Asynchronous Heterogeneous Linear Quadratic Regulator Design

Leonardo F. Toso*, Han Wang*, and James Anderson

2024 IEEE 63rd Conference on Decision and Control (CDC) | 979-8-3503-1633-9/24/$31.00 ©2024 IEEE | DOI: 10.1109/CDC56724.2024.10885806

Abstract— **We address the problem of designing an LQR controller in a distributed setting, where M similar but not identical systems share their locally computed policy gradient (PG) estimates with a server that aggregates the estimates and computes a controller that, on average, performs well on all systems. Learning in a distributed setting has the potential to offer statistical benefits – multiple datasets can be leveraged simultaneously to produce more accurate policy gradient estimates. However, the interplay of heterogeneous trajectory data and varying levels of local computational power introduce bias to the aggregated PG descent direction, and prevents us from fully exploiting the parallelism in the distributed computation. The latter stems from synchronous aggregation, where straggler systems negatively impact the runtime. To address this, we propose an asynchronous policy gradient algorithm for LQR control design. By carefully controlling the "staleness" in the asynchronous aggregation, we show that the designed controller converges to each system's near-optimal controller up to a heterogeneity bias. Furthermore, we prove exact local convergence at a sub-linear rate.**

I. Introduction

Policy gradient (PG) methods stand as one of the fundamental pillars underpinning the success of model-free reinforcement learning (RL), offering a versatile framework for learning parameterized policies directly from experience [1], [2]. Within optimal control, in particular for the linear quadratic regulator (LQR) problem, PG approaches and their *non-asymptotic* performance guarantees have made the task of learning optimal controllers (in a model-free setting), both possible and systematic. In particular, the authors in [3] demonstrated that despite the problem's non-convexity, a derivative-free PG method (i.e., where policy gradients are estimated through simulation data) can converge to the global optimal LQR solution. The rate of convergence is further proved to be linear in [4].

Although policy gradient has shown to be effective in learning near-optimal LQR controllers, in a model-free centralized setting [3]–[7], it is often assumed that a single agent can access a sufficiently large simulation dataset to produce accurate policy gradient estimates. However, estimation variance may lead to sub-optimal solutions in a low data regime. A recent line of work focused on distributed learning for estimation [8]–[11] and control [12]–[16], considers a multi-agent setting, where small contributions (of for example estimated gradients [12], system models [8], task weights [10], among others) from each agent can be leveraged to achieve statistical benefits. In particular, [12] proposes a federated heterogeneous policy gradient approach to solve the model-free LQR problem. This work shows that by aggregating multiple systems' policy gradient estimates, the sample complexity enjoys a reduction proportional to the number of collaborating systems. However, aggregating heterogeneous PG estimates will inevitably introduce bias to the policy gradient descent direction, as discussed in [12].

Critically, the work above implicitly assumes that the policy gradient estimates from all participating agents are promptly available at the beginning of each aggregation step. However, network communication-induced constraints (e.g., rate-limited channels [17]) and agent drop-outs [18] may lead to the presence of straggler agents whose late arrival at the server could drastically affect the parallelism of the distributed computation. Namely, an agent with high computational power and/or a fast and reliable communication channel will have its performance throttled as it waits for the idle server (which is waiting for the slower agents) to broadcast the new updated controller.

To circumvent this limitation, we propose an asynchronous policy gradient approach, where only a batch of the fastest reported estimates at each iteration step are aggregated. This simple modification in the aggregation scheme will mitigate the presence of straggler systems, i.e., by adapting the batch size, fast agents will not be bothered by long delays incurred by waiting for the slow ones to finish their estimates. The trade-off is that the policy gradients produced by the slower agents will be used in the next round of aggregation, they are now out of date, or "stale". This staleness may negatively impact the convergence rate of the proposed approach.

Motivated by this, we aim to investigate how the convergence of model-free distributed LQR design is affected by the interplay between stale and heterogeneous PG estimates. In particular, we aim to answer the following questions: ***Can an asynchronous algorithm produce a controller that is near-optimal – even in the presence of staleness and heterogeneous system dynamics?*** If so, how does the staleness affect the policy gradient convergence? Can we still expect a linear convergence to hold in this setting?

A. Contributions

Our main contributions are summarized as follows:

- This is the first work to investigate how aggregating stale PG estimates affects the convergence of an asynchronous model-free distributed LQR design (Algorithm 1). We highlight that in our setting, we are dealing with multiple, *different* systems. We show that the

* Leonardo F. Toso and Han Wang contributed equally to this work. This work is supported in part by NSF awards 2144634 & 2231350. The authors are with the Department of Electrical Engineering at Columbia University, New York, NY, 10027, USA. Email: {lt2879, hw2786, james.anderson}@columbia.edu.

staleness effect can be mitigated by carefully controlling the step size in the PG updates. Moreover, in contrast to a synchronous aggregation scheme, as proposed in [12], our approach fully exploits the parallelism in the distributed computation. In particular, it alleviates straggler agents' impact by selecting only a batch of the fastest reported PG estimates at each iteration.

- We establish the global convergence guarantees of Algorithm 1. To achieve this, we derive an upper bound for the staleness term (Lemma 4), which is approximately of the order of the magnitude of the current iteration step's PG. We prove that our asynchronous algorithm produces a controller that is ϵ-near optimal up to a heterogeneity bias (Corollary 2). This bias arises due to the heterogeneity among the M systems. We demonstrate that staleness impedes global convergence; in particular, the total number of iterations N to achieve each system's ϵ-near optimal controller will be amplified by $\tau_{\max}^{3/2}$ (Corollary 2), where $\tau_{\max}$ is the maximum staleness across systems and PG steps.
- We also provide local convergence guarantees for Algorithm 1. Compared to the global convergence analysis, the heterogeneity bias and the staleness effect disappear when converging to a local stationary solution, i.e., our algorithm can exactly locally converge even under the asynchronous aggregation and heterogeneous setting. However, it comes at the cost of a slower convergence, i.e., Algorithm 1 sub-linearly converges to such fixed point (Corollary 1). Notably, our tighter local convergence bound demonstrates a linear *speedup* w.r.t. the number of aggregated PG estimates. This improves upon previous work [19], [20].

B. Related Work

Model-free LQR Design: The setting where a single agent uses its data to estimate policy gradients and perform controller updates to solve the model-free LQR problem has been widely studied [3]–[7]. Although the results on the global and linear convergence are positive and demonstrate the effectiveness of the method, Ziemann et al. [21] show that PG is very much affected by the limits of control, i.e., poor controlability leads to arbitrarily noisy gradient estimates. On the other hand, [12], [13], [15] have demonstrated the value of collaboration, in a multi-agent setting, to reduce the variance in the estimated gradient and achieve sample efficiency when learning LQR controllers. Most relevant to our work is [12], where the authors consider a synchronous policy gradient approach to tackle the model-free LQR problem. In contrast to [12], our work considers an asynchronous aggregation scheme where the impact of straggler agents is mitigated and the effect of aggregating stale PG estimates is thoroughly characterized in our local and global convergence analysis.

Asynchronous Optimization: Asynchronous stochastic gradient descent (SGD) has been a topic of study in stochastic optimization over the past decade, where many papers [22]–[24] investigate the connection of large batches and staleness in the ergodic convergence rate of such approach. Most relevant to our work are the recent papers on asynchronous distributed learning [19], [20], [25]–[27], where in contrast to asynchronous SGD, the heterogeneity in the local data distribution and local data privacy concerns are taken into account. In contrast to this line of work, we consider an asynchronous policy gradient approach to solve the LQR optimal control problem. Here, not only are the convergence guarantees characterized, but so is the per-iteration closed-loop stability of the collaborating agents. Our theoretical guarantees reveals a linear speedup with respect to the number of aggregated policy gradient estimates in the local convergence rate. We believe that these results can be used to achieve tighter bounds in the more general work of [19], [20] on asynchronous federated learning.

C. Notation

Let $[M]$ denote the set of integers $\{1, \ldots, M\}$. We use $\mathcal{J}(K)$ to denote the LQR cost for an arbitrary system with problem parameters (A, B, Q, R). When required, $\mathcal{J}^{(i)}(K)$ denotes the LQR cost for a specific tuple $(A^{(i)}, B^{(i)}, Q^{(i)}, R^{(i)})$, where $i \in [M]$. The spectral radius of a square matrix is $\rho(\cdot)$, and $\sigma_{\min}(\cdot)$ denotes the minimum singular value. Unless otherwise stated, $\|\cdot\|$ is the spectral norm. We use $\mathcal{O}(\cdot)$ to omit constant factors in the argument.

II. Problem Formulation

Let us begin with the standard setup of multi-agent LQR design for heterogeneous systems [12], [13]. Consider M discrete-time and linear time-invariant (LTI) dynamical systems over an infinite time-horizon, described by

$$x_{t+1}^{(i)} = A^{(i)} x_t^{(i)} + B^{(i)} u_t^{(i)}, \quad \forall i \in [M], \tag{1}$$

where $A^{(i)} \in \mathbb{R}^{n_x \times n_x}$, $B^{(i)} \in \mathbb{R}^{n_x \times n_u}$ are the system matrices, and $x_t^{(i)} \in \mathbb{R}^{n_x}$, $u_t^{(i)} \in \mathbb{R}^{n_u}$ denote the state and control input of system i at time instant t, respectively. The initial state $x_0^{(i)}$ of (1) is drawn from an arbitrary distribution $\mathcal{X}_0$ that satisfies Assumption 1 below. We specifically account for the fact that the M systems are *not identical*, i.e., in general $A^{(i)} \neq A^{(j)}$ and $B^{(i)} \neq B^{(j)}$. We quantify the level of heterogeneity at the end of this section.

From the perspective of the i^{th} system, the goal is to design an optimal static state feedback controller $K_i^\star \in \mathcal{K}^{(i)} := \{K \in \mathbb{R}^{n_u \times n_x} \mid \rho(A^{(i)} - B^{(i)}K) < 1\}$, that provides a control policy $u_t^{(i)} = -K_i^\star x_t^{(i)}$ that, subject to (1), minimizes the quadratic cost function

$$\mathcal{J}^{(i)}(K) := \mathbb{E}\left[\sum_{t=0}^{\infty} x_t^{(i)\top}\left(Q^{(i)} + K^\top R^{(i)} K\right) x_t^{(i)}\right], \tag{2}$$

where $Q^{(i)} \in \mathbb{R}^{n_x \times n_x}$ and $R^{(i)} \in \mathbb{R}^{n_u \times n_u}$ denote the (positive semidefinite and definite, respectively) cost matrices, and the expectation is with respect to $x_0^{(i)} \sim \mathcal{X}_0$.

Assumption 1 (Initial state distribution): The initial state distribution $\mathcal{X}_0$ satisfies $\mathbb{E}[x_0^{(i)}] = 0$ (i.e., zero mean) with covariance $\Sigma_0 = \mathbb{E}[x_0^{(i)} x_0^{(i)\top}] \succ \mu I_{n_x}$ for some $\mu > 0$.[1]

[1] Assumption 1 is a standard assumption in PG-LQR literature [3], [5], [6] and guarantees that all stationary solutions are global optima.

This work considers the model-free setting where the tuple $(A^{(i)}, B^{(i)}, Q^{(i)}, R^{(i)})$ is *unknown* and each system only has limited access to simulation data. Thus, designing $K_i^\star$ through the well-established Riccati equation [28] is not possible. As in [3], [5], we must resort to derivative-free policy gradient approaches to optimize (2).

However, due to limited local trajectory data, accurate policy gradient estimations in the single-agent setting may require more data than is available. As proposed in [12], instead of designing $K_i^\star$ for each system $i \in [M]$, we consider the problem of leveraging simulation data from multiple (differing but "similar") systems in order to compute a controller $\bar{K}^\star \in \mathcal{K} := \cap \mathcal{K}^{(i)}$ that, i) stabilizes each system, and ii) on average, performs well for all of them. Moreover, such an approach should be sample efficient, in the sense that a small data contribution from multiple systems can be leveraged into a larger dataset to perform more accurate PG estimates. With regards to ii), $\bar{K}^\star$ minimizes

$$\bar{K}^\star := \underset{K \in \mathcal{K}}{\operatorname{argmin}} \left\{ \bar{\mathcal{J}}(K) := \frac{1}{M} \sum_{i=1}^{M} \mathcal{J}^{(i)}(K) \right\}, \tag{3}$$

subject to (1). To solve (3), we first consider an arbitrary initial stabilizing controller $\bar{K} \in \mathcal{K}$ and step-size $\eta \in \mathbb{R}_{>0}$, and iteratively perform policy gradient updates of the form:

$$\bar{K} \leftarrow \bar{K} - \eta \widehat{\nabla} \bar{\mathcal{J}}(\bar{K}),$$

where $\widehat{\nabla} \bar{\mathcal{J}}(\cdot)$ is an estimate of the true gradient $\nabla \bar{\mathcal{J}}(\cdot)$.

In [12], at each iteration $n \in \{0, 1, 2, \ldots\}$, all M policy gradient estimates generated by the systems, $\widehat{\nabla} \mathcal{J}^{(i)}(\bar{K}_n)$, are aggregated to produce $\widehat{\nabla} \bar{\mathcal{J}}(\bar{K}_n)$, where $\bar{K}_n$ denotes the controller at iteration n. This is then used to compute the new policy $\bar{K}_{n+1}$. It is implicitly assumed that the policy gradients are all available to produce $\widehat{\nabla} \bar{\mathcal{J}}(\bar{K}_n)$. Such an assumption (i.e., synchronous aggregation) does not take into account network communication effects. While synchronously aggregating such estimates may offer sample efficiency [12]; however, the presence of straggler systems will prevent full exploitation of the parallelism in the distributed computation. To alleviate this limitation, we consider an asynchronous distributed LQR design, where, at each iteration step n, only a subset $[b_s] \subseteq [M]$ of the first reported PG estimates contribute to the update of $\bar{K}_{n+1}$, i.e.,

$$\bar{K}_{n+1} = \bar{K}_n - \frac{\eta}{b_s} \sum_{s=1}^{b_s} \widehat{\nabla} \mathcal{J}^{(s)}(\bar{K}_{n-\tau_s(n)}), \tag{4}$$

where $\tau_s(n) \in \mathbb{N}$ denotes the staleness in the controller that system $s \in [b_s]$ possesses when locally estimating its policy gradient at step n. Due to heterogeneity, in the synchronous setting [12], $\limsup_{n \to \infty} \bar{K}_{n+1} \neq K_i^\star$, and $\mathcal{J}^{(i)}(\bar{K}_n) - \mathcal{J}^{(i)}(K_i^\star)$ is upper bounded by the dissimilarity across systems.

Intuitively, besides heterogeneity, aggregating over stale estimates may also produce sub-optimal solutions to the LQR design. Therefore, in this work, we analyze the interplay between heterogeneity and the staleness in the convergence of the policy update (4). Figure 1 illustrates the comparison between the synchronous and asynchronous policy gradient aggregation for the model-free LQR problem. As we can see on the left-hand side, after the server communicates the updated controller (step 1), it needs to wait (step 4), for all systems to complete their estimates (step 2) and communicate back to the server (step 3), before performing the next controller update (step 5). On the other hand, in the asynchronous setup, the server simply broadcasts the updated controller as soon as it has b_s updates stored. This batch of updates will likely include updates that were received too late to be included in the previous broadcast. Obviously as the batch size is decreased, the staleness increases.

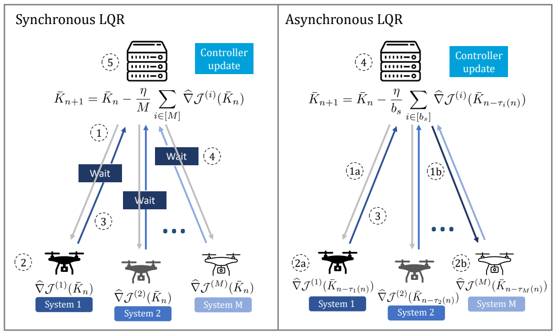

Synchronous LQR
Asynchronous LQR
Controller update
Controller update
$\bar{K}_{n+1} = \bar{K}_n - \frac{\eta}{M} \sum_{i \in [M]} \widehat{\nabla} \mathcal{J}^{(i)}(\bar{K}_n)$
$\bar{K}_{n+1} = \bar{K}_n - \frac{\eta}{b_s} \sum_{i \in [b_s]} \widehat{\nabla} \mathcal{J}^{(i)}(\bar{K}_{n-\tau_i(n)})$
5
1
2
3
4
Wait
Wait
Wait
$\widehat{\nabla} \mathcal{J}^{(1)}(\bar{K}_n)$
$\widehat{\nabla} \mathcal{J}^{(2)}(\bar{K}_n)$
$\widehat{\nabla} \mathcal{J}^{(M)}(\bar{K}_n)$
System 1
System 2
System M
4
1a
1b
3
2a
2b
$\widehat{\nabla} \mathcal{J}^{(1)}(\bar{K}_{n-\tau_1(n)})$
$\widehat{\nabla} \mathcal{J}^{(2)}(\bar{K}_{n-\tau_2(n)})$
$\widehat{\nabla} \mathcal{J}^{(M)}(\bar{K}_{n-\tau_M(n)})$
System 1
System 2
System M

Fig. 1. Illustrative schematic to compare the synchronous and asynchronous PG approaches for the model-free LQR problem. Figure inspired by [26].

Despite of the staleness in the updated controller (step 1a or 1b) that a system may have when estimating the PG (step 2a or 2b), only the first b_s PG estimates reported back to the server (step 3) are aggregated (step 4). The impact of straggler systems is then alleviated by adapting b_s.

Before presenting our algorithm that implements asynchronous policy gradient updates, we first need to define the sub-level set of stabilizing controllers $\mathcal{S} \subseteq \mathcal{K}$, an introduce an assumption.

Definition 1: Let $\bar{K}_0$ and $K_i^\star$ be the initial and optimal stabilizing controllers of system $i \in [M]$, respectively. The stabilizing sub-level set of $\mathcal{K}$ is $\mathcal{S} \triangleq \cap \mathcal{S}^{(i)}$, with

$$\mathcal{S}^{(i)} := \left\{ K \mid \mathcal{J}^{(i)}(K) - \mathcal{J}^{(i)}(K_i^\star) \leq \gamma \Delta_0^{(i)} \right\},$$

where $\Delta_0^{(i)} \triangleq \mathcal{J}^{(i)}(\bar{K}_0) - \mathcal{J}^{(i)}(K_i^\star)$ denotes the initial distance to optimality, and $\gamma \geq 1$ is an arbitrary scalar.

Assumption 2 (Initial stabilizing controller $\bar{K}_0$): The initial controller $\bar{K}_0$ stabilizes all systems, i.e. $\bar{K}_0 \in \mathcal{S}$.

The assumption on the initial stabilizing controller is standard in policy gradient methods for LQR design [3], [4], [6]. If $\bar{K}_0$ is not stabilizing for all the systems (1), then (4) will not produce a stabilizing controller, since $\mathcal{J}^{(i)}(\bar{K}_0)$ is undefined for the corresponding unstabilized systems. In addition, we emphasize that although $\bar{K}_0$ stabilizes (1) $\forall i$, it may provide a sub-optimal cost $\bar{\mathcal{J}}(\bar{K}_0) \geq \bar{\mathcal{J}}(\bar{K}^\star)$.

III. Asynchronous Policy Gradient Algorithm

Our asynchronous policy gradient algorithm for heterogeneous model-free LQR control is described in Algorithm

1 below. It implements the controller update in (4). The subset $[b_s] \subseteq [M]$ of policy gradient estimates are computed through a two-point zeroth-order gradient estimation approach, as detailed in Algorithm 2. Upon initializing all systems $i \in [M]$ with an initial stabilizing controller $\bar{K}_0$, (step 2 of Algorithm 1), in parallel, each system estimates its own policy gradient using its local simulation data in step 3. The zeroth-order estimation in Algorithm 2 performs an empirical estimation of the first-order Gaussian Stein's identity [29]. That is, given a smoothing radius r, the current controller K is perturbed by a random matrix U, such that $\|U\|_F = r$, to produce $K^1 = K + U$ and $K^2 = K - U$. Such smoothing controllers are then played by the i^{th} system to collect simulation data and compute the costs $\mathcal{J}^{(i)}(K^1)$, and $\mathcal{J}^{(i)}(K^2)$. Then, by averaging over m samples, Algorithm 2 returns a biased empirical estimation $\widehat{\nabla}\mathcal{J}(K)$ of $\mathbb{E}\left[\nabla\mathcal{J}^{(i)}(K)\right] = \mathbb{E}\left[\frac{n_x n_u}{2r^2}(\mathcal{J}^{(i)}(K^1) - \mathcal{J}^{(i)}(K^2))U\right]$.

Algorithm 1 Asynchronous LQR

```
1:  Input: Stabilizing controller $\bar{K}_0$, step-size $\eta$, batch size $b_s$, iterations $N$, smoothing radius $r$, and samples $m$.
2:  Initialize the local controllers $K_i = \bar{K}_0 \; \forall i \in [M]$, batch and iteration counters $s = n = 0$, and $\overline{\nabla} \leftarrow 0$.
3:  In parallel compute and send $\widehat{\nabla}_i = \texttt{ZO}(K_i, r, m)$ to the server $\forall i \in [M]$.
4:  While $n < N$
5:    If the server receives an estimate then
6:      Accumulate $\overline{\nabla} = \overline{\nabla} + \widehat{\nabla}_i$, $s \leftarrow s + 1$,
7:    If $s = b_s$ then
8:      $\bar{K}_{n+1} = \bar{K}_n - \frac{\eta}{b_s}\overline{\nabla}$,  $n \leftarrow n + 1$, $\overline{\nabla} \leftarrow 0$, $s \leftarrow 0$,
9:      If system $i \in [M]$ is done, then $K_i \leftarrow \bar{K}_{n+1}$ and
10:       Compute $\widehat{\nabla}_i = \texttt{ZO}(K_i, r, m)$,
11:       Send $\widehat{\nabla}_i$ to the server,
12: Output: $\bar{K}_N$.
```

Once a system $i \in [M]$ is done estimating its PG, it sends the estimate to a server that accumulates them (step 6). A policy gradient update (4) is only performed when the number of accumulated gradient estimates is equivalent to the batch size b_s (step 9), indicating that the b_s fastest reported PG estimates at iteration n are ready to be aggregated. After N iterations of steps 4-11, Algorithm 1 returns $\bar{K}_N$. In Section IV, we characterize the properties of $\bar{K}_N$ based on local and global convergence rates. We re-emphasize that Algorithm 1 aggregates stale policy gradient estimates in step 8. In addition, due to the heterogeneity, such staleness in the controller that each system access to perform its gradient estimate is later assumed to be bounded[2] in Assumption 3.

Remark 1: We exploit a two-point gradient estimation approach since it offers a lower estimation variance compared to the one-point counterpart [5]. Moreover, for simplicity and by following [5], [7], in Algorithm 2, it is implicitly assumed to have access to the true infinite horizon costs $\mathcal{J}^{(i)}(K^1)$ and $\mathcal{J}^{(i)}(K^2)$. However, since such quantities are lower bounded by any finite-horizon approximation, our results can be readily extended to that setting as well [6].

Algorithm 2 $\texttt{ZO}$: Two-point Zeroth-order Estimation

```
1: Input: Stabilizing controller $K$, number of samples $m$ and smoothing radius $r$.
2: for all samples $l \in [m]$ do
3:   Draw $U_l \in \mathbb{R}^{n_u \times n_x}$, such that $\|U_l\|_F = r$,
4:   Smooth controllers: $K_l^1 = K + U_l$ and $K_l^2 = K - U_l$,
5:   Compute and store costs $\mathcal{J}(K_l^1)$, and $\mathcal{J}(K_l^2)$,
6: end for
7: Return $\widehat{\nabla}\mathcal{J}(K) = \frac{n_x n_u}{2r^2 m}\sum_{l=1}^{m}(\mathcal{J}(K_l^1) - \mathcal{J}(K_l^2))U_l$
```

Assumption 3 (Bounded staleness): For any system $i \in [M]$ and iteration n, $\tau_i(n) \leq \tau_{\max}$, for some $\tau_{\max} \in [1, \infty)$.

The above assumption is common in the convergence analysis of asynchronous stochastic gradient descent algorithms [19], [20], [31]. It guarantees that the asynchronous aggregation in Algorithm 1 is performed within a finite time.

Before jumping to the convergence and stability analysis of Algorithm 1, we first revisit some properties of the policy gradient LQR [3] and heterogeneity bound [12] that are instrumental in deriving the main results of this work.

Lemma 1 (Local smoothness): Given a pair of stabilizing controllers $K, K' \in \mathcal{S}$ such that $\|K' - K\|_F < \infty$, the LQR gradient is h_{grad}-Lipschitz, i.e.,

$$\left\|\nabla\mathcal{J}^{(i)}(K') - \nabla\mathcal{J}^{(i)}(K)\right\|_F \leq h_{\text{grad}}\|K' - K\|_F,$$

where h_{grad} depends on the LQR problem parameters.

The proof of the LQR gradient's local smoothness was first introduced in [3], and the explicit expression of h_{grad} was further provided in [12, Appendix D.1].

Lemma 2 (Gradient dominance): Let $K_i^\star$ be the LQR optimal controller associated with system $i \in [M]$. Given a stabilizing controller $K \in \mathcal{S}$ the squared norm of the LQR gradient is lower bounded as follows:

$$\|\nabla\mathcal{J}^{(i)}(K)\|_F^2 \geq \lambda\left(\mathcal{J}^{(i)}(K) - \mathcal{J}^{(i)}(K_i^\star)\right), \quad \forall i \in [M],$$

where $\lambda = 4\mu^2 \max_{i\in[M]} \sigma_{\min}(R^{(i)})/\|\Sigma_{K_i^\star}\|$ denotes the gradient dominance constant and $\Sigma_{K_i^\star} = \mathbb{E}[x_t^{(i)} x_t^{(i)\top}]$ corresponds to the closed-loop state covariance matrix incurred by playing (1) with its corresponding optimal controller.

Lemmas 1-2 are paramount to prove the global convergence of the policy gradient LQR in the single-agent setting [3]. Even though, the gradient dominance property of each $\mathcal{J}^{(i)}(K)$ does not imply the same property to the the average cost $\bar{\mathcal{J}}(K)$, i.e., due system and cost heterogeneity. We can still leverage such result when characterizing the distance to optimality, i.e., $\Delta_N^{(i)} := \mathcal{J}^{(i)}(\bar{K}_N) - \mathcal{J}^{(i)}(K_i^\star)$, for all systems $i \in [M]$, in Section IV.

We now quantify the level of heterogeneity between the M systems. Note that we could include a common bound on the spectral norm difference among system and cost matrices at the expense of less precise downstream results.

[2] This can be relaxed to unbounded staleness in the homogeneous asynchronous distributed learning setting, as discussed in [30].

Assumption 4: There exist positive scalars $\epsilon_A, \epsilon_B, \epsilon_Q, \epsilon_R$, such that system and cost heterogeneity is bounded. That is,

$$\max_{i \neq j} \|A^{(i)} - A^{(j)}\| \leq \epsilon_A, \quad \max_{i \neq j} \|B^{(i)} - B^{(j)}\| \leq \epsilon_B,$$
$$\max_{i \neq j} \|Q^{(i)} - Q^{(j)}\| \leq \epsilon_Q, \quad \max_{i \neq j} \|R^{(i)} - R^{(j)}\| \leq \epsilon_R.$$

Lemma 3 (Gradient heterogeneity [13]): Given a pair of distinct systems $i \neq j \in [M]$, following the dynamics in (1), and a stabilizing controller $K \in \mathcal{S}$. The following holds,

$$\|\nabla \mathcal{J}^{(i)}(K) - \nabla \mathcal{J}^{(j)}(K)\|_F^2 \leq \epsilon_{\text{het}}.$$

where ϵ_{het} scales quadratically with the system and cost heterogeneity levels $(\epsilon_A, \epsilon_B, \epsilon_Q, \epsilon_R)$ of Assumption 4.

The proof of the above lemma as well as the explicit expression of ϵ_{het} is presented in [13, Appendix 6.2]. Note that, for a small gradient heterogeneity level ϵ_{het}, this lemma conveys that the PG descent directions of systems $i \neq j \in [M]$ are close, which is then also close to the descent direction of the average cost. This lemma is crucial to quantify the effect of heterogeneity in the convergence and stability analysis of our asynchronous policy gradient aggregation.

IV. Convergence and Stability Analysis

We now present the main theoretical results of this work. First, we show that Algorithm 1 can exactly converge to a local optimum of (3) at a sub-linear convergence rate. Second, we provide global convergence guarantees for our proposed approach. In the global convergence analysis, due to heterogeneity, our algorithm will converge to a ball that contains each system's optimal controller. The size of the convergence ball depends on the heterogeneity level across systems. We demonstrate that, even in the presence of a staleness, this convergence is achieved at a linear rate with respect to the tolerance level. Moreover, we establish a linear convergence rate that has a dependence on the maximum staleness $\tau_{\max}$. We refer to our extended version [32] for the proofs of our main results presented below.

A. Local convergence guarantees

The local convergence of Algorithm 1 is characterized through the ergotic convergence rate, i.e., how $\frac{1}{N}\sum_{n=0}^{N-1} \mathbb{E}\|\nabla \bar{\mathcal{J}}(\bar{K}_n)\|_F^2$ scales with the number of iterations N, batch size b_s, heterogeneity ϵ_{het} and staleness $\tau_{\max}$.

Theorem 1: Let Assumptions 1-4 hold. Suppose the step-size satisfies $\eta \leq h_{\text{grad}}\eta_{\text{ergodic}}$. Then, it holds that

$$\frac{1}{N}\sum_{n=0}^{N-1} \mathbb{E}\|\nabla \bar{\mathcal{J}}(\bar{K}_n)\|_F^2 \leq \frac{2\bar{\Delta}_0}{\eta N} + \frac{c_{\text{dim}}\epsilon_{\text{het}}\left(\eta + \eta^2\tau_{\max}\right)}{b_s} + c_{\text{bias}},$$

where $\bar{\Delta}_0 = \mathbb{E}\left[\bar{\mathcal{J}}(\bar{K}_0) - \bar{\mathcal{J}}(\bar{K}^\star)\right]$, for some positive constants $c_{\text{dim}} = \mathcal{O}(n_x^2)$ and $c_{\text{bias}} = \mathcal{O}(r^2)$, and

$$\eta_{\text{ergodic}} = \min\left\{\frac{1}{8}, \frac{\sqrt{b_s}}{\sqrt{6}\tau_{\max}}, \frac{1}{\max\{\sqrt{32c_{\text{step}}}\tau_{\max}, 2c_{\text{step}}\}}\right\},$$

with $c_{\text{step}} = \mathcal{O}\left(n_x^2 + \frac{n_x^2}{b_s}\right)$.

With this theorem, we are now ready to state the convergence of our Algorithm 1 to a first-order stationary point.

Corollary 1: Let the arguments of Theorem 1 hold. Suppose that the step-size is set such that $\eta = \mathcal{O}\left(\sqrt{\frac{b_s}{N}}\right)$. Then, $\bar{K}_N \in \mathcal{S}$ satisfy the following ergodic convergence rate[3]:

$$\frac{1}{N}\sum_{n=0}^{N-1} \mathbb{E}\|\nabla \bar{\mathcal{J}}(\bar{K}_n)\|_F^2 \leq \mathcal{O}\left(\frac{\bar{\Delta}_0}{\sqrt{Nb_s}} + \frac{\epsilon_{\text{het}}}{\sqrt{Nb_s}} + \frac{\tau_{\max}^2\epsilon_{\text{het}}}{N}\right). \tag{5}$$

Corollary 1 presents the local convergence guarantee for our proposed approach with respect to the total number of iterations N and batch size b_s. The main message of (5) is that Algorithm 1 achieves a local convergence rate of $\mathcal{O}(\frac{1}{\sqrt{Nb_s}}) + \mathcal{O}(\frac{\tau_{\max}^2}{N})$. In particular, the second term $\mathcal{O}(\frac{\tau_{\max}^2}{N})$ reveals the effect of the staleness, which becomes negligible when $N \geq b_s$. In the first dominant term $\mathcal{O}(\frac{1}{\sqrt{Nb_s}})$, we demonstrate that our algorithm enjoys a linear speedup with respect to the batch size b_s. This result improves upon previous work in the asynchronous distributed learning setting [19], [20], [31], where no speedup is established.

B. Global convergence guarantees

We characterize the global convergence of the proposed asynchronous LQR design, by analyzing the interplay between staleness $\tau_{\max}$ and heterogeneity ϵ_{het} in the optimality gap $\Delta_N^{(i)}$, i.e., the cost difference between the designed controller $\bar{K}_N$ and each system's optimal controller $K_i^\star$. To this end, let us first provide an upper bound on the staleness effect throughout the iterations of Algorithm 1.

Lemma 4: Suppose that the step-size is set according to $\eta = \mathcal{O}\left(\tau_{\max}^{-\frac{1}{2}}\right)$. Then, it holds that[3]

$$\mathbb{E}\|\bar{K}_{l+1} - \bar{K}_l\|_F^2 \leq \eta^2\tau_{\max}\mathcal{O}\left(\epsilon_{\text{het}} + \mathbb{E}\|\nabla \mathcal{J}^{(i)}(\bar{K}_n)\|_F^2\right),$$

$\forall l \in [n - \tau_{\max}, n-1]$ and $n \in [N-1]$.

The proof for the above lemma is detailed in Appendix B of our extended version [32]. Further in this section, we present the proof sketch of our theoretical convergence guarantees, where the induction reasoning that leads to Lemma 4 is discussed. By Lemma 4, we can conclude that the staleness effect:

$$\mathbb{E}\|\bar{K}_n - \bar{K}_{n-\tau_i(n)}\|_F^2 \leq \eta^2\tau_{\max}^3\mathcal{O}\left(\epsilon_{\text{het}} + \mathbb{E}\|\nabla \mathcal{J}^{(i)}(\bar{K}_n)\|_F^2\right),$$

can be approximately upper bounded by the product of the norm-squared policy gradient at the n-th iteration and the step-size squared η^2. By choosing a sufficiently small step-size η, the impact of the staleness can become negligible since it is in the high-order terms with respect to η. However, the adoption of small step-sizes η to overcome the stragglers will slow down the convergence. We rigorously characterize such trade-off between $\tau_{\max}$ and η in the following theorem.

Theorem 2 (Optimality gap): Let Assumptions 1-4 hold. Suppose that the step-size is such that $\eta \leq \eta_{\text{gap}}$. Then, the optimality gap $\Delta_N^{(i)} = \mathbb{E}\left[\mathcal{J}^{(i)}(\bar{K}_N) - \mathcal{J}^{(i)}(K_i^\star)\right]$ satisfy:

$$\Delta_N^{(i)} \leq c_{\text{cont}}^N\Delta_0^{(i)} + \lambda^{-1}(c_{\text{dim}}\epsilon_{\text{het}} + c_{\text{bias}}),$$

[3] We omit $\mathcal{O}(c_{\text{bias}})$ in that bound, since r can be set sufficiently small such that its contribution becomes negligible. See Appendix A of [32].

where $c_{\text{cont}} = 1-\frac{\eta\lambda}{4} \in (0,1)$ denotes the contraction rate and $\eta_{\text{gap}} = \min\left\{\eta_{\text{ergodic}}, \frac{1}{h_{\text{grad}}\max\left\{8\tau_{\max},\sqrt{2}\tau_{\max}^{3/2}\right\}}, \frac{1}{\tau_{\max}\sqrt{2c_{\text{bias}}}}\right\}$.

As stated in the above theorem, the optimality gap $\Delta_N^{(i)}$, is composed of a contraction term, with contraction rate c_{cont}, and an additive bias characterized by ϵ_{het} and c_{bias}. As the number of iterations N grows, the contraction shrinks to zero. In addition, as c_{bias} is in the order of r^2, the smoothing radius can be set sufficiently small such that its contribution becomes negligible in the bias term. However, ϵ_{het} is fixed and dominates the unavoidable bias in the optimality gap. In addition, the step-size η is in the order of $\mathcal{O}\left(\frac{1}{\tau_{\max}^{3/2}}\right)$. This condition demonstrates that as the staleness $\tau_{\max}$ increases, the step-size η needs to be reduced to preserve a global convergence guarantee of Algorithm 1. Next, we highlight the impact of $\tau_{\max}$ on the number of iterations N to achieve a controller that is ϵ-close to its optimal controller.

Corollary 2 (Linear convergence): Let the arguments of Theorem 2 hold. Suppose that the number of iterations N of Algorithm 1 and the smoothing radius r of Algorithm 2 satisfy: $N \geq \mathcal{O}\left(\tau_{\max}^{3/2}\log\left(\frac{\Delta_0^{\vee}}{\epsilon}\right)\right), r \leq \mathcal{O}(\epsilon)$, for a small tolerance $\epsilon \in (0,1)$, where $\Delta_0^{\vee} := \max_{i\in[M]}\Delta_0^{(i)}$. Then, the optimality gap satisfies: $\Delta_N^{(i)} \leq \mathcal{O}\left(\epsilon + \epsilon_{\text{het}}\right)$.

Corollary 2 shows that by carefully controlling the step-size η, number of iterations N and smoothing radius r, the designed controller $\bar{K}_N$ is ϵ-close to each system's optimal controllers up to a heterogeneity bias. Note that the number of iterations N will increase with the maximum staleness – which is of order $\tau_{max}^{3/2}$. In other words, the staleness will slow down the global convergence rate.

C. Proof sketch

We now discuss the main steps and reasoning to obtain the theoretical convergence results presented in this work. First, for the local convergence rate, we note that as long as Lemma 1 holds for any system $i \in [M]$, it implies that the gradient of the average LQR cost is also h_{grad}-Lipschitz. Therefore, the gap in the average cost between two consecutive iterations of Algorithm 1, i.e., $\bar{\Delta}_n = \bar{\mathcal{J}}(\bar{K}_{n+1}) - \bar{\mathcal{J}}(\bar{K}_n)$, is approximately upper bounded as follows

$$\begin{aligned}\mathbb{E}\left[\bar{\Delta}_n\right] &\lesssim -\eta\mathbb{E}\|\nabla\bar{\mathcal{J}}(\bar{K}_n)\|_F^2 - \eta\mathbb{E}\left\|\nabla\bar{\mathcal{J}}_r(\bar{K}_{n-\tau_i(n)})\right\|_F^2\\ &+\frac{\eta}{M}\sum_{i=1}^{M}\underbrace{\mathbb{E}\left\|\bar{K}_n-\bar{K}_{n-\tau_i(n)}\right\|_F^2}_{\text{staleness term}}+\eta(r^2+\eta\epsilon_{\text{het}}),\end{aligned}$$

where we use $\lesssim$ to omit constant factors in the expression, and $\nabla\bar{\mathcal{J}}_r(\bar{K}) := \mathbb{E}\widehat{\nabla}\bar{\mathcal{J}}(\bar{K})$. In addition, the staleness term can be upper bounded as follows

$$\begin{aligned}\mathbb{E}\left\|\bar{K}_n-\bar{K}_{n-\tau_i(n)}\right\|_F^2 &= \mathbb{E}\left\|\sum_{l=n-\tau_i(n)}^{n-1}\bar{K}_{l+1}-\bar{K}_l\right\|_F^2\\ &\leq \tau_{\max}\sum_{l=n-\tau_i(n)}^{n-1}\mathbb{E}\left\|\bar{K}_{l+1}-\bar{K}_l\right\|_F^2,\end{aligned} \tag{6}$$

then, by summing the above expression over the iterations n, the staleness effect is shown to be in the following order.

$$\begin{aligned}\sum_{n=0}^{N-1}\mathbb{E}\left\|\bar{K}_n-\bar{K}_{n-\tau_i(n)}\right\|_F^2 &\lesssim \tau_{\max}^2N\eta^3r^2+\frac{\tau_{\max}^2N\eta^3\epsilon_{\text{het}}}{b_s}\\ &+\tau_{\max}^2\eta^3\sum_{n=0}^{N-1}\mathbb{E}\left\|\nabla\bar{\mathcal{J}}_r(\bar{K}_{n-\tau_i(n)})\right\|_F^2,\end{aligned}$$

which can be used in the expression of the expected average gap, i.e., $\mathbb{E}\left[\bar{\Delta}_n\right]$, and with a proper selection of the step-size η we can obtain the result presented in Theorem 1. We emphasize that since the interplay between staleness $\tau_{\max}^3$ and heterogeneity ϵ_{het} is accompanied by η^3, its contribution in the local convergence rate should only appears in a high-order term when we set $\eta = \mathcal{O}\left(\sqrt{\frac{b_s}{N}}\right)$.

On the other hand, what changes in the global convergence guarantees is how the staleness effect is upper bounded. To this end, we use the result in [32, Lemma 10]. The proof of this lemma relies on an induction approach, where for any two consecutive iterations of Algorithm 1, we have

$$\mathbb{E}\|\bar{K}_{n+1}-\bar{K}_n\|_F^2 \lesssim \tau_{\max}\eta^2\epsilon_{\text{het}}+\tau_{\max}\eta^2\mathbb{E}\|\nabla\mathcal{J}^{(i)}(\bar{K}_n)\|_F^2, \tag{7}$$

then, since the staleness term (6) is evaluated within the interval $l \in [n-\tau_{\max}, n-1]$, the bound in (7) combined with an induction step implies in Lemma 4. Therefore, by using lemma 4 with the local smoothness and gradient dominance properties (i.e., Lemmas 1-2), we obtain the global convergence results of Theorem 2 and Corollary 2.

D. Stability guarantees

An important requirement that policy gradient methods need to satisfy within control tasks, is the ability of iteratively preserving the closed-loop stability of the collaborating systems with respect to the designed controller, i.e., $\bar{K}_n$ should stabilize (1) for every iteration n. Note that, one of the conditions to ensure such requirement, is to have access to an initial stabilizing controller $\bar{K}_0 \in \mathcal{S}$. However, it is also necessary to impose conditions on the step-size η, smoothing radius r, and heterogeneity ϵ_{het} to ensure that big steps, non-accurate PG estimates, and large heterogeneous settings[4] will not produce unstabilizing policy updates. We summarize such conditions in the following theorem.

Theorem 3 (Per-iteration stabilizing controllers): Let Assumptions 1-4 hold. Suppose that the step-size is set such that $\eta \leq \eta_{\text{gap}}$. In addition, suppose that the heterogeneity and smoothing radius satisfy $\epsilon_{\text{het}} \leq \frac{\gamma\lambda\Delta_0^{\vee}}{64}$ and $r^2 \leq \frac{\gamma\lambda\Delta_0^{\vee}}{64h_{\text{grad}}^2}$, respectively. Then, Algorithm 1 produces a stabilizing controller $\bar{K}_n \in \mathcal{S}$ for all iterations $n \in \{0,1,\ldots,N-1\}$.

The proof for this theorem is detailed in Appendix C of [32], where given an initial stabilizing controller, we exploit an induction approach along with Definition 1 to derive the necessary conditions on the step-size η, smoothing radius r

[4] The authors in [12] discuss the necessity of a low heterogeneity regime when designing stabilizing controllers in the multi-agent setting.

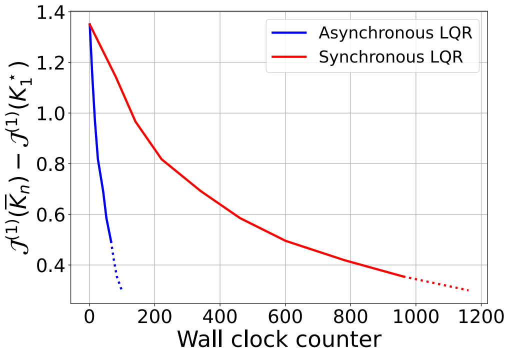


Fig. 2. Optimality gap (with respect to the nominal system and cost matrices) as a function of the wall clock counter, for the asynchronous LQR design (Algorithm 1) and the synchronous LQR approach [12], both in the presence of a single straggler system with $\tau_{\max} = 20$. We set $(\epsilon_A = 5.46, \epsilon_B = 2.74, \epsilon_Q = 3.96, \epsilon_R = 2.82) \times 10^{-2}$ and $b_s = 20$.

and heterogeneity ϵ_{het} that ensure the design of stabilizing controllers $\bar{K}_n$, for all iterations n of Algorithm 1.

V. Numerical Experiments

We highlight the effect of the staleness $\tau_{\max}$, batch size b_s and heterogeneity level ϵ_{het} on the speed of convergence and optimality gap of Algorithm 1. We also illustrate the benefit of asynchronous aggregation over the synchronous counterpart [12], when dealing with straggler systems in the learning process. To this end, let us first consider *nominal* system matrices:[5]

$$\underbrace{\begin{bmatrix} 1.22 & 0.03 & -0.02 & -0.32 \\ 0.01 & 0.47 & 4.70 & 0.00 \\ 0.02 & -0.06 & 0.40 & 0.00 \\ 0.01 & -0.04 & 0.72 & 1.55 \end{bmatrix}}_{A^{(1)}}, \underbrace{\begin{bmatrix} 0.01 & 0.99 \\ -3.44 & 1.66 \\ -0.83 & 0.44 \\ -0.47 & 0.25 \end{bmatrix}}_{B^{(1)}},$$

and cost matrices $Q^{(1)} = I_4$ and $R^{(1)} = I_2$. Therefore, by applying random perturbations to $(A^{(1)}, B^{(1)}, Q^{(1)}, R^{(1)})$, with radius $(\epsilon_A, \epsilon_B, \epsilon_Q, \epsilon_R)$, we generate $M = 100$ tuples $(A^{(i)}, B^{(i)}, Q^{(i)}, R^{(i)})$, for $i \in [M]$, to characterize our heterogeneous multi-agent LQR setting. See Appendix D of [32] for more details on our experimental setup.

With M system and cost matrices in hands, we first compare the proposed asynchronous LQR design of Algorithm 1 over the synchronous federated LQR approach in [12], both in the presence of a single straggler system with $\tau_{\max} = 20$. Since, in Algorithm 1, the server performs controller updates upon receiving the fastest b_s PG estimates, such quicker systems are not affected by straggler systems when $\tau_{\max}$ is sufficiently large. To illustrate this, Figure 2 shows how long, in terms of a wall clock counter, Algorithm 1 takes to design a controller $\bar{K}_N$ that achieves a certain optimality gap compared to the synchronous LQR design. This figure shows that due to the presence of stragglers, the synchronous federated LQR approach [12] needs to *wait* a long time for all of the M PG estimates to be reported to the server to then proceed with the controller update. On the other hand, Algorithm 1 fully enjoys the parallelism of distributed computation, even when dealing with slow systems. However, as discussed, this comes with the price of aggregating stale PG estimates. Figure 3 illustrates the effect of $\tau_{\max}$ in the optimality gap of Algorithm 1.

Figure 3 depict the optimality gap $\Delta_n^{(1)}$ as a function of the iteration count n, for a varying: (a) staleness $\tau_{\max}$, (b) batch size b_s, and (c) heterogeneity level ϵ_{het}. Note that, for convenience, we evaluate the global convergence of Algorithm 1 on the nominal system and cost, i.e., $i = 1$. However, we emphasize that a similar result should also be observed for any $i \in [M]$. Figure 3-(a) shows the impact of $\tau_{\max}$ in the number of iterations N needed to achieve a certain optimality gap; it highlights that the number of iterations required to design a controller $\bar{K}_N$ such that $\Delta_N^{(1)} \leq 0.3$ is larger when $\tau_{\max} = 3$ and $\tau_{\max} = 10$ compared to $\tau_{\max} = 1$. Moreover, as predicted in Corollary 2, the staleness $\tau_{\max}$ only affects the speed of convergence and does not impact the accuracy in the optimality gap $\Delta_N^{(i)}$.

Furthermore, in alignment with Corollary 1, Figure 3-(b) illustrates the benefit of aggregating multiple system's PG estimates. As predicted, an increase in the batch size b_s leads to a faster convergence of Algorithm 1. Lastly, as illustrated in Figure 3-(c), due to the heterogeneous setting, Algorithm 1 returns an ϵ-near optimal controller up to a heterogeneity bias. Therefore, as ϵ_{het} increases, the unavoidable bias (Corollary 2) also increases.

VI. Conclusions and Future Work

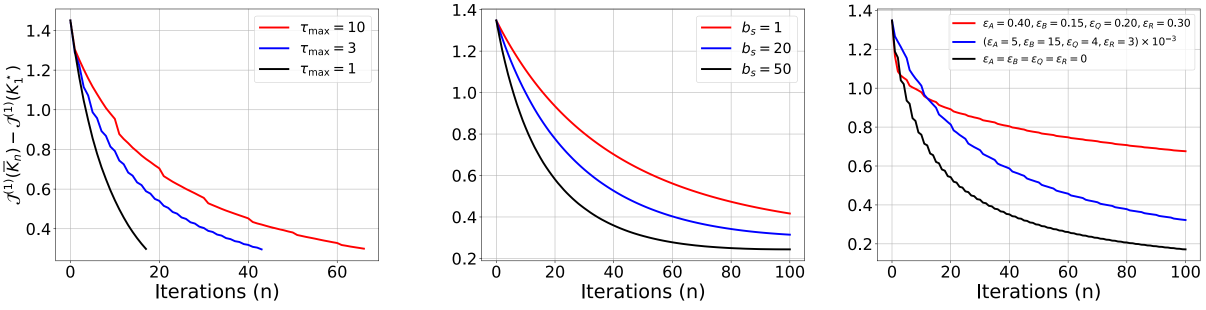
To understand how aggregating stale policy gradient estimates affect model-free LQR design, we characterized the convergence and stability guarantees of an asynchronous and heterogeneous PG method applied to the multi-agent LQR problem. Despite straggler systems, the proposed asynchronous aggregation scheme fully exploits the parallelism in the distributed computation (see Figure 2). Nevertheless, such parallelism comes with the price of aggregating stale policy gradient estimates. Our analysis demonstrated that, by carefully controlling the step-size, the staleness effect remains limited to a high-order term of the ergodic convergence rate (Corollary 1). Moreover, the optimality gap bound remains untouched as in the synchronous case [12]. We showed that the staleness impacts the speed of convergence through a multiplicative factor (Corollary 2). We provided numerical results to illustrate and validate our theory (i.e., Figure 3), where we also highlight the effect of the heterogeneity level ϵ_{het} and batch size b_s to the convergence of Algorithm 1. Future work may explore other aggregation schemes, beyond a simple average, to alleviate the staleness effect even more in the local and global convergence bounds.

References

[1] R. S. Sutton, D. McAllester, S. Singh, and Y. Mansour, "Policy gradient methods for reinforcement learning with function approximation," *Advances in neural information processing systems*, vol. 12, 1999.

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

[5] Code: https://github.com/jd-anderson/AsyncLQR.



(a) Varying $\tau_{\max}$

(b) Varying b_s

(c) Varying ϵ_{het}

Fig. 3. Optimality gap with respect to the iteration count n. (a) $b_s = 20$ and $(\epsilon_A = 5.46, \epsilon_B = 2.74, \epsilon_Q = 3.96, \epsilon_R = 2.82) \times 10^{-2}$. (b) $\tau_{\max} = 1$ and $(\epsilon_A = 5.25, \epsilon_B = 2.80, \epsilon_Q = 4.00, \epsilon_R = 2.82) \times 10^{-3}$. (c) $\tau_{\max} = 5$ and $b_s = 20$.

[3] 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*. PMLR, 2018, pp. 1467–1476.

[4] H. Mohammadi, A. Zare, M. Soltanolkotabi, and M. R. Jovanović, "Convergence and sample complexity of gradient methods for the model-free linear–quadratic regulator problem," *IEEE Transactions on Automatic Control*, vol. 67, no. 5, pp. 2435–2450, 2021.

[5] 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 *The 22nd international conference on artificial intelligence and statistics*. PMLR, 2019, pp. 2916–2925.

[6] B. Gravell, P. M. Esfahani, and T. Summers, "Learning optimal controllers for linear systems with multiplicative noise via policy gradient," *IEEE Transactions on Automatic Control*, vol. 66, no. 11, pp. 5283–5298, 2020.

[7] L. F. Toso, H. Wang, and J. Anderson, "Oracle Complexity Reduction for Model-free LQR: A Stochastic Variance-Reduced Policy Gradient Approach," *arXiv preprint arXiv:2309.10679*, 2023.

[8] H. Wang, L. F. Toso, and J. Anderson, "Fedsysid: A federated approach to sample-efficient system identification," in *Learning for Dynamics and Control Conference*. PMLR, 2023, pp. 1308–1320.

[9] L. F. Toso, H. Wang, and J. Anderson, "Learning personalized models with clustered system identification," in *2023 62nd IEEE Conference on Decision and Control (CDC)*. IEEE, 2023, pp. 7162–7169.

[10] T. T. Zhang, L. F. Toso, J. Anderson, and N. Matni, "Sample-efficient linear representation learning from non-IID non-isotropic data," in *The Twelfth International Conference on Learning Representations*, 2024.

[11] Y. Chen, A. M. Ospina, F. Pasqualetti, and E. Dall'Anese, "Multi-Task System Identification of Similar Linear Time-Invariant Dynamical Systems," *arXiv preprint arXiv:2301.01430*, 2023.

[12] H. Wang, L. F. Toso, A. Mitra, and J. Anderson, "Model-free Learning with Heterogeneous Dynamical Systems: A Federated LQR Approach," *arXiv preprint arXiv:2308.11743*, 2023.

[13] L. F. Toso, D. Zhan, J. Anderson, and H. Wang, "Meta-Learning Linear Quadratic Regulators: A Policy Gradient MAML Approach for the Model-free LQR," *arXiv preprint arXiv:2401.14534*, 2024.

[14] T. T. Zhang, K. Kang, B. D. Lee, C. Tomlin, S. Levine, S. Tu, and N. Matni, "Multi-task imitation learning for linear dynamical systems," in *Learning for Dynamics and Control Conference*. PMLR, 2023, pp. 586–599.

[15] Y. Tang, Z. Ren, and N. Li, "Zeroth-order feedback optimization for cooperative multi-agent systems," *Automatica*, vol. 148, p. 110741, 2023.

[16] L. Wang, K. Zhang, A. Zhou, M. Simchowitz, and R. Tedrake, "Fleet Policy Learning via Weight Merging and An Application to Robotic Tool-Use," *arXiv preprint arXiv:2310.01362*, 2023.

[17] A. Mitra, L. Ye, and V. Gupta, "Towards model-free lqr control over rate-limited channels," *arXiv preprint arXiv:2401.01258*, 2024.

[18] H. Wang, S. Marella, and J. Anderson, "Fedadmm: A federated primal-dual algorithm allowing partial participation," in *2022 IEEE 61st Conference on Decision and Control (CDC)*. IEEE, 2022, pp. 287–294.

[19] J. Nguyen, K. Malik, H. Zhan, A. Yousefpour, M. Rabbat, M. Malek, and D. Huba, "Federated learning with buffered asynchronous aggregation," in *International Conference on Artificial Intelligence and Statistics*. PMLR, 2022, pp. 3581–3607.

[20] M. T. Toghani and C. A. Uribe, "Unbounded gradients in federated learning with buffered asynchronous aggregation," in *2022 58th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*. IEEE, 2022, pp. 1–8.

[21] I. Ziemann, A. Tsiamis, H. Sandberg, and N. Matni, "How are policy gradient methods affected by the limits of control?" in *2022 IEEE 61st Conference on Decision and Control (CDC)*. IEEE, 2022, pp. 5992–5999.

[22] A. Agarwal and J. C. Duchi, "Distributed delayed stochastic optimization," *Advances in neural information processing systems*, vol. 24, 2011.

[23] S. Chaturapruek, J. C. Duchi, and C. Ré, "Asynchronous stochastic convex optimization: the noise is in the noise and sgd don't care," *Advances in Neural Information Processing Systems*, vol. 28, 2015.

[24] S. Stich, A. Mohtashami, and M. Jaggi, "Critical parameters for scalable distributed learning with large batches and asynchronous updates," in *International Conference on Artificial Intelligence and Statistics*. PMLR, 2021, pp. 4042–4050.

[25] H. Karimi, J. Nutini, and M. Schmidt, "Linear convergence of gradient and proximal-gradient methods under the polyak-łojasiewicz condition," in *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2016, Riva del Garda, Italy, September 19-23, 2016, Proceedings, Part I 16*. Springer, 2016, pp. 795–811.

[26] X. Ma, Q. Wang, H. Sun, R. Q. Hu, and Y. Qian, "Csmaafl: Client scheduling and model aggregation in asynchronous federated learning," *arXiv preprint arXiv:2306.01207*, 2023.

[27] A. Adibi, N. D. Fabbro, L. Schenato, S. Kulkarni, H. V. Poor, G. J. Pappas, H. Hassani, and A. Mitra, "Stochastic approximation with delayed updates: Finite-time rates under markovian sampling," *arXiv preprint arXiv:2402.11800*, 2024.

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

[29] C. Stein, "A bound for the error in the normal approximation to the distribution of a sum of dependent random variables," in *Proceedings of the Sixth Berkeley Symposium on Mathematical Statistics and Probability, Volume 2: Probability Theory*, vol. 6. University of California Press, 1972, pp. 583–603.

[30] K. Mishchenko, F. Bach, M. Even, and B. E. Woodworth, "Asynchronous sgd beats minibatch sgd under arbitrary delays," *Advances in Neural Information Processing Systems*, vol. 35, pp. 420–433, 2022.

[31] A. Koloskova, S. U. Stich, and M. Jaggi, "Sharper convergence guarantees for asynchronous sgd for distributed and federated learning," *Advances in Neural Information Processing Systems*, vol. 35, pp. 17 202–17 215, 2022.

[32] L. F. Toso, H. Wang, and J. Anderson, "Asynchronous Heterogeneous Linear Quadratic Regulator Design," *arXiv preprint arXiv:2404.09061*, 2024.