)$-strongly stable w.r.t. $A$, $B$, 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, $V_{i,t}$ for action-sampling may not be positive semidefinite (PSD). Therefore, we address it by adding to $V_{i,t}$, a small term $((1e - 15)I)$ to keep it PSD. The entire process is repeated for 50 Monte Carlo simulations, and in the figures, we present the average plots.

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

| 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 |

From Fig. 1(c), 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 article, 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 Trans. Syst. Man, Cybern., vol. TSMC-2, no. 4, pp. 559–559, Sep. 1972.
[2] D. P. Bertsekas, “Dynamic programming and optimal control: Volume I,” Athena Scientific, vol. 4, 2012.
[3] K. Zhou et al., “Robust and Optimal Control,” vol. 40. Hoboken, NJ, USA: Prentice-Hall, 1996.
[4] A. Cohen, A. Hasidim, T. Koren, N. Lazic, Y. Mansour, and K. Talwar, “Online linear quadratic control,” in Proc. Int. Conf. Mach. Learn., 2018, pp. 1029–1038.
[5] Y. Abbasi-Yadkori and C. Szepesvári, “Regret bounds for the adaptive control of linear quadratic systems,” in Proc. Annu. Conf. Learn. Theory JMLR Workshop Conf. Proc., 2011, pp. 1–26.
[6] M. Ibrahimi, A. Javanmard, and B. V. Roy, “Efficient reinforcement learning for high dimensional linear quadratic systems,” in Proc. Adv. Neural Inf. Process. Syst., 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 Proc. Int. Conf. Neural Inf. Process. Syst., 2018, pp. 4192–4201.
[8] A. Cohen, T. Koren, and Y. Mansour, “Learning linear-quadratic regulators efficiently with only $\sqrt{T}$ regret,” in Proc. Int. Conf. Mach. Learn., 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 J. Optim., vol. 25, no. 2, pp. 944–996, 2015.

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

[11] F. Borrelli and T. Keviczky, “Distributed LQR design for identical dynamically decoupled systems,” IEEE Trans. Autom. Control, vol. 53, no. 8, pp. 1901–1912, Sep. 2008.

[12] A. Mosebach and J. Lunze, “Synchronization of autonomous agents by an optimal networked controller,” in Proc. IEEE Euro Control Conf., 2014, pp. 208–213.

[13] Y. Cao and W. Ren, “Optimal linear-consensus algorithms: An LQR perspective,” IEEE Trans. Syst., Man, Cybern., Part B (Cybern.), vol. 40, no. 3, pp. 819–830, Jun. 2010.

[14] J. Jiao, H. L. Trentelman, and M. K. Camlibel, “A suboptimality approach to distributed linear quadratic optimal control,” IEEE Trans. Autom. Control, vol. 65, no. 3, pp. 1218–1225, Mar. 2020.

[15] J. Jiao, H. L. Trentelman, and M. K. Camlibel, “Distributed linear quadratic optimal control: Compute locally and act globally,” IEEE Control Syst. Lett., vol. 4, no. 1, pp. 67–72, Jan. 2020.

[16] M. Simchowitz and M. Mesbahi, “Distributed q-learning for dynamically decoupled systems,” in Proc. IEEE Amer. Control Conf., 2019, pp. 772–777.

[17] S. Fattahi, N. Matni, and S. Sojoudi, “Efficient learning of distributed linear-quadratic control policies,” SIAM J. Control Optim., vol. 58, no. 3, pp. 2927–2951, 2020.

[18] L. Furieri, Y. Zheng, and M. Kamgourpour, “Learning the globally optimal distributed LQ regulator,” in Proc. Learn. Dyn. Control, 2020, pp. 287–297.

[19] L. Furieri and M. Kamgourpour, “First order methods for globally optimal distributed controllers beyond quadratic invariance,” in Proc. IEEE Amer. Control Conf., 2020, pp. 4588–4593.

[20] K. J. Aström and P. Eykhoff, “System identification—A survey,” Automatica, vol. 7, no. 2, pp. 123–161, 1971.

[21] L. Ljung, “System identification,” in Wiley Encyclopedia of Electrical and Electronics Engineering. New York, NY, USA: Wiley, pp. 1–19, 1999.

[22] H.-F. Chen and L. Guo, Identification and Stochastic Adaptive Control. Berlin, Germany: Springer, 2012.

[23] G. C. Goodwin, “Dynamic system identification: Experiment design and data analysis,” Math. Sci. Eng., vol. 136, pp. iii–x, 1–291, 1977.

[24] S. Dean, H. Mania, N. Matni, B. Recht, and S. Tu, “On the sample complexity of the linear quadratic regulator,” Found. Comput. Math., vol. 20, no. 4, pp. 633–679, 2020.

[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 Proc. Conf. Learn. Theory, 2018, pp. 439–473.

[26] T. Sarkar and A. Rahklin, “Near optimal finite time identification of arbitrary linear dynamical systems,” in Proc. Int. Conf. Mach. Learn., 2019, pp. 5610–5618.

[27] S. Oymak and N. Ozay, “Non-asymptotic identification of LTI systems from a single trajectory,” in Proc. IEEE Amer. Control Conf., 2019, pp. 5655–5660.

[28] T. Sarkar, A. Rahklin, and M. A. Dahleh, “Finite time LTI system identification,” J. Mach. Learn. Res., vol. 22, pp. 1–61, 2021.

[29] A. Tsiamis and G. J. Pappas, “Finite sample analysis of stochastic system identification,” in Proc. IEEE Conf. Decis. Control, 2019, pp. 3648–3654.

[30] M. Simchowitz, R. Boczar, and B. Recht, “Learning linear dynamical systems with semi-parametric least squares,” in Proc. Conf. Learn. Theory, 2019, pp. 2714–2802.

[31] S. Fattahi, “Learning partially observed linear dynamical systems from logarithmic number of samples,” in Proc. Learn. Dyn. Control, 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 Proc. Int. Conf. Mach. Learn., 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 Proc. Int. Conf. Artif. Intell. Statist., 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 Syst. Lett., vol. 5, no. 3, pp. 989–994, Jul. 2021.

[35] H. Mohammad, M. Soltanolkotabi, and M. R. Jovanovic, “Random search for learning the linear quadratic regulator,” in Proc. IEEE Amer. Control Conf., 2020, pp. 4798–4803.

[36] A. Cassel, A. Cohen, and T. Koren, “Logarithmic regret for learning linear quadratic regulators efficiently,” in Proc. Int. Conf. Mach. Learn., 2020, pp. 1328–1337.

[37] M. Simchowitz and D. Foster, “Naive exploration is optimal for online LQR,” in Proc. Int. Conf. Mach. Learn., 2020, pp. 8937–8948.

[38] S. Lale, K. Azizzadenesheli, B. Hassibi, and A. Anandkumar, “Reinforcement learning with fast stabilization in linear dynamical systems,” in Proc. Int. Conf. Artif. Intell. Statist., 2022, pp. 5354–5360.

[39] E. Hazan, K. Singh, and C. Zhang, “Learning linear dynamical systems via spectral filtering,” in Proc. Adv. Neural Inf. Process. Syst., 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,” in Proc. Int. Conf. Learn. Representations, 2018. [Online]. Available: https://api.semanticscholar.org/CorpusID:64923464

[41] N. Agarwal, B. Bullins, E. Hazan, S. M. Kakade, and K. Singh, “Online control with adversarial disturbances,” in Proc. Int. Conf. Mach. Learn., 2019, pp. 154–165.

[42] N. Agarwal, E. Hazan, and K. Singh, “Logarithmic regret for oblivious control,” in Proc. Adv. Neural Inf. Process. Syst., vol. 32, 2019.

[43] M. Simchowitz, K. Singh, and E. Hazan, “Improper learning for non-stochastic control,” in Proc. Conf. Learn. Theory, 2020, pp. 3320–3436.

[44] C. Yu, G. Shi, S.-J. Chung, Y. Yue, and A. Wierman, “The power of predictions in online control,” in Proc. Adv. Neural Inf. Process. Syst., vol. 33, 2020.

[45] R. Zhang, Y. Li, and N. Li, “On the regret analysis of online LQR control with predictions,” in Proc. IEEE Amer. Control Conf., 2021, pp. 697–703.

[46] T.-J. Chang and S. Shahrampour, “Distributed online linear quadratic control for linear time-invariant systems,” in Proc. IEEE Amer. Control Conf., 2021, pp. 923–928.

[47] E. Hazan, S. Kakade, and K. Singh, “The nonstochastic control problem,” in Proc. Algorithmic Learn. Theory, 2020, pp. 408–421.

[48] S. Lale, K. Azizzadenesheli, B. Hassibi, and A. Anandkumar, “Logarithmic regret bound in partially observable linear dynamical systems,” in Proc. Adv. Neural Inf. Process. Syst., vol. 33, 2020.

[49] J. S. Liu, Monte Carlo Strategies in Scientific Computing. Berlin, Germany: Springer, 2008.

[50] G. Guo, Y. Zhao, and G. Yang, “Cooperation of multiple mobile sensors with minimum energy cost for mobility and communication,” Inf. Sci., 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 Trans. Knowl. Data Eng., vol. 25, no. 11, pp. 2483–2493, Nov. 2013.

[52] S. Shahrampour and A. Jadbabaie, “Distributed online optimization in dynamic environments using mirror descent,” IEEE Trans. Autom. Control, vol. 63, no. 3, pp. 714–725, Mar. 2018.