2023 59th Annual Allerton Conference on Communication, Control, and Computing (Allerton) | 979-8-3503-2814-1/23/$31.00 ©2023 IEEE | DOI: 10.1109/Allerton58177.2023.10313509

Federated Certainty Equivalence Control for Linear Gaussian Systems with Unknown Decoupled Dynamics and Quadratic Common Cost

Xinghan Wang, Nasimeh Heydaribeni, Farinaz Koushanfar, Tara Javidi

***Abstract*— We study the decentralized linear quadratic control problem with unknown transition dynamics where agents have partial state observations and can control partial inputs over an additive channel. We propose an algorithm that strikes a balance between system identification (exploration) and certainty equivalence control (exploitation) in order to minimize regret and communication cost. Our non-asymptotic analysis demonstrates that regret of our algorithm scales at a rate of $\mathcal{O}(\sqrt{T})$ for a time horizon of T, while maintaining low communication between agents. Numerical analysis provides validation for our regret analysis and facilitates comparisons between various exploration strategies.**

I. Introduction

Decentralized multi-agent systems are ubiquitous across various applications such as decentralized control of robots and drones [1], [2], decentralized autonomous vehicles [3], non-cooperative games [4], among others. Extensive research in the literature has focused on decentralized multi-agent systems with known system dynamics, exploring various frameworks such as decentralized optimal control [5], [6], [7], multi-agent planning [8], and non-cooperative game theory [9]. However, realistically, the environment model is often only partially observed or entirely unknown. Multi-agent reinforcement Learning (MARL) [10] is designed to address the broader context of multi-agent sequential decision-making, where the agents lack complete knowledge of the environment model. In such conditions, agents learn the environment by interacting with the system and gathering rewards.

In this study, we focus on decentralized LQ problem, which is a multi-agent learning problem with linear system dynamics and quadratic cost function. In particular, we consider a scenario with two agents, each of which observes the system state partially. The system is controlled by the collective actions of the two agents, i.e., the sum of the actions. Leveraging LQ systems as a learning benchmark holds considerable advantages due to their theoretical tractability and extensive relevance across diverse engineering domains. Our particular problem arises for instance when robots that are decoupled in their dynamics and observations are tasked with collaborating to have coupled behavior.

Reinforcement Learning (RL) and MARL can be broadly categorized into two approaches: model-based and model-free. In model-free approaches [11], the policy is directly optimized by interacting with the environment and reward collection. Conversely, model-based approaches [12] involve learning the environment model through interaction with the system and subsequently determining the optimal policy based on the estimated system model. The model-based approaches for decentralized LQ problem are very limited. In [13], [14] the authors consider a decentralized multi-agent system with nested information structure. Our work compliments this line of work by considering a more general information structure. In the absence of nested information structure, there needs to be carefully designed collaboration strategies in place for the agents to learn the system collaboratively.

In this paper, we propose FedCE, a collaborative learning policy which allows agents to efficiently explore and exploit. In our proposed algorithm, we partition time into exploitation and exploration intervals, carefully designing their durations to achieve high performance in both system model learning and minimizing regret. During the exploration phase, we employ Least Square Estimation (LSE) techniques to obtain local partial system model estimates. These estimates are then shared between agents at the end of each exploration interval. Subsequently, both agents compute Certainty Equivalence controllers, which they apply during the exploitation interval. As time progresses and the model estimates improve, the relative length of exploration intervals compared to exploitation intervals decreases, leading to reduced communication between agents over time. We analyze FedCE in terms of its regret bound and demonstrate that the regret scales at a rate of $\mathcal{O}(\sqrt{T})$ for a time horizon of T.

In summary, our contributions in this paper are as follows.

- We propose FedCE, the first federated (decentralized yet collaborative) algorithm consisting of three building blocks; (1) Certainty equivalence controller for exploitation, (2) Coordinated exploration, and (3) Communication and knowledge sharing.
- We show that it is possible to schedule the algorithm components such that the rate of communication and control policy computation is negligible.
- We show that the regret in FedCE scales at a rate of $\mathcal{O}(\sqrt{T})$ for a time horizon of T.
- We provide results confirming the value of collaboration between the two agents.
- We provide extensive numerical analysis that support our theoretical results.

The remainder of the paper is structured as follows. In

This work was supported partially by NSF TILOS AI Institute, AutoCOMBOT MURI Grant, and ONR Award N00014-22-1-2363.

The authors are with the department of Electrical and Computer Engineering, University of California, San Diego, 9500 Gilman Drive, La Jolla, CA 92093, USA.

x2wang, nheydaribeni, fkoushanfar, tjavidi @ucsd.edu

Section II, we review the problem of linear quadratic (LQ) control with known system model and the associated optimal linear controller. In Section III, we formally introduce the problem of decentralized LQ control with unknown system model. We then propose our adaptive algorithm for the considered problem in Section IV, which we call FedCE. In Section V, we analyze the learning of FedCE algorithm, as well as the growth rate of the regret. We conclude with extensive simulations in Section VI that validate our theoretical analysis.

A. Notation

We use lowercase and uppercase letters for vectors and matrices, respectively. We denote the i-th element of vector v by v_i. We use $||\cdot||$ to denote the Euclidean norm for vectors and matrices. $\mathcal{N}(\theta, \sigma)$ denotes the pdf of a Gaussian random variable with mean θ and variance σ^2. We denote the trace of matrix A by $tr(A)$. The identity matrix of size $n \times n$ is denoted by I_n.

II. Preliminaries

In this section, we provide some background on the well-known problem of centralized LQ control as the baseline of the problem that we have studied in this paper and present the known results in this area.

A. Centralized LQ control with known system model

Consider a discrete-time linear system with the following system dynamics:

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

where $x(t) \in \mathbb{R}^n$ is the state of the system, $u(t) \in \mathbb{R}^d$ is the control input from some controller, and $w(t) \in \mathbb{R}^n$ is the random disturbance which is sampled i.i.d. from a Gaussian distribution $\mathcal{N}(0, \sigma_w I_n)$. $A \in \mathbb{R}^{n\times n}$ and $B \in \mathbb{R}^{n\times d}$ are matrices representing the system model. We denote the system model by $\theta = [A, B]^T$.

The step cost incurred at time t, given current state $x(t)$ and input $u(t)$, is a quadratic function defined as

$$c(t) = x(t)^T Q x(t) + u(t)^T R u(t). \tag{2}$$

The goal is to find a control policy $\pi : \mathbb{R}^n \to \mathbb{R}^d$, that minimizes the infinite-horizon average expected cost:

$$J^\pi(\theta) = \limsup_{T\to\infty} \frac{1}{T} \mathbb{E}^\pi [\sum_{t=1}^{T} c(t)] \tag{3}$$

It is common to make the following assumptions to solve the above problem.

Assumption 1: *The cost matrices Q and R are symmetric and positive definite. Without loss of generality, we assume that the minimum singular value of R is greater than or equal to 1.*

Assumption 2: *The system dynamics (A, B) is stabilizable, i.e. there exists K such that $(A + BK)$ is stable.*

When θ is known, given Assumptions 1 and 2, we have a well-studied stochastic Linear Quadratic Control problem, where it is known that the optimal controller is a linear feedback controller (Linear Quadratic Regulator, or LQR) of the form: $u(t) = -K(\theta)x(t)$, where $K(\theta)$ is given by

$$K(\theta) = (R + B^T P(\theta) B)^{-1} B^T P(\theta) A, \tag{4}$$

and P^* is the solution to the Algebraic Riccati Equation:

$$\begin{aligned} P(\theta) = Q + A^T P(\theta) A \\ - A^T P(\theta) B (R + B^T P(\theta) B)^{-1} B^T P(\theta) A. \end{aligned} \tag{5}$$

We also know that the optimal cost under this LQR is:

$$J^*(\theta) = \sigma_w^2 tr(P(\theta)) \tag{6}$$

B. Centralized Adaptive LQ control with unknown system model

When the system model $\theta = [A, B]^T$ is unknown, we are facing an online setting where the system dynamics need to be learned as the agent interacts with the system. The performance of a policy π is measured by the cumulative regret in horizon T, which is defined as:

$$\mathcal{R}(\pi, T) = \sum_{t=1}^{T} \Big[c(t) - J^*(\theta) \Big] \tag{7}$$

which is the difference between the cost incurred by π and the optimal infinite horizon averaged cost. The goal is to find adaptive policy π that minimizes the cumulative regret.

III. Problem Statement: Decentralized LQ control with unknown system model

In this paper, we study the decentralized version of the LQ problem stated in section II with unknown system model. Consider a discrete-time linear system jointly controlled by two agents, each having a partial observation of the system state. The system dynamics is given by

$$\begin{aligned} \begin{bmatrix} x_1(t+1) \\ x_2(t+1) \end{bmatrix} = \begin{bmatrix} A_{11} & 0 \\ 0 & A_{22} \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} + \\ \begin{bmatrix} B_{11} & 0 \\ 0 & B_{22} \end{bmatrix} \Big(u_1(t) + u_2(t) \Big) + \begin{bmatrix} w_1(t) \\ w_2(t) \end{bmatrix} \end{aligned} \tag{8}$$

Each agent i observes a partial state $x_i(t) \in \mathbb{R}^{n_i}$, with $\sum_{i=1}^{2} n_i = n$ and $A^{ii} \in \mathbb{R}^{n_i \times n_i}$ for $i = 1, 2$. Furthermore, each agent i is associated with a local control input $u_i(t) \in \mathbb{R}^d$, and $B^{ii} \in \mathbb{R}^{n_i \times d}$. The joint control input to the system is $u(t) = u_1(t) + u_2(t)$. We assume the additive noise is sampled i.i.d. from a Gaussian distribution $\mathcal{N}(0, \sigma_w I_n)$. We assume that the system model matrices are unknown to both of the agents.

Let $x(t) = \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix}$, $w(t) = \begin{bmatrix} w_1(t) \\ w_2(t) \end{bmatrix}$, $A^* = \begin{bmatrix} A_{11} & 0 \\ 0 & A_{22} \end{bmatrix}$, $B^* = \begin{bmatrix} B_{11} & 0 \\ 0 & B_{22} \end{bmatrix}$, the system dynamics can be rewritten as follows.

$$x(t+1) = A^* x(t) + B^* u(t) + w(t) \tag{9}$$

Given state $x(t)$ and joint control $u(t)$, the agents receive a quadratic cost of

$$c(t) = x(t)^T Q x(t) + u(t)^T R u(t) \tag{10}$$

We assume the above problem satisfies assumptions 1, 2.

Given the decentralized nature of the problem, the goal is to find control policies $\pi(t) = (\pi_1(t), \pi_2(t))$, where $\pi_i(t) : \mathbb{R}^{n_i} \to \mathbb{R}^d$ and $u_i(t) = \pi_i(t)[x_i(t)]$, that minimizes the infinite-horizon average expected cost

$$J^\pi(\theta) = \limsup_{T\to\infty} \frac{1}{T} \mathbb{E}^\pi [\sum_{t=1}^{T} c(t)] \tag{11}$$

In the rest of the paper, we will refer to the above problem as the DecLQ problem.

In the next lemma, we present the optimal controllers when the system model $\theta^* = [A^*, B^*]^T$ is known. The structure of this controller will later be used for the case when the system model is not known.

Lemma 1: *In DecLQ problem, if the system model $\theta^* = [A^*, B^*]^T$ is known, the optimal controllers are given by*

$$u_1(t) = \begin{bmatrix} K_{11}(\theta^*) \\ K_{21}(\theta^*) \end{bmatrix} x_1(t), u_2(t) = \begin{bmatrix} K_{12}(\theta^*) \\ K_{22}(\theta^*) \end{bmatrix} x_2(t) \tag{12}$$

where $K(\theta^*) = \begin{bmatrix} K_{11}(\theta^*) & K_{12}(\theta^*) \\ K_{21}(\theta^*) & K_{22}(\theta^*) \end{bmatrix}$ is calculated from equations (4) *and* (5)*. We also have the optimal cost to be $J^*(\theta^*)$, which is the optimal cost of the centralized controller.*

Proof: When θ^* is known, one can look at DecLQ problem from a centralized agent's point of view. Meaning, assume we have a centralized agent that observes both $x_1(t)$ and $x_2(t)$ and generates the control action $u(t)$. Therefore, we know that the optimal controller will be $u(t) = K(\theta^*)x(t)$. If we denote $K(\theta^*) = \begin{bmatrix} K_{11}(\theta^*) & K_{12}(\theta^*) \\ K_{21}(\theta^*) & K_{22}(\theta^*) \end{bmatrix}$, we can write

$$u(t) = \begin{bmatrix} K_{11}(\theta^*) \\ K_{21}(\theta^*) \end{bmatrix} x_1(t) + \begin{bmatrix} K_{12}(\theta^*) \\ K_{22}(\theta^*) \end{bmatrix} x_2(t).$$

We denote $u_1(t) = \begin{bmatrix} K_{11}(\theta^*) \\ K_{21}(\theta^*) \end{bmatrix} x_1(t)$ and $u_2(t) = \begin{bmatrix} K_{12}(\theta^*) \\ K_{22}(\theta^*) \end{bmatrix} x_2(t)$. Therefore, in a decentralized case, if θ^* is known, both agents can compute the matrix $K(\theta^*)$ and generate their corresponding control actions that will collectively achieve the optimal cost of the centralized controller, $J^*(\theta^*)$ (which is the best one can achieve in a decentralized setting). ■

As mentioned before, the structure of the optimal control for the case where the system model is known will be used in the algorithm that we propose for the the case where the system model is unknown. If θ^* is unknown, agents 1 and 2 need to learn it in order to minimize the infinite horizon average cost. The performance of the learners is measured by the cumulative regret over horizon T,

$$\mathcal{R}(\pi, T) = \sum_{t=1}^{T} \Big[c(t) - J^*(\theta^*) \Big] \tag{13}$$

IV. Adaptive Algorithm for DecLQ

We propose an algorithm for the DecLQ problem, which is based on the certainty equivalence controller used in the literature for the centralized LQ problem. When the system model is unknown, the agent(s) need to learn the system model over time by interacting with the system through an adaptive algorithm. The learning usually happens by forming a belief over the system model and updating it over time. This belief will then be used to construct a controller, which is usually time-varying and adaptive due to the belief evolving by time. There are different ways that one can utilize the belief over the system model to construct the controller. Thomson Sampling (TS) [15], [16], Optimization in the Face of Uncertainty (OFU) [17], [18], [19], and Certainty Equivalence (CE) [20], [21] are the most well-known algorithms used in the literature. We note that since in CE, one only needs to have an estimation of the system model, it is sufficient to only keep track of the estimation as opposed to an entire belief over the system model. We will use CE to build our adaptive algorithm for the DecLQ problem. We refer to this algorithm as the FedCE (Federated Certainty Equivalence) algorithm. Before describing FedCE algorithm, we highlight the main ideas used in its design and how they contribute to the overall performance of the algorithm.

Key Idea 1, Coordinated Exploration: It is evident that in our decentralized decoupled setting with partial state observation, the agents can not learn the system dynamics if both of them are exerting actions in all dimensions. The reason is that the actions are not shared with each other and the ambiguity in the control action of the other agent prevents them from learning the system. Instead, we devised coordinated exploration intervals in which, each agent only controls its own part of the state and stays silent for the other parts. During these intervals, we have a completely decoupled system in terms of dynamics and the agents can learn their corresponding system dynamics.

Key Idea 2, Federated Learning: In FedCE, agents learn the whole system model in a federated fashion. As mentioned, in the exploration intervals, the agents learn their corresponding model dynamics. However, in order to learn the optimal controller, the agents need the whole system dynamics. Therefore, they learn the other parts of the system dynamics through federated learning.

Key Idea 3, Balanced Exploration vs Exploitation: Due to the decentralized and decouples nature of our system model, the agents can only learn the system dynamics in the exploration intervals. Therefore, the longer the exploration intervals, the faster agents can learn the system dynamics. In addition, during the exploration intervals, the actions are not optimal or even near optimal. Therefore, the agents incur

a linear regret over those intervals. Consequently, we need to set a balance between the exploration and exploitation interval durations.

A. FedCE Algorithm

FedCE is summarized in Alg. 1 and in the following, we will explain it in detail. FedCE algorithm consists of three different phases. A **Warm-up** period to form a good initial estimation of the system model; **Certainty Equivalence (CE)** phase to exploit the model estimation and apply semi-optimal control actions; and **Exploration** phase to learn the system model. Note that CE and Exploration phases are distributed across separate time intervals.

Warm-up: In the Warm-up period, which starts from time $t = 0$ and ends at $t = T_w$ for some predetermined T_w, agents apply decoupled actions to enable decentralized learning of the system model (system identification). Decoupled actions are defined as

$$u_1(t) = \begin{bmatrix} u_1^{(1)}(t) \\ 0 \end{bmatrix} \text{ and } u_2(t) = \begin{bmatrix} 0 \\ u_2^{(2)}(t) \end{bmatrix}, \qquad (14)$$

where $u_i^{(i)}(t) \sim \sigma_i^1$ and σ_i^1 is a predetermined probability distribution over $\mathbb{R}^{n_i}$.

During Warm-up period, each agent learns its corresponding system model using the least square estimate (LSE) method. In particular, agent 1 learns $\theta_1^* = [A_{11}, B_{11}]^T$ and agent 2 learns $\theta_2^* = [A_{22}, B_{22}]^T$ according to the sequential LSE method, in which the estimates on the system model, denoted by $\hat{\theta}_1(t)$ and $\hat{\theta}_2(t)$ for θ_1^* and θ_2^*, respectively, are updated sequentially for every new decoupled observation. We denote $\phi_i(t) = \begin{bmatrix} x_i(t) \\ u_i^{(i)}(t) \end{bmatrix}$ for $i = 1, 2$. The LSE update rule is as follows for $i = 1, 2$.

$$\hat{\theta}_i(t+1) = \hat{\theta}_i(t) + V_i(t)^{-1}\phi_i(t)(x_i(t+1)^T - \phi_i(t)^T\hat{\theta}_i(t)) \quad (15a)$$

$$V_i(t+1) = V_i(t) + \phi_i(t)\phi_i(t)^T \quad (15b)$$

After the estimation updates at $t = T_w$, the agents share their corresponding model estimates with each other. Therefore, at the end of the warm-up period, both agents have the estimates $\hat{\theta}$ on the entire system model, θ^*. We drop the time dependence to show the latest (most up-to-date) estimates. For example, at $t = T_w$ we have $\hat{\theta}_i = \hat{\theta}_i(T_w)$.

Certainty Equivalence (CE): In the CE phase, the agents will compute the certainty equivalence controllers, $K(\hat{\theta})$, using their latest model estimate, $\hat{\theta}$, using equations (4) and (5). We note that the CE controller is the optimal controller for a hypothetical system with system model equal to $\hat{\theta}$. The control actions will be generated according to the computed CE controller. This phase is distributed into intervals of length T_{CE}^n, where n denotes the n^{th} CE interval. Each CE interval is followed by an interval of Exploration phase, which will be explained next.

Exploration: Exploration phase is similar to Warm-up phase but it is distributed into intervals of length T_{Exp}^n, where n denotes the n^{th} Exploration phase. Each Exploration interval follows a CE interval and the agents apply the decoupling actions of Warm-up phase defined in (14) with $u_i^{(i)}(t) \sim \sigma_i^n$, and σ_i^n is a predetermined probability distribution over $\mathbb{R}^{n_i}$. We will specify the required conditions on σ_i^n in the next section. The agents will then update their corresponding system model estimations according to equation (15). At the end of each Exploration interval, agents share their updated system model estimations with each other.

The Exploration intervals should get sparser as time increases because the model estimates become better through time, and therefore, there is less need for exploration. In Section V, we will describe the necessary conditions for the scheduling method and an example of such scheduling to have the desired regret behavior.

V. Convergence and Regret Analysis

In order to analyze the regret of FedCE algorithm, we first describe the conditions that we need to put on the exploration action distributions σ_i^n, for $i = 1, 2$ and $n = 1, 2, \cdots$, and on the CE/Exploration interval scheduling; More specifically, the values of T_{CE}^n and T_{Exp}^n, for $n = 1, 2, \cdots$. We define $(\hat{u}_i^{(i)}(s))_{s=1,2,\cdots}$ to be agent i's control action sequence during Exploration intervals (s stands for the s_{th} exploration time). In particular, we have $\hat{u}_i^{(i)}(s) = u_i^{(i)}(t)$, where t corresponds to the s_{th} exploration time. We refer to $(\hat{u}_i^{(i)}(s))_{s=1,2,\cdots}$ as the exploration control action sequence.

The next two definitions are extracted from [22].

Definition 1 (Persistent Excitation): *The exploration control action sequence $(\hat{u}_i^{(i)}(s))_{s=1,2,\cdots}$ is said to be persistently exciting if there exists a positive definite matrix U such that for large S, we have*

$$\frac{1}{S}\sum_{s=1}^{S} \hat{u}_i^{(i)}(s)\hat{u}_i^{(i)}(s)^T \geq U \qquad (16)$$

The above definition states the condition when the exploration control actions are rich enough to facilitate the system model learning.

The next definition is about the stability of the system during the exploration intervals.

Definition 2 (Exploration Stability): *The exploration control action sequence $(\hat{u}_i^{(i)}(s))_{s=1,2,\cdots}$ is said to satisfy*

Algorithm 1 FedCE

Inputs: σ_i^n, T_{CE}^n, and T_{Exp}^n for $n = 1, 2, \cdots$.
for $t = 0, \cdots, T_w$ **do**
Let $u_1(t) = \begin{bmatrix} u_1^{(1)}(t) \\ 0 \end{bmatrix}$, $u_2(t) = \begin{bmatrix} 0 \\ u_2^{(2)}(t) \end{bmatrix}$,
$u_i^{(i)}(t) \sim \sigma_i^1$, for $i = 1, 2$.
Agent 1 learns A_{11}, B_{11} using Eq. (15).
Agents 2 learns A_{22}, B_{22} using Eq. (15).
end for
Agents merge their estimations to get $\hat{\theta} = [\hat{A}, \hat{B}]^T$.
for $n = 1, 2, \cdots$ **do**
Compute $K(\hat{\theta})$ according to Eq. (4).
for $s = 1, \cdots, T_{CE}^n$ **do** $\Longrightarrow$ *CE Phase*
$t = t + 1$.
Let

$$u_1(t) = \begin{bmatrix} K_{11}(\hat{\theta}) \\ K_{21}(\hat{\theta}) \end{bmatrix} x_1(t) \text{ and } u_2(t) = \begin{bmatrix} K_{12}(\hat{\theta}) \\ K_{22}(\hat{\theta}) \end{bmatrix} x_2(t).$$

end for
for $s = 1, \cdots, T_{Exp}^n$ **do** $\Longrightarrow$ *Exploration Phase*
$t = t + 1$.
Let $u_1(t) = \begin{bmatrix} u_1^{(1)}(t) \\ 0 \end{bmatrix}$, $u_2(t) = \begin{bmatrix} 0 \\ u_2^{(2)}(t) \end{bmatrix}$,
$u_i^{(i)}(t) \sim \sigma_i^n$, for $i = 1, 2$
Agent 1 learns A_{11}, B_{11} using Eq. (15).
Agents 2 learns A_{22}, B_{22} using Eq. (15).
end for
Agents merge their estimations to get $\hat{\theta} = [\hat{A}, \hat{B}]^T$.
end for

the Exploration Stability condition if there exists a positive definite matrix V such that for large S, we have

$$\frac{1}{S} \sum_{s=1}^{S} \hat{u}_i^{(i)}(s) \hat{u}_i^{(i)}(s)^T \leq V \tag{17}$$

Definition 3 (Rich and Stable Exploration): *If Persistent Excitation and Exploration Stability conditions hold for the control sequence $(\hat{u}_i^{(i)}(s))_{s=1,2,\cdots}$ that is generated according to the distributions $\hat{u}_i^{(i)}(s) \sim \sigma_i^n$, when s^{th} exploration time belongs to the n^{th} exploration episode, then $(\sigma_i^n)_{n=1,2,\cdots}$ satisfies the Rich and Stable Exploration condition.*

Using the above definitions, we can have the following lemma on the system model learning rate of FedCE.

Lemma 2: *Let A^* and B^* be the true system model matrices, and $\hat{A}(s)$ and $\hat{B}(s)$ be least square mean estimates of A^* and B^* using s samples of the system (after s exploration times). Assume that the Rich and Stable Exploration condition holds for $(\sigma_i^n)_{n=1,2,\cdots}$ for $i = 1, 2$. Then with probability close to 1, $||A^* - \hat{A}(s)||$ and $||B^* - \hat{B}(s)||$ are bounded by $\mathcal{O}(\sqrt{\frac{1}{s}})$ for large enough s.*

Proof: According to [22], if Persistent Excitation and Exploration Stability conditions hold, then the estimation $\hat{\theta}_i(s)$ converges to the true system model θ_i^* almost surely, and we can write the following.

$$\hat{\theta}_i(s) = \theta_i^* + (\sum_{k=1}^{s} \phi_i(k)\phi_i(k)^T)^{-1} \sum_{k=1}^{s} \phi_i(k) w(k)^T \tag{18}$$

Persistent Excitation condition indicates that $sU \leq \sum_{k=1}^{s} \phi_i(k)\phi_i(k)^T \leq sV$ and consequently, $\lambda_{max}((\sum_{k=1}^{s} \phi_i(k)\phi_i(k)^T)^{-1}) \leq \frac{1}{s} l$ for some positive l, and $tr((\sum_{k=1}^{s} \phi_i(k)\phi_i(k)^T)) \leq sm$ for some positive m. By writing equation (18) for column c of $\hat{\theta}_i(s) - \theta_i^*$, denoted by $\hat{\theta}_i(s)[:, c] - \theta_i^*[:, c]$, and taking the L^2 norm, we have

$$||(\hat{\theta}_i(s)[:, c] - \theta_i^*[:, c])||_2^2 \tag{19}$$

$$\leq \frac{1}{s^2} l^2 (\sum_{k=1}^{s} \phi_i(k)^T w(k)[c]) (\sum_{k=1}^{s} \phi_i(k) w(k)[c]) \tag{20}$$

$$\leq \frac{1}{s^2} l^2 (\sum_{k=1}^{s} \phi_i(k)^T \phi_i(k) w(k)[c]^2) \tag{21}$$

$$\leq \frac{1}{s^2} l^2 (\sum_{k=1}^{s} \phi_i(k)^T \phi_i(k)) \gamma \sigma_w^2 \leq \frac{1}{s} l^2 m \gamma \sigma_w^2 \tag{22}$$

The last two inequalities are valid with a probability close to 1, and this probability quickly approaches 1 as γ is increased. When γ is set to 4, the probability is nearly equal to 0.9999.

Therefore, $||A^* - \hat{A}(s)||$ and $||B^* - \hat{B}(s)||$ are bounded by $\mathcal{O}(\sqrt{\frac{1}{s}})$ for large enough s. ■

In Lemma 2, we provided the learning rate with respect to the number of exploration times. The next theorem provides a bound on the learning rate with respect to the total time horizon, T.

Theorem 1 (Bounded Error): *If Rich and Stable Exploration condition holds for and if the total number of exploration times is $\mathcal{O}(\sqrt{T})$, where T is the total time horizon, then $||A^* - \hat{A}(T)||$ and $||B^* - \hat{B}(T)||$ are bounded by $\mathcal{O}(T^{-\frac{1}{4}})$.*

We skip the proof of the above theorem since it is evident by using Lemma 2.

In the next theorem, we present the bound on the regret of the FedCE algorithm.

Theorem 2 (Regret Bound): *Assume that Rich and Stable Exploration condition holds for $(\sigma_i^n)_{n=1,2,\cdots}$ for $i = 1, 2$. Also, assume that the total exploration time is $\mathcal{O}(\sqrt{T})$ for*

the total time horizon of T. Then the expected regret of the proposed FedCE algorithm satisfies

$$\mathcal{R}(\text{FedCE}, T) = \mathcal{O}(\sqrt{T}) \tag{23}$$

Proof: The regret of Algorithm 1 consists of two parts that we denote by $\mathcal{R}_{CE}$ and $\mathcal{R}_{Exp}$. In particular, we have

$$\mathcal{R}(\text{FedCE}, T) = \mathcal{R}_{CE}(\text{FedCE}, T) + \mathcal{R}_{Exp}(\text{FedCE}, T).$$

The first part, $\mathcal{R}_{CE}$, is related to CE intervals and the fact that we have an error in estimating the model and consequently, we have an error in constructing the optimal controller through certainty equivalence. The second part of the regret, $\mathcal{R}_{Exp}$, is related to Exploration phase due to not exerting the optimal action over those intervals. In the following, we will compute these two parts of the regret separately.

In order to compute $\mathcal{R}_{CE}$, we borrow some results from [20]. In particular, we use Theorem 2 of this reference, which gives an upper bound on the suboptimality of the certainty equivalence controller as a function of the model estimation error. In that theorem, they show that if the model estimation error is less than or equal to ϵ, then the average cost incurred through CE controller, denoted by J_{CE}, minus the optimal average cost, denoted by J^*, is bounded by a quadratic function of ϵ. We know from Theorem 1 that the model estimation error scales as $\mathcal{O}(T^{\frac{-1}{4}})$. Therefore, $J_{CE} - J^*$ scales as $\mathcal{O}(T^{\frac{-1}{2}})$. From the definition of regret, we have $\mathcal{R}_{CE} = \mathcal{O}(T(J_{CE} - J^*))$. Therefore, we have $\mathcal{R}_{CE} = \mathcal{O}(T * T^{\frac{-1}{2}}) = \mathcal{O}(\sqrt{T})$.

It is easy to see that the second part of the regret, $\mathcal{R}_{Exp}$ scales linearly with the time of exploration. Note that this is the worst possible regret. The reason is that for every time step that we do not exert the optimal action, we pay a constant regret. Furthermore, assuming the state has deviated too much from the CE path (the path it would have taken if we were only applying CE controller), we can force it back to its path with at most the same time steps that have caused the deviation. Therefore, this part of the regret would be linear in the exploration time and we have $\mathcal{R}_{Exp} = \mathcal{O}(T_{Exp})$, where T_{Exp} is the total exploration time. Based on the assumption in the theorem, we have $T_{Exp} = \mathcal{O}(\sqrt{T})$. Consequently, we have $\mathcal{R}_{Exp} = \mathcal{O}(\sqrt{T})$.

Combining the above results for $\mathcal{R}_{CE}$ and $\mathcal{R}_{Exp}$, we have $\mathcal{R}(\text{FedCE}, T) = \mathcal{O}(\sqrt{T})$. ∎

A. Satisfying Rich and Stable Exploration

In this subsection, we provide an example of probability distributions $(\sigma_i^n)_{n=1,2,\cdots}$ for $i = 1, 2$, and CE/Exploration scheduling, i.e., T_{EX}^n and T_{Exp}^n for $n = 1, 2, \cdots$, that satisfies the Rich and Stable Exploration. According to Theorem 2, we also need to have the total exploration time to be $\mathcal{O}(\sqrt{T})$. Therefore, our example must also satisfy this condition.

Lemma 3: *If we have σ_i^n to be an element-wise Bernouli distribution over the set $\{a^n, -a^n\}$ with parameter $p^n = 0.5$, $a^n = \max\{\frac{a^1}{\sqrt{n}}, \delta\}$ for some positive a^1, and $T_{Exp} = kn$, then Rich and Stable Exploration condition holds.*

Proof: Consider the n^{th} exploration episode containing times $b+1, \cdots, b+T_{Exp}^n$. For large enough n, we have

$$\sum_{k=b+1}^{b+T_{Exp}^n} \hat{u}_i^{(i)}(k)\hat{u}_i^{(i)}(k)^T \approx T_{Exp}^n Cov(\hat{u}_i^{(i)}(b+1))$$

We also have

$$\delta^2 I_{n_i} \leq Cov(\hat{u}_i^{(i)}(b+1)) \leq (a^1)^2 I_{n_i}$$

Consequently, for large enough s, we can write

$$\delta^2 I_{n_i} \leq \frac{1}{s}\sum_{k=1}^{s} \hat{u}_i^{(i)}(k)\hat{u}_i^{(i)}(k)^T \leq (a^1)^2 I_{n_i}$$

Setting $U = \delta^2 I_{n_i}$ and $V = (a^1)^2 I_{n_i}$ will conclude the proof. ∎

Lemma 4: *If we set $T_{Exp}^n = kn$ and $T_{EX}^n = kn^3$, then the total exploration time is $\mathcal{O}(\sqrt{T})$.*

Proof: We denote the total time horizon with n CE/Exploration intervals with $T(n)$, and we have $T(n) = T_w + \sum_{l=1}^{n}(T_{CE}^n + T_{Exp}^n)$. We also denote the total exploration time in $T(n)$ by $T_{Exp}(n)$. According to the scheduling of $T_{CE}^n = kn^3$ and $T_{Exp}^n = kn$, we have $T(n) = \mathcal{O}(n^4)$, and $T_{Exp}(n) = \mathcal{O}(n^2)$. Therefore, we have $T_{Exp}(n) = \mathcal{O}(\sqrt{T(n)})$. We can drop n and write $T_{Exp} = \mathcal{O}(\sqrt{T})$, where T_{Exp} is the total exploration time in T total steps. ∎

VI. Numerical Analysis

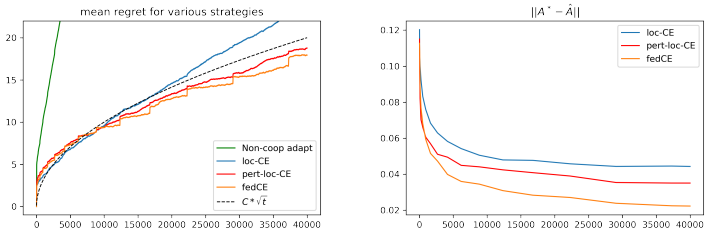
In this section, we illustrate the performance of our proposed FedCE algorithm through numerical simulations. We use the following dynamics and cost matrices:

$$A = \begin{bmatrix} 0.95 & 0 \\ 0 & 0.95 \end{bmatrix}, B = I, Q = 0.05 * \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}, R = I \tag{24}$$

We set the horizon to be $T = 40000$ and the length of the the warm-up phase $T_w = 20$. Each simulation setting is averaged over 100 independent runs, and the cumulative regret $\mathcal{R}(\text{FedCE}, T)$ and estimation error $||A - \hat{A}||$ are recorded. We set the initial state $x(0) = \mathbf{0}$ and the process

noise variance $\sigma_w = 0.2$. We use the example of Section V-A for σ_i^n, T_{EX}^n, and T_{Exp}^n for FedCE algorithm (set $\delta = 0.01$ and consider different k and a^1 in our experiments). In our numerical analysis results, when we refer to FedCE, we are referring to this case. We further compare this version with cases when other alternative examples are utilized. We compare the performance of FedCE with the following alternative algorithms:

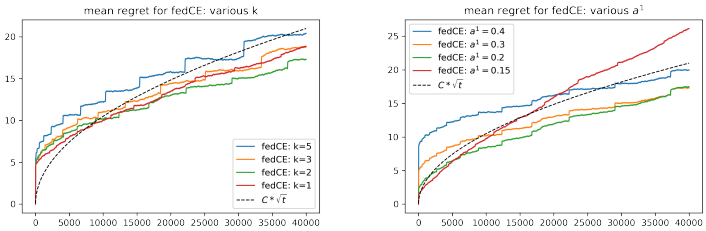
- **Non-cooperative Adaptive Algorithm (Non-Coop-Adapt)**: Similar to FedCE, except for the control policy applied in the CE intervals. Instead of applying CE controller, apply the sub-optimal control: $u_1(t) = \begin{bmatrix} K(\hat{\theta}^{11}) \\ 0 \end{bmatrix} x_1(t), u_2(t) = \begin{bmatrix} 0 \\ K(\hat{\theta}^{22}) \end{bmatrix} x_2(t)$. Where $K(\hat{\theta}^{11}), K(\hat{\theta}^{22})$ are computed from Eq.(4) with $(\hat{A}^{11}, \hat{B}^{11}, Q^{11}, R^{11})$, $(\hat{A}^{22}, \hat{B}^{22}, Q^{22}, R^{22})$ respectively.
- **Local Certainty Equivalence (Loc-CE)**: Similar to FedCE but during Exploration intervals, set $u_1(t) = \begin{bmatrix} K^{11}(\hat{\theta}) \\ 0 \end{bmatrix} x_1(t), u_2(t) = \begin{bmatrix} 0 \\ K^{22}(\hat{\theta}) \end{bmatrix} x_2(t)$.
- **Perturbed Local Certainty Equivalence (Pert-Loc-CE)**: Similar to FedCE but during Exploration intervals, set $u_1(t) = \begin{bmatrix} K^{11}(\hat{\theta}) \\ 0 \end{bmatrix} x_1(t) + \begin{bmatrix} I \\ 0 \end{bmatrix} \eta_1(t), u_2(t) = \begin{bmatrix} 0 \\ K^{22}(\hat{\theta}) \end{bmatrix} x_2(t) + \begin{bmatrix} 0 \\ I \end{bmatrix} \eta_2(t)$, where $\eta_1(t), \eta_2(t) \sim \mathcal{N}(0, a^n)$ for the n^{th} exploration episode.



(a) **(b)**

Fig. 1: Performance of FedCE and other alternative algorithms. (a) Mean Regret, (b) Model Estimation Error.

In Figure 1, we show the mean regret and model estimation error of FedCE and other alternative algorithms. For the analysis in this figure, we set $a^1 = 0.2$, $k = 2$ using a grid search. Figure 1(a) verifies the $\mathcal{O}(\sqrt{T})$ regret bound for FedCE as indicated by Theorem 2. We also show in this figure that the regret of **Non-Coop-Adapt** scales almost linearly, resulting from the fact that it treats the cost matrices as completely decoupled, incurring linear regret from the off-diagonal entries of Q, R. This result shows that sublinear regret is hopeless without communication between the two agents. Allowing communication significantly improves regret, as shown by **Loc-CE** and **Pert-Loc-CE** plots. However, the choice of exploration actions in **Loc-CE** provide insufficient excitation and therefore hinders the learning of system dynamics, as illustrated in 1(b). Although the added perturbations in **Pert-Loc-CE** improved the performance of the algorithm compared to **Loc-CE**, this exploration scheme performs much worse than FedCE. Since the exploration in **Loc-CE** and **Pert-Loc-CE** is based on CE controller, the actions are relatively "conservative", meaning the states are pulled back towards the lowest cost state (in this case **0**) with little magnitude, causing smaller signal to noise ratios and consequently inaccurate system identification from least squares.



(a) **(b)**

Fig. 2: Ablation studies of FedCE under different (**a**) scaling factor k, and (**b**) initial excitation a^1.

In Figure 2, we carry out an ablation study on Exploration/CE length scaling factor k and initial excitation a^1. In Figure 2(a), we study the effect of k on the regret. As k becomes smaller, exploration episodes are more frequent but less lengthy, creating a trade-off between the exploration regret and system learning. In Figure 2(b), FedCE is initialized with different a^1. Similarly, there appears to be a trade-off between system model learning (better with larger a^1) and the regret per exploration period (more with larger a^1).

In Figure 3, we apply FedCE to the same system as Figure 1, with the following differences. $x(0) = [0, 10]'$, and $Q = 0.05 * \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}$. For this system the lowest cost state is when $x_1 = x_2$. We study the effect of R, i.e. cost matrix for actions, on the speed by which x_1 and x_2 converge to each other. As R takes larger magnitudes, larger actions are more penalized, therefore x_1 and x_2 converge more slowly since it is preferred to take several smaller steps instead of a giant leap.

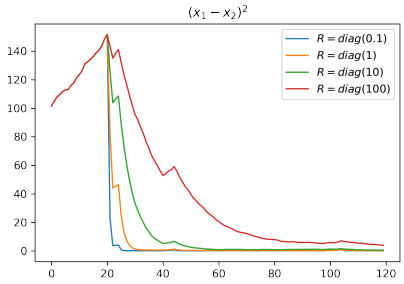
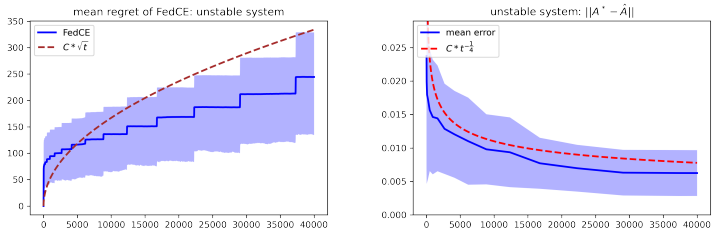


Fig. 3: Convergence of x_1 and x_2 to each other for various R.



(a) **(b)**

Fig. 4: Performance of FedCE for an unstable system.

In Figure 4, we apply FedCE to an unstable system. with $A = \begin{bmatrix} 1.05 & 0 \\ 0 & 1.05 \end{bmatrix}$. Due to the instability of the system, each exploration phase incurs more regret since random actions can bring the system further from the lowest cost state. Despite this fact, FedCE is still able to achieve a sublinear regret bounded by $\mathcal{O}(\sqrt{T})$, and the rate of learning is of the order of $\mathcal{O}(T^{-\frac{1}{4}})$ that matches the theoretical results from Theorem 1.

REFERENCES

[1] Christopher Amato, George Konidaris, Gabriel Cruz, Christopher A Maynor, Jonathan P How, and Leslie P Kaelbling. Planning for decentralized control of multiple robots under uncertainty. In *2015 IEEE international conference on robotics and automation (ICRA)*, pages 1241–1248. IEEE, 2015.

[2] Marco Aggravi, Giuseppe Sirignano, Paolo Robuffo Giordano, and Claudio Pacchierotti. Decentralized control of a heterogeneous human–robot team for exploration and patrolling. *IEEE Transactions on Automation Science and Engineering*, 19(4):3109–3125, 2021.

[3] Eugene Vinitsky, Nathan Lichtlé, Kanaad Parvate, and Alexandre Bayen. Optimizing mixed autonomy traffic flow with decentralized autonomous vehicles and multi-agent reinforcement learning. *ACM Transactions on Cyber-Physical Systems*, 7(2):1–22, 2023.

[4] Takako Fujiwara-Greve. *Non-cooperative game theory*. Springer, 2015.

[5] Nader Motee, Ali Jadbabaie, and Bassam Bamieh. On decentralized optimal control and information structures. In *2008 American Control Conference*, pages 4985–4990. IEEE, 2008.

[6] Xiaofan Wu, Florian Dörfler, and Mihailo R Jovanović. Input-output analysis and decentralized optimal control of inter-area oscillations in power systems. *IEEE Transactions on Power Systems*, 31(3):2434–2444, 2015.

[7] Serdar Yüksel and Tamer Basar. Stochastic networked control systems. *AMC*, 10:12, 2013.

[8] Frans A Oliehoek, Christopher Amato, et al. *A concise introduction to decentralized POMDPs*, volume 1. Springer, 2016.

[9] Peng Hang, Chen Lv, Yang Xing, Chao Huang, and Zhongxu Hu. Human-like decision making for autonomous driving: A noncooperative game theoretic approach. *IEEE Transactions on Intelligent Transportation Systems*, 22(4):2076–2087, 2020.

[10] Kaiqing Zhang, Zhuoran Yang, and Tamer Başar. Multi-agent reinforcement learning: A selective overview of theories and algorithms. *Handbook of reinforcement learning and control*, pages 321–384, 2021.

[11] Bin Hu, Kaiqing Zhang, Na Li, Mehran Mesbahi, Maryam Fazel, and Tamer Başar. Toward a theoretical foundation of policy optimization for learning control policies. *Annual Review of Control, Robotics, and Autonomous Systems*, 6:123–158, 2023.

[12] Thomas M Moerland, Joost Broekens, Aske Plaat, Catholijn M Jonker, et al. Model-based reinforcement learning: A survey. *Foundations and Trends® in Machine Learning*, 16(1):1–118, 2023.

[13] Seyed Mohammad Asghari, Yi Ouyang, and Ashutosh Nayyar. Regret bounds for decentralized learning in cooperative multi-agent dynamical systems. In *Conference on Uncertainty in Artificial Intelligence*, pages 121–130. PMLR, 2020.

[14] Seyed Mohammad Asghari, Mukul Gagrani, and Ashutosh Nayyar. Regret analysis for learning in a multi-agent linear-quadratic control problem. In *2020 American Control Conference (ACC)*, pages 3926–3931. IEEE, 2020.

[15] Yasin Abbasi-Yadkori and Csaba Szepesvári. Bayesian optimal control of smoothly parameterized systems. In *UAI*, pages 1–11, 2015.

[16] Marc Abeille and Alessandro Lazaric. Improved regret bounds for thompson sampling in linear quadratic control problems. In *International Conference on Machine Learning*, pages 1–9. PMLR, 2018.

[17] Yasin Abbasi-Yadkori and Csaba Szepesvári. Regret bounds for the adaptive control of linear quadratic systems. In *Proceedings of the 24th Annual Conference on Learning Theory*, pages 1–26. JMLR Workshop and Conference Proceedings, 2011.

[18] Mohamad Kazem Shirani Faradonbeh, Ambuj Tewari, and George Michailidis. Optimism-based adaptive regulation of linear-quadratic systems. *IEEE Transactions on Automatic Control*, 66(4):1802–1808, 2020.

[19] Morteza Ibrahimi, Adel Javanmard, and Benjamin Roy. Efficient reinforcement learning for high dimensional linear quadratic systems. *Advances in Neural Information Processing Systems*, 25, 2012.

[20] Horia Mania, Stephen Tu, and Benjamin Recht. Certainty equivalence is efficient for linear quadratic control. *Advances in Neural Information Processing Systems*, 32, 2019.

[21] Mohamad Kazem Shirani Faradonbeh, Ambuj Tewari, and George Michailidis. Input perturbations for adaptive control and learning. *Automatica*, 117:108950, 2020.

[22] PR Kumar and Pravin Varaiya. Stochastic systems: estimation, identification and adaptive control, 1986.