Stochastic Linear Quadratic Optimal Control Problem: A Reinforcement Learning Method

Na Li, Xun Li, Jing Peng, Zuo Quan Xu

***Abstract*— This paper adopts a reinforcement learning (RL) method to solve infinite horizon continuous-time stochastic linear quadratic problems, where the drift and diffusion terms in the dynamics may depend on both the state and control. Based on Bellman's dynamic programming principle, we present an online RL algorithm to attain optimal control with partial system information. This algorithm computes the optimal control rather than estimates the system coefficients and solves the related Riccati equation. It only requires local trajectory information, which significantly simplifies the calculation process. We shed light on our theoretical findings using two numerical examples.**

***Index Terms*— Reinforcement learning, stochastic optimal control, linear quadratic problem.**

I. Introduction

Reinforcement learning (RL), rooted in animal learning and early learning control work, has attached great attention to machine learning research. Unlike other machine learning techniques such as supervised and unsupervised learning, the RL method focuses on optimizing the reward without explicitly exploiting the hidden structure. Trial-and-error search and delayed rewards are the most prominent features of RL. One discovers the best strategy through trials and errors, and his actions affect not only the immediate reward but also all later rewards. In this approach, the controller must first *exploit* his experience to give the control and then, based on the reward, *explore* new strategies for the future. The most significant challenge is the trade-off between exploitation and exploration. See [18], [24], [29] for details.

Optimal control, along with regulation and tracking problems, is among the most important research topics in control theory ([3], [35]). When the appropriate model is not available, indirect and direct, adaptive control techniques are utilized to provide the best control. The indirect method seeks to discover the system's structure and then derives the optimal control using the discovered system's information. By contrast, the direct method does not identify the structure of the system; instead, it adjusts the control directly to make the error between the plant output and the desired output tend to zero asymptotically (see [21]).

According to Sutton *et al.* [25], RL methods may be seen as a direct approach to optimal control problems, as it computes the optimal controls directly without knowing the system's structure. The significance of RL methods is that it provides a new adaptive structure, which successively reinforces the reward function such that the adaptive controller converges to the optimal control. In comparison, indirect adaptive techniques must first estimate the system's structure before determining the control, which is intrinsically complicated.

Linear quadratic (LQ) problem is an essential class of optimal control problems in both theory and practice, since a wide range of nonlinear problems can be approximated by the linear problem. This paper proposes an RL algorithm to solve stochastic LQ (SLQ) optimal control problems.

A. Related Work

RL techniques have been extensively explored under both discrete-time and continuous-time frameworks for deterministic optimal control problems. Bradtke *et al.* [4] presented a Q-learning policy iteration for a discrete-time LQ problem by the so-called Q-function (Watkins [31], Werbos [32]). For its recent applications to discrete-time models, we refer to Rizvi and Lin [22], Luo *et al.* [16], Kiumarsia *et al.* [13]. Baird [2] first adopted an RL approach to obtain the optimal control for a continuous-time discrete-state system. Murray *et al.* [20] proposed an iterative adaptive dynamic programming (ADP) scheme for non-linear systems. Recently, a number of new RL methods have been developed for optimal control problems in continuous-time cases (e.g., [7], [11], [14], [17], [28], [34]). Vrabie *et al.* [28] developed a new policy iteration technique for continuous-time linear systems under partial information. Jiang and Jiang [11] studied a type of nonlinear polynomial system and proposed a novel ADP based on the Hamilton-Jacobi-Bellman equation of a relaxed problem. Modares *et al.* [17] designed a model-free off-policy RL algorithm for a linear continuous-time system. Their method is also applicable to regulation and tracking problems. We refer to Kiumarsi *et al.* [12] and Chen *et al.* [6] for more related works.

A critical approach to obtain optimal control of SLQ problems on the infinite horizon is to solve the related stochastic algebraic Riccati equation (SARE). Using analytical and computational approaches, Ait Rami and Zhou [1] tackled an indefinite SLQ control problem to treat the related SARE via semidefinite programming (SDP). Later, Sun and Yong [23] proved that the admissible control set is non-empty for every initial state, equivalent to the control system's stabilizability. Because SAREs are dependent on the coefficients in the dynamics and the cost functional, the algorithms based on SAREs must be implemented offline.

Duncan *et al.* [8] solved an SLQ problem for a linear diffusion system by applying an indirect method. In this case, the coefficients of the drift term are unknown, and the diffusion term is independent of the state and control. Recently, academics have been increasingly interested in studying SLQ problems using RL techniques, even though many applications are highly restricted compared to deterministic problems. Wong and Lee [33] addressed a discrete-time SLQ problem with white Gaussian signals by Q-learning in

N. Li acknowledges financial support from the NSFC (No. 12171279, No. 11801317), the Natural Science Foundation of Shandong Province (No. ZR2019MA013), and the Colleges and Universities Youth Innovation Technology Program of Shandong Province (No. 2019KJI011). X. Li acknowledges financial support from the Research Grants Council of Hong Kong under grants (No. 15213218, No. 15215319, No. 15216720), and PolyU 1-TA03 and 4-ZZKR. J. Peng and Z. Q. Xu acknowledge financial support from the NSFC (No. 11971409), the Hong Kong RGC (GRF No. 15204216, No. 15202817), and the PolyU-SDU Joint Research Center on Financial Mathematics and the CAS AMSS-POLYU Joint Laboratory of Applied Mathematics, The Hong Kong Polytechnic University.

N. Li is with School of Statistics and Mathematics, Shandong University of Finance and Economics, Jinan, Shandong, 250014, China (e-mail: naibor@163.com).

X. Li is with Department of Applied Mathematics, The Hong Kong Polytechnic University, Hong Kong, China (e-mail: li.xun@polyu.edu.hk).

J. Peng is with Department of Applied Mathematics, The Hong Kong Polytechnic University, Hong Kong, China (e-mail: jing.peng@connect.polyu.hk).

Z. Q. Xu is with Department of Applied Mathematics, The Hong Kong Polytechnic University, Hong Kong, China (e-mail: maxu@polyu.edu.hk).

a direct approach. Fazel *et al.* [9] studied a time-homogeneous LQ regulator (LQR) problem with a random initial state and found the optimal policy by a model-free local search method. The method provides the global convergence for the decent gradient methods and a higher convergence rate than the naive gradient method. Later, Mohammadi *et al.* [19] gave a random search method with two-point gradient estimates for continuous-time LQR problems. They improved the related works on the required function evaluations and simulation time. Wang *et al.* [29] applied an RL technique to a non-linear stochastic continuous-time diffusion system based on the classical relaxed stochastic control (see, for example, [10], [36]) such that the optimum trade-off is accomplished between investigating the black box environment and using present information. Following up with [29], Wang and Zhou [30] derived a Gaussian feedback exploration policy to solve a continuous-time mean-variance portfolio optimization problem.

B. Motivation

There are two primary motivations for studying the continuous-time SLQ problem by the RL method in this paper, which are listed in the following two paragraphs.

The most notable advantage of the LQ framework is that the optimal controls can usually be expressed by an explicit closed-form. To obtain the optimal control, one has to solve the related Riccati equation via SDP, referring to [1]. This approach requires all the information of the system. However, we sometimes only know the observation of the state process rather than all of the system's characteristics. Therefore, the SDP method may be impractical. As mentioned earlier, utilizing RL techniques can generate the optimal control only by the trajectory information. This idea motivates us to build a new RL algorithm to compute the optimal control directly rather than solve the Riccati equation. More precisely, the RL algorithm can learn what to do based on data along the trajectories; no complete system knowledge is required to implement our algorithm.

As mentioned in Subsection I-A, Duncan *et al.* [8] studied an SLQ problem where the diffusion term is independent of the state and control. In financial and economic practice, however, decision makers' actions usually impact the trend of the system (drift term) and the uncertainty of the system (diffusion term). Therefore, it is necessary to consider the case where the diffusion term is affected by both the state and control. This case motivates us to analyze a more comprehensive linear system where drift and diffusion terms depend on the state and control in this paper. The problem can also be viewed as the scenario where multiplicative noises appear in the state and control. Noises frequently have a multiplicative effect on various plant components; see a practical example in [27]. Moreover, due to the presence of control in the diffusion term, the weighting matrix R in the problem is allowed to be indefinite, which is a crucial instance in both theory and practice; see, for example, Chen *et al.* [5], and Yong and Zhou [35].

C. Contribution

Inspired by the above related work, especially [20] and [28], this paper develops an online RL algorithm to solve SLQ problems over an infinite time horizon, which primarily uses stochastic Bellman dynamic programming (DP) rather than solves the related Riccati equation. The algorithm computes the optimal control based on the local trajectories rather than the system's structure. In other words, our algorithm only focuses on getting the optimal control regardless of modelling the system's internal structure. In practice, the controller only needs partial information of the system dynamics to get the optimal control by updating policy and improving the evaluator based on the online data of state trajectories. Our main contributions are stated as follows.

(i) The policy iteration is implemented along the trajectories online using only the system's partial information. To the best of our knowledge, this is the first time to study SLQ problems for Itô type continuous-time system with the state and control in the diffusion term by RL methods. As a byproduct, the solution of the Riccati equation is also derived.

(ii) Our algorithm only needs the local exploration over the time interval $[t, t+\Delta t]$, with $t \geq 0$ and $\Delta t > 0$ arbitrarily chosen. The stochastic DP allows us to adopt the optimality equation as the policy evaluation to reinforce a *target function* over a short interval $[t, t+\Delta t]$, rather than reinforce the cost functional on the entire infinite time horizon $[t, +\infty)$. Only local trajectory information is required in this scenario, significantly simplifying the calculation processing.

(iii) Given a mean-square stabilizing control at initial, we prove that all the following up controls are stabilizable by our policy improvement. In contrast, Wang *et al.* [29] did not discuss the stabilizable issue in their case. Also, the convergence of the controls in our RL algorithm is proved.

(iv) Similar to Fazel *et al.* [9] and Mohammadi *et al.* [19], our RL algorithm is also partially model-free. Fazel *et al.* [9] and Mohammadi *et al.* [19] studied the problems with the random initial state in discrete-time and continuous-time, respectively. Differently, we study the Itô type linear system with the diffusion term and deterministic initial state. Moreover, the SLQ problem is also different from [33], in which the system is only disturbed by white Gaussian signals.

The rest of this paper is organized as follows. Section II presents an SLQ problem and gives an online RL algorithm to compute its optimal feedback control. We also discuss the stabilizability and convergence properties of the algorithm. We implement the algorithm and provide two numerical examples in Section III.

Notation. Let $\mathbb{N}$ denote the set of positive integers. Let $l, m, n, k, L, M, N, \mathcal{K} \in \mathbb{N}$ be the given constants. We denote by $\mathbb{R}^n$ the n-dimensional Euclidean space with the norm $\|\cdot\|$. Let $\mathbb{R}^{n\times m}$ be the set of all $n \times m$ real matrices. We denote by $A^\top$ the transpose of a vector or matrix A, where $\top$ denotes the transpose. Let $\mathcal{S}^n$ be the collection of all symmetric matrices in $\mathbb{R}^{n\times n}$. As usual, if a matrix $A \in \mathcal{S}^n$ is positive semidefinite (resp. positive definite), we write $A \geq 0$ (resp. > 0). All the positive semidefinite (resp. positive definite) matrices are collected by $\mathcal{S}^n_+$ (resp. $\mathcal{S}^n_{++}$). If A, $B \in \mathcal{S}^n$, then we write $A \geq B$ (resp. $>$) if $A - B \geq 0$ (resp. > 0). Denote $s, t \geq 0$ as the time on infinite horizon. Let $(\Omega, \mathcal{F}, \mathbb{P}, \mathbb{F})$ be a complete filtered probability space on which a one-dimensional standard Brownian motion $W(\cdot)$ is defined with $\mathbb{F} \equiv \{\mathcal{F}_t\}_{t\geq 0}$ being its natural filtration augmented by all $\mathbb{P}$-null sets. We define the Hilbert space $L^2_{\mathbb{F}}(\mathbb{R}^n)$, which is the space of $\mathbb{R}^n$-valued $\mathbb{F}$-progressively measurable processes $\varphi(\cdot)$ with the finite norm $\|\varphi(\cdot)\| = \left[\mathbb{E}\int_t^\infty |\varphi(s)|^2 ds\right]^{\frac{1}{2}} < \infty$. Furthermore, $\mathbf{O}$ denotes zero matrices with appropriate dimensions, and $\varnothing$ denotes the empty set.

II. Online Algorithm for SLQ Optimal Control Problem

In this paper, we consider the following time-invariant stochastic linear dynamical control system

$$\begin{cases} dX(s) = [AX(s) + Bu(s)]\, ds \\ \qquad\qquad + [CX(s) + Du(s)]\, dW(s), \quad s \geq t, \\ X(t) = x \in \mathbb{R}^n, \end{cases} \tag{1}$$

where the coefficients $A,\ C \in \mathbb{R}^{n\times n}$ and $B,\ D \in \mathbb{R}^{n\times m}$ are constant matrices. The state process $X(\cdot)$ is an n-dimensional vector, the control u is an m-dimensional vector, and $X(t) = x$ is the deterministic initial value. On the right side of the system (1), the first term is called the *drift term*, and the second term is called the *diffusion term*. Here, the dimension of Brownian motion is set to be one just for simplicity, and the case of multi-dimensional Brownian motion can be dealt with in the same way. We also denote this system by $[A, C; B, D]$ for simplicity.

Definition 2.1: The system $[A, C; B, D]$ is called mean-square stabilizable (with respect to x) if there exists a constant matrix $K \in \mathbb{R}^{m\times n}$ such that the (unique) strong solution of

$$\begin{cases} dX(s) = (A + BK)\, X(s)dt + (C + DK)\, X(s)dW(s), \quad s \geq t, \\ X(t) = x \end{cases} \tag{2}$$

satisfying $\lim_{s\to\infty} \mathbb{E}[X(s)^\top X(s)] = 0$. In this case, K is called a stabilizer of the system $[A, C; B, D]$ and the feedback control $u(\cdot) = KX(\cdot)$ is called stabilizing. The set of all stabilizers is denoted by $\mathcal{X} = \mathcal{X}([A, C; B, D])$.

The following assumption is used to avoid trivial cases.

Assumption 2.1: The system (1) is mean-square stabilizable, *i.e.*,

$$\mathcal{X}([A, C; B, D]) \neq \varnothing.$$

The following lemma provides an equivalent condition for the existence of the stabilizers for system (1), please refer to Theorem 1 in [1] or Lemma 2.2 in [23].

Lemma 2.1: A matrix $K \in \mathbb{R}^{m\times n}$ is a stabilizer of the system $[A, C; B, D]$ if and only if there exists a matrix $P \in \mathcal{S}^n_{++}$ such that

$$(A+BK)^\top P+P(A+BK)+(C+DK)^\top P(C+DK) < 0.$$

In this case, for any $\Lambda \in \mathcal{S}^n$ (resp., $\mathcal{S}^n_+$, $\mathcal{S}^n_{++}$), the Lyapunov equation $(A+BK)^\top P+P(A+BK)+(C+DK)^\top P(C+DK)+\Lambda = 0$ admits a unique solution $P \in \mathcal{S}^n$ (resp., $\mathcal{S}^n_+$, $\mathcal{S}^n_{++}$).

This result shows that the set $\mathcal{X}([A, C; B, D])$ is, in fact, independent of the initial state x. When the system $[A, C; B, D]$ is mean-square stabilizable, we define the corresponding set of admissible controls as $\mathcal{U}_{ad} = \{u(\cdot) \in L^2_{\mathbb{F}}(\mathbb{R}^m) : u(\cdot) \text{ is stabilizing}\}$. In this paper, we consider a quadratic cost functional given by

$$\begin{aligned} &J(t, x; u(\cdot)) \\ &= \mathbb{E}^{\mathcal{F}_t} \int_t^\infty \Big[X(s)^\top QX(s)+2u(s)^\top SX(s)+u(s)^\top Ru(s)\Big]ds, \end{aligned} \tag{3}$$

where Q, S, and R are given constant matrices of proper sizes.

Problem (SLQ). Given $t \geq 0$ and $x \in \mathbb{R}^n$, find a control $u^*(\cdot) \in \mathcal{U}_{ad}$ such that

$$J(t, x; u^*(\cdot)) = \inf_{u(\cdot)\in\mathcal{U}_{ad}} J(t, x; u(\cdot)) \triangleq V(t, x),$$

where $V(t, x)$ is called the *value function* of Problem (SLQ).

Problem (SLQ) is called well-posed at (t, x) if $V(t, x) > -\infty$. A well-posed problem is called *attainable* if there is a control $u^*(\cdot) \in \mathcal{U}_{ad}$ such that $J(t, x; u^*(\cdot)) = V(t, x)$. In this case, $u^*(\cdot)$ is called an *optimal control* and the corresponding solution of (1), $X^*(\cdot)$ is called the *optimal trajectory* (corresponding to $u^*(\cdot)$), and $(X^*(\cdot), u^*(\cdot))$ is called an *optimal pair*.

Under the condition $R > 0$ and $Q - S^\top R^{-1}S \geq 0$, $V(t, x) \geq 0$ so that Problem (SLQ) is well-posed for any given $t \geq 0$ and $x \in \mathbb{R}^n$. If $R > 0$ and $Q - S^\top R^{-1}S = 0$, then

$$\begin{aligned} &J(t, x; u(\cdot)) \\ &= \mathbb{E}^{\mathcal{F}_t} \int_t^\infty \big[(SX(s)+Ru(s))^\top R^{-1}(SX(s)+Ru(s))\big]ds \geq 0. \end{aligned}$$

Clearly, 0 is a lower bound and can be achieved evidently by the unique optimal control $u^*(\cdot) = -R^{-1}SX(\cdot)$. From now on, we focus on the following case.

Assumption 2.2: $R > 0$ and $Q - S^\top R^{-1}S > 0$.

Problem (SLQ) is discussed under Assumption 2.2 for simplicity. Based on the methods introduced in [1] or [15], the corresponding results in this paper can be extended to the indefinite case. The following lemma is well known; please see Theorem 3.3 in Chapter 4 of [35] or Theorem 13 in [1].

Lemma 2.2: Suppose $P \in \mathcal{S}^n_{++}$ satisfies the following Lyapunov equation

$$\begin{aligned} &(A + BK)^\top P + P(A + BK) + (C + DK)^\top P(C + DK) \\ &\qquad + K^\top RK + S^\top K + K^\top S + Q = 0, \end{aligned} \tag{4}$$

where $K = -(R + D^\top PD)^{-1}(B^\top P + D^\top PC + S)$. Then $u(\cdot) = KX(\cdot)$ is an optimal control of Problem (SLQ) and $V(t, x) = x^\top Px$. Moreover, the Bellman's DP is

$$\begin{aligned} x^\top Px = \mathbb{E}^{\mathcal{F}_t}\Big\{ \int_t^{t+\Delta t} &X(s)^\top \Big[Q + 2K^\top S + K^\top RK\Big]X(s)ds \\ &+ X(t + \Delta t)^\top PX(t + \Delta t)\Big\} \end{aligned} \tag{5}$$

for any constant $\Delta t > 0$.

Our key observation is that, based on (5), to compute the value function V is equivalent to calculating P. We only need to know the local state trajectories $X(\cdot)$ on $[t, t + \Delta t]$, therefore it requires us to provide a reasonable online algorithm to solve Problem (SLQ). Indeed, the value of P can be inferred from (5) by the local trajectories of $X(\cdot)$. On the other hand, we can also compute P by the following expression

$$x^\top Px = \mathbb{E}^{\mathcal{F}_t} \int_t^\infty X(s)^\top \Big[Q + 2K^\top S + K^\top RK\Big]\, X(s)ds, \tag{6}$$

which is obtained by sending Δt to infinity in (5). But it requires the entire state trajectories $X(\cdot)$ on $[t, \infty)$.

At each iteration i ($i = 1, 2, \cdots$), the state trajectory is denoted by $X^{(i)}$ corresponding to the control law $K^{(i)}$. Now, we present Algorithm 1 as follows.

Algorithm 1 *Policy Iteration for Problem (SLQ)*

1: **Initialization:** Select any stabilizer $K^{(0)}$ for the system (1).
2: Let $i = 0$ and $\varepsilon > 0$.
3: **do** {
4: Obtain local state trajectories $X^{(i)}$ by running system (1) with $K^{(i)}$ on $[t, t + \Delta t]$.
5: **Policy Evaluation** (Reinforcement): Solve $P^{(i+1)}$ from the identity

$$\begin{aligned} &x^\top P^{(i+1)}x - \mathbb{E}^{\mathcal{F}_t}\big[X^{(i)}(t + \Delta t)^\top P^{(i+1)}X^{(i)}(t + \Delta t)\big] \\ &= \mathbb{E}^{\mathcal{F}_t} \int_t^{t+\Delta t} X^{(i)}(s)^\top \Big[Q + 2K^{(i)\top}S \\ &\qquad + K^{(i)\top}RK^{(i)}\Big]X^{(i)}(s)ds. \end{aligned} \tag{7}$$

6: **Policy Improvement** (Update): Update $K^{(i+1)}$ by the formula

$$\begin{aligned} K^{(i+1)} = -(R + D^\top P^{(i+1)}D)^{-1}(B^\top P^{(i+1)} \\ + D^\top P^{(i+1)}C + S). \end{aligned} \tag{8}$$

7: $i \leftarrow i + 1$.
8: } **until** $\|P^{(i+1)} - P^{(i)}\| < \varepsilon$.

Algorithm 1 is an online algorithm based on local state trajectories, reinforced by Policy Evaluation (7) and updated by Policy Improvement (8). Algorithm 1 has three significant advantages over the offline algorithm: (i) The observation period consisting of an initial time $t \in [0,\infty)$ and a length $\Delta t > 0$ can be freely chosen; (ii) Different from (6) exploring the entire state space on the whole interval $[t,\infty)$, we only need to record local observations of the state on the short period $[t, t+\Delta t]$, which dramatically reduces the computation at each iteration; (iii) This algorithm can be implemented without using any information of A in the system $[A, C; B, D]$, so it is *partially model-free*. Especially when $D = \mathbf{O}$, Algorithm 1 can be implemented without using the information of A and C.

If system (1) is multi-dimensional, e.g., the Brownian motion is $W = (W_1, W_2, \cdots, W_M)$, the diffusion term becomes $\sum_{l=1}^{M}(C_l X + D_l u)dW_l$ with the coefficients $C_l \in \mathbb{R}^{n\times n}$ and $D_l \in \mathbb{R}^{n\times n}$, $l = 1, 2, \cdots, M$, Algorithm 1 needs only one modification: the Policy Improvement (8) is changed to be $K^{(i+1)} = -(R + \sum_{l=1}^{M} D_l^\top P^{(i+1)} D_l)^{-1}(B^\top P^{(i+1)} + \sum_{l=1}^{M} D_l^\top P^{(i+1)} C_l + S)$. The corresponding results in this paper can be proved in the same way for the multi-dimensional case referring to [26].

Lemma 2.3: Suppose that Assumption 2.2 holds and the system $[A, C; B, D]$ is stabilizable with $K^{(i)}$. Then solving Policy Evaluation (7) in Algorithm 1 is equivalent to solving Lyapunov Recursion

$$\begin{aligned}&(A+BK^{(i)})^\top P^{(i+1)} + P^{(i+1)}(A+BK^{(i)})\\ &\quad + (C+DK^{(i)})^\top P^{(i+1)}(C+DK^{(i)})\\ &\quad + K^{(i)\top} R K^{(i)} + S^\top K^{(i)} + K^{(i)\top} S + Q = 0.\end{aligned} \tag{9}$$

Proof Suppose $K^{(i)}$ is a stabilizer for the system (1). By Assumption 2.2, we have

$$\begin{aligned}&K^{(i)\top} R K^{(i)} + S^\top K^{(i)} + K^{(i)\top} S + Q\\ &= Q - S^\top R^{-1} S + (RK^{(i)} + S)^\top R^{-1}(RK^{(i)} + S) > 0.\end{aligned}$$

By Lemma 2.1, Lyapunov Recursion (9) admits a unique solution $P^{(i+1)} \in \mathcal{S}^n_{++}$.

Inserting the feedback control $u^{(i)}(\cdot) = K^{(i)} X^{(i)}(\cdot)$ into the system (1) and applying Itô's formula to $X^{(i)}(\cdot)^\top P^{(i+1)} X^{(i)}(\cdot)$, we have

$$\begin{aligned}&d\left[X^{(i)}(s)^\top P^{(i+1)} X^{(i)}(s)\right]\\ &= \Big\{X^{(i)}(s)^\top \Big[(A+BK^{(i)})^\top P^{(i+1)} + P^{(i+1)}(A+BK^{(i)})\\ &\quad + (C+DK^{(i)})^\top P^{(i+1)}(C+DK^{(i)})\Big] X^{(i)}(s)\Big\} ds\\ &\quad + \{...\}\, dW(s).\end{aligned} \tag{10}$$

Integrating (10) from t to $t+\Delta t$, we obtain

$$\begin{aligned}&X^{(i)}(t+\Delta t)^\top P^{(i+1)} X^{(i)}(t+\Delta t) - x^\top P^{(i+1)} x\\ &= \int_t^{t+\Delta t} X^{(i)}(s)^\top \Big[(A+BK^{(i)})^\top P^{(i+1)} + P^{(i+1)}(A+BK^{(i)})\\ &\quad + (C+DK^{(i)})^\top P^{(i+1)}(C+DK^{(i)})\Big] X^{(i)}(s) ds\\ &\quad + \int_t^{t+\Delta t} \{...\}\, dW(s).\end{aligned}$$

Since $\mathbb{E}^{\mathcal{F}_t} \int_t^{t+\Delta t} \{...\}\, dW(s) = 0$, taking conditional expectation $\mathbb{E}^{\mathcal{F}_t}$ on both sides, one gets

$$\begin{aligned}&\mathbb{E}^{\mathcal{F}_t}[X^{(i)}(t+\Delta t)^\top P^{(i+1)} X^{(i)}(t+\Delta t)] - x^\top P^{(i+1)} x\\ &= \mathbb{E}^{\mathcal{F}_t} \int_t^{t+\Delta t} X^{(i)}(s)^\top \Big[(A+BK^{(i)})^\top P^{(i+1)} + P^{(i+1)}(A+BK^{(i)})\\ &\quad + (C+DK^{(i)})^\top P^{(i+1)}(C+DK^{(i)})\Big] X^{(i)}(s) ds.\end{aligned} \tag{11}$$

From Lyapunov Recursion (9), we have

$$\begin{aligned}&\mathbb{E}^{\mathcal{F}_t}[X^{(i)}(t+\Delta t)^\top P^{(i+1)} X^{(i)}(t+\Delta t)] - x^\top P^{(i+1)} x\\ &= -\mathbb{E}^{\mathcal{F}_t} \int_t^{t+\Delta t} \Big\{X^{(i)}(s)^\top \Big[Q + 2K^{(i)\top} S\\ &\qquad + K^{(i)\top} R K^{(i)}\Big] X^{(i)}(s)\Big\} ds,\end{aligned}$$

which confirms Policy Evaluation (7).

On the other hand, if $P^{(i+1)} \in \mathcal{S}^n$ is the solution of (7), for any constant $\tau > t$, a calculation similar to (11) gives

$$\begin{aligned}&\mathbb{E}^{\mathcal{F}_\tau} \int_\tau^{\tau+\Delta t} X^{(i)}(s)^\top \Big[(A+BK^{(i)})^\top P^{(i+1)} + P^{(i+1)}(A+BK^{(i)})\\ &\quad + (C+DK^{(i)})^\top P^{(i+1)}(C+DK^{(i)})\Big] X^{(i)}(s) ds\\ &+ \mathbb{E}^{\mathcal{F}_\tau} \int_\tau^{\tau+\Delta t} \Big\{X^{(i)}(s)^\top \Big[Q + 2K^{(i)\top} S\\ &\qquad + K^{(i)\top} R K^{(i)}\Big] X^{(i)}(s)\Big\} ds = 0.\end{aligned} \tag{12}$$

Dividing Δt on both sides of (12), we see

$$\begin{aligned}&\frac{1}{\Delta t}\mathbb{E}^{\mathcal{F}_\tau} \int_\tau^{\tau+\Delta t} \Big\{X^{(i)}(s)^\top \Big[(A+BK^{(i)})^\top P^{(i+1)}\\ &+ P^{(i+1)}(A+BK^{(i)}) + (C+DK^{(i)})^\top P^{(i+1)}(C+DK^{(i)})\\ &+ Q + 2K^{(i)\top} S + K^{(i)\top} R K^{(i)}\Big] X^{(i)}(s)\Big\} ds = 0.\end{aligned}$$

Let $\Delta t \to 0$ and denote the state at time τ by x_τ, then

$$\begin{aligned}&x_\tau^\top \Big[(A+BK^{(i)})^\top P^{(i+1)} + P^{(i+1)}(A+BK^{(i)})\\ &\quad + (C+DK^{(i)})^\top P^{(i+1)}(C+DK^{(i)})\\ &\quad + K^{(i)\top} R K^{(i)} + S^\top K^{(i)} + K^{(i)\top} S + Q\Big] x_\tau = 0.\end{aligned}$$

Because x_τ can take any value in $\mathbb{R}^n$, Lyapunov Recursion (9) holds. By Lemma 2.1 and

$$K^{(i)\top} R K^{(i)} + S^\top K^{(i)} + K^{(i)\top} S + Q > 0,$$

we have $P^{(i+1)} \in \mathcal{S}^n_{++}$. □

By Lemma 2.3, solving Lyapunov Recursion (9) with Policy Improvement (8) is equivalent to solving Algorithm 1; that is, they admit the same solution $P^{(i+1)}$ at each iteration. The latter has a significant advantage over the former in that it does not necessitate knowing all the system's information. Indeed, the information of A is embedded in the state trajectories $X^{(i)}$ online, so we can use the observation of state trajectories to reinforce (7) without knowing A in our algorithm. The coefficients B, C, and D are used to update control law $K^{(i+1)}$ in Policy Improvement (8). In particular, C is not required to be known when $D = \mathbf{O}$.

Once a stabilizer $K^{(0)}$ in Algorithm 1 is initialized, one first runs the system repeatedly with the control $K^{(i)}$ from the initial state x and records the resultant state trajectories $X^{(i)}$ on interval $[t, t+\Delta t]$ to reinforce the *target function*:

$$\begin{aligned}&\Delta J^{(i)}(t, t+\Delta t; X^{(i)}, K^{(i)})\\ &:= \mathbb{E}^{\mathcal{F}_t}\Big\{\int_t^{t+\Delta t} X^{(i)}(s)^\top \Big[Q + 2K^{(i)\top} S + K^{(i)\top} R K^{(i)}\Big] X^{(i)}(s) ds\Big\}.\end{aligned} \tag{13}$$

Then one solves $P^{(i+1)}$ by (7) and obtains an updated control $K^{(i+1)}$ by (8). This procedure is iterated until it converges. In this procedure, $\{K^{(i)}\}_{i=1}^{\infty}$ should be the stabilizers of the system $[A, C; B, D]$ of adaptive process at each iteration, i.e., it is necessary to require that $K^{(i)}$ is stepwise stable. The following lemma illustrates the stepwise stable property of $K^{(i)}$.

Theorem 2.1: Suppose that Assumptions 2.1 and 2.2 hold. Also suppose $K^{(0)}$ is a stabilizer for the system $[A, C; B, D]$. Then all

the policies $\{K^{(i)}\}_{i=1}^{\infty}$ updated by (8) are stabilizers. Moreover, the solution $P^{(i+1)} \in \mathcal{S}_{++}^n$ in Algorithm 1 is unique at each step.

Proof Because $K^{(0)}$ is a stabilizer for the system $[A, C; B, D]$, by the same argument in the proof of Lemma 2.3, there exists a unique solution $P^{(i+1)} \in \mathcal{S}_{++}^n$ of Lyapunov Recursion (9) with $i = 0$.

We prove by mathematical induction. Suppose $i \geq 1$, $K^{(i-1)}$ is a stabilizer and $P^{(i)} \in \mathcal{S}_{++}^n$ is the unique solution of Lyapunov Recursion (9). We now show $K^{(i)} = -(R+D^\top P^{(i)} D)^{-1}(B^\top P^{(i)} + D^\top P^{(i)} C + S)$ is also a stabilizer and $P^{(i+1)} \in \mathcal{S}_{++}^n$. To this end, we first notice

$$\begin{aligned}
&(A+BK^{(i)})^\top P^{(i)} + P^{(i)}(A+BK^{(i)}) \\
&\quad + (C+DK^{(i)})^\top P^{(i)}(C+DK^{(i)}) \\
&= (A+BK^{(i-1)})^\top P^{(i)} + P^{(i)}(A+BK^{(i-1)}) \\
&\quad + (C+DK^{(i-1)})^\top P^{(i)}(C+DK^{(i-1)}) \\
&\quad - \Big[(K^{(i-1)}-K^{(i)})^\top B^\top P^{(i)} + P^{(i)} B(K^{(i-1)}-K^{(i)}) \\
&\quad + (C+DK^{(i-1)})^\top P^{(i)}(C+DK^{(i-1)}) \\
&\quad - (C+DK^{(i)})^\top P^{(i)}(C+DK^{(i)})\Big] \\
&= -\big[K^{(i-1)\top} R K^{(i-1)} + S^\top K^{(i-1)} + K^{(i-1)\top} S + Q\big] \\
&\quad - (K^{(i-1)}-K^{(i)})^\top D^\top P^{(i)} D (K^{(i-1)}-K^{(i)}) \\
&\quad - \Big[(K^{(i-1)}-K^{(i)})^\top \big[B^\top P^{(i)} + D^\top P^{(i)}(C+DK^{(i)})\big] \\
&\quad + \big[P^{(i)}B + (C+DK^{(i)})^\top P^{(i)} D\big]^\top (K^{(i-1)}-K^{(i)})\Big].
\end{aligned} \tag{14}$$

From Policy Improvement (8), $B^\top P^{(i)} + D^\top P^{(i)} C = -(R + D^\top P^{(i)} D)K^{(i)} - S$. Plugging this into (14) and using $Q - S^\top R^{-1} S > 0$, we obtain

$$\begin{aligned}
&(A+BK^{(i)})^\top P^{(i)} + P^{(i)}(A+BK^{(i)}) \\
&\quad + (C+DK^{(i)})^\top P^{(i)}(C+DK^{(i)}) \\
&= -\big[Q - S^\top R^{-1} S + (K^{(i)} + R^{-1}S)^\top R (K^{(i)} + R^{-1}S)\big] \\
&\quad - (K^{(i-1)}-K^{(i)})^\top (R + D^\top P^{(i)} D)(K^{(i-1)}-K^{(i)}) \\
&< 0.
\end{aligned}$$

So $K^{(i)}$ is a stabilizer by Lemma 2.1. Moreover, Lyapunov Recursion (9) admits a unique solution $P^{(i+1)} \in \mathcal{S}_{++}^n$ since

$$\begin{aligned}
&K^{(i)\top} R K^{(i)} + S^\top K^{(i)} + K^{(i)\top} S + Q \\
&\quad = Q - S^\top R^{-1} S + (RK^{(i)} + S)^\top R^{-1} (RK^{(i)} + S) > 0.
\end{aligned}$$

From Lemma 2.3, Lyapunov Recursion (9) is equivalent to Policy Evaluation (7), so $P^{(i+1)} \in \mathcal{S}_{++}^n$ is the unique solution in Algorithm 1. □

Now, we prove the convergence of Algorithm 1.

Theorem 2.2: The iteration $\{P^{(i)}\}_{i=1}^{\infty}$ of Algorithm 1 converges to the unique solution $P \in \mathcal{S}_{++}^n$ of the following SARE

$$\begin{aligned}
&A^\top P + PA^\top + C^\top PC + Q \\
&\quad - (PB + C^\top PD + S^\top)(R + D^\top PD)^{-1} \\
&\quad \times (B^\top P + D^\top PC + S) = 0.
\end{aligned} \tag{15}$$

Also, the unique optimal control of Problem (SLQ) is

$$u^* = -(R + D^\top PD)^{-1}(B^\top P + D^\top PC + S)X^*, \tag{16}$$

where $X^*(\cdot)$ is determined by system (2) with

$$K = -(R + D^\top PD)^{-1}(B^\top P + D^\top PC + S). \tag{17}$$

Moreover, K is a stabilizer of the system $[A, C; B, D]$.

Proof From Lemma 2.3, Algorithm 1 is equivalent to Lyapunov Recursion (9) with Policy Improvement (8). We now prove $\{P^{(i)}\}_{i=1}^{\infty}$ in (9) combining with (8) converges to the solution P of SARE (15). Note $P^{(i+1)}$ satisfies Lyapunov Recursion (9). Denote $\Delta P^{(i+1)} = P^{(i)} - P^{(i+1)}$, and $\Delta K^{(i)} = K^{(i-1)} - K^{(i)}$ for $i = 1, 2, \cdots$, then

$$\begin{aligned}
0 &= (A+BK^{(i-1)})^\top P^{(i)} + P^{(i)}(A+BK^{(i-1)}) \\
&\quad + (C+DK^{(i-1)})^\top P^{(i)}(C+DK^{(i-1)}) \\
&\quad + K^{(i-1)\top} R K^{(i-1)} + S^\top K^{(i-1)} + K^{(i-1)\top} S \\
&\quad - \big[(A+BK^{(i)})^\top P^{(i+1)} + P^{(i+1)}(A+BK^{(i)}) \\
&\quad + (C+DK^{(i)})^\top P^{(i+1)}(C+DK^{(i)}) \\
&\quad + K^{(i)\top} R K^{(i)} + S^\top K^{(i)} + K^{(i)\top} S\big] \\
&= (A+BK^{(i)})^\top \Delta P^{(i+1)} + \Delta P^{(i+1)}(A+BK^{(i)}) \\
&\quad + (C+DK^{(i)})^\top \Delta P^{(i+1)}(C+DK^{(i)}) \\
&\quad + \Delta K^{(i-1)\top} B^\top P^{(i)} + P^{(i)} B \Delta K^{(i-1)} \\
&\quad + (C+DK^{(i-1)})^\top P^{(i)}(C+DK^{(i-1)}) \\
&\quad - (C+DK^{(i)})^\top P^{(i)}(C+DK^{(i)}) \\
&\quad + K^{(i-1)\top} R K^{(i-1)} - K^{(i)\top} R K^{(i)} \\
&\quad + S^\top \Delta K^{(i)} + \Delta K^{(i)} S.
\end{aligned} \tag{18}$$

It follows from Policy Improvement (8) that we have

$$\begin{aligned}
&(C+DK^{(i-1)})^\top P^{(i)}(C+DK^{(i-1)}) \\
&\quad - (C+DK^{(i)})^\top P^{(i)}(C+DK^{(i)}) \\
&= \Delta K^{(i)\top} D^\top P^{(i)} D \Delta K^{(i)} \\
&\quad + \Delta K^{(i)\top} D^\top P^{(i)}(C+DK^{(i)}) \\
&\quad + (C+DK^{(i)})^\top P^{(i)} D \Delta K^{(i)}.
\end{aligned} \tag{19}$$

Note

$$\begin{aligned}
K^{(i-1)\top} R K^{(i-1)} - K^{(i)\top} R K^{(i)} &= \Delta K^{(i)\top} R \Delta K^{(i)} \\
&\quad + \Delta K^{(i)\top} R K^{(i)} + K^{(i)\top} R \Delta K^{(i)}.
\end{aligned} \tag{20}$$

Combining (18)-(20), we deduce

$$\begin{aligned}
&(A+BK^{(i)})^\top \Delta P^{(i+1)} + \Delta P^{(i+1)}(A+BK^{(i)}) \\
&\quad + (C+DK^{(i)})^\top \Delta P^{(i+1)}(C+DK^{(i)}) \\
&\quad + \Delta K^{(i)\top}(R + D^\top P^{(i)} D)\Delta K^{(i)} \\
&\quad + \Delta K^{(i)\top}\big[B^\top P^{(i)} + D^\top P^{(i)} C + S + (R + D^\top P^{(i)} D)K^{(i)}\big] \\
&\quad + \big[B^\top P^{(i)} + D^\top P^{(i)} C + S + (R + D^\top P^{(i)} D)K^{(i)}\big]^\top \Delta K^{(i)} \\
&= 0.
\end{aligned}$$

By (8), we have $-(R+D^\top P^{(i)} D)K^{(i)} = B^\top P^{(i)} + D^\top P^{(i)} C + S$, so

$$\begin{aligned}
&(A+BK^{(i)})^\top \Delta P^{(i+1)} + \Delta P^{(i+1)}(A+BK^{(i)}) \\
&\quad + (C+DK^{(i)})^\top \Delta P^{(i+1)}(C+DK^{(i)}) \\
&\quad + \Delta K^{(i)\top}(R + D^\top P^{(i)} D)\Delta K^{(i)} = 0.
\end{aligned} \tag{21}$$

Since $K^{(i)}$ is a stabilizer of the system (1) and $\Delta K^{(i)\top}(R + D^\top P^{(i)} D)\Delta K^{(i)} \geq 0$, Lyapunov equation (21) admits a unique solution $\Delta P^{(i+1)} \geq 0$ by Lemma 2.1. Therefore, $\{P^{(i)}\}_{i=1}^{\infty}$ is monotonically decreasing. Notice $P^{(i)} > 0$, so $\{P^{(i)}\}_{i=1}^{\infty}$ converges to some $P \geq 0$.

Next, we prove that P is the solution of SARE (15). When $i \to \infty$,

$$\begin{aligned}
&(R + D^\top P^{(i)} D)^{-1}(B^\top P^{(i)} + D^\top P^{(i)} C + S) \\
&\quad \to (R + D^\top PD)^{-1}(B^\top P + D^\top PC + S),
\end{aligned}$$

which means that $\{K^{(i)}\}_{i=1}^{\infty}$ converges to K given by (17). Moreover, (P, K) satisfies

$$\begin{aligned}(A+BK)^\top P + P(A+BK) + (C+DK)^\top P(C+DK) \\ + K^\top RK + S^\top K + KS + Q = 0.\end{aligned} \tag{22}$$

Since $K^\top RK + S^\top K + KS + Q > 0$, we get

$$(A+BK)^\top P + P(A+BK) + (C+DK)^\top P(C+DK) < 0,$$

which implies that K is a stabilizer of the system (1). By (22) and Lemma 2.1, $P \in \mathcal{S}_{++}^n$ is the unique solution of (22). Moreover, when plugging (17) into (22), (22) becomes SARE (15). From Theorem 13 in [1], we see that (16) is the unique optimal control. □

III. Online Implementation of Partially Model-Free RL Algorithm

A. Online Implementation

In this section, we illustrate the implementation of Algorithm 1 in detail. Since there are $N := \frac{n(n+1)}{2}$ independent parameters in the positive definite matrix $P^{(i+1)}$, we need to observe the state along trajectories at least N intervals $[t_j, t_j + \Delta t_j]$ on $[0, \infty)$ to reinforce the target function $\Delta J^{(i)}(t_j, t_j + \Delta t_j; X^{(i)}, K^{(i)})$ defined by (13) with $j = 1, 2, \cdots, N$. From Policy Evaluation (7) in Algorithm 1, for initial state x_{t_j} at time t_j, one needs to solve a set of simultaneous equations

$$\begin{aligned}&x_{t_j}^\top P^{(i+1)} x_{t_j} - \mathbb{E}^{\mathcal{F}_{t_j}}\left[X^{(i)}(t_j+\Delta t_j)^\top P^{(i+1)} X^{(i)}(t_j+\Delta t_j)\right] \\ &= \Delta J^{(i)}(t_j, t_j+\Delta t_j; X^{(i)}, K^{(i)})\end{aligned} \tag{23}$$

with $j = 1, 2, \cdots, N$ at each iteration i. Sometimes, we suppress $X^{(i)}$ and $K^{(i)}$ in target function (13) to avoid heavy notation.

We will use vectorization and Kronecker product theory to solve the above system (23); see [20] for details. Define $\mathrm{vec}(M)$ for $M \in \mathbb{R}^{n\times m}$ as a vectorization map from a matrix into an nm-dimensional column vector for compact representations, which stacks the columns of M on top of one another. For example,

$$\mathrm{vec}\left(\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \\ a_{31} & a_{32}\end{bmatrix}\right) = (a_{11}, a_{21}, a_{31}, a_{12}, a_{22}, a_{32})^\top.$$

Let $A \otimes B$ be a Kronecker product of matrices A and B, then we have $\mathrm{vec}(ABC) = (C^\top \otimes A)\mathrm{vec}(B)$. Denote

$$\Delta X_j^{(i)} = x_{t_j}^\top \otimes x_{t_j}^\top - \mathbb{E}^{\mathcal{F}_{t_j}}[X^{(i)}(t_j+\Delta t_j)^\top \otimes X^{(i)}(t_j+\Delta t_j)^\top],$$

the set of equations (23) is transformed to

$$\begin{bmatrix} \Delta X_1^{(i)} \\ \Delta X_2^{(i)} \\ \vdots \\ \Delta X_N^{(i)} \end{bmatrix} \mathrm{vec}(P^{(i+1)}) = \begin{bmatrix} \Delta J^{(i)}(t_1, t_1+\Delta t) \\ \Delta J^{(i)}(t_2, t_2+\Delta t_2) \\ \vdots \\ \Delta J^{(i)}(t_N, t_N+\Delta t_N) \end{bmatrix}. \tag{24}$$

Denote

$$\mathbb{X}^{(i)} = \begin{bmatrix} \Delta X_1^{(i)} \\ \Delta X_2^{(i)} \\ \vdots \\ \Delta X_N^{(i)} \end{bmatrix}, \quad \mathbb{J}^{(i)} = \begin{bmatrix} \Delta J^{(i)}(t_1, t_1+\Delta t) \\ \Delta J^{(i)}(t_2, t_2+\Delta t) \\ \vdots \\ \Delta J^{(i)}(t_N, t_N+\Delta t_N) \end{bmatrix},$$

then (24) is rewritten as

$$\mathbb{X}^{(i)}\mathrm{vec}(P^{(i+1)}) = \mathbb{J}^{(i)}. \tag{25}$$

In practice, we derive the expectation $\mathbb{E}^{\mathcal{F}_{t_j}}[\cdot]$ in $\Delta X_j^{(i)}$ by calculating the mean-value based on $\mathcal{K}$ sample paths X_k, $k = 1, 2, \cdots, \mathcal{K}$. Precisely, we calculate

$$\begin{aligned}&\mathbb{E}^{\mathcal{F}_{t_j}}[X^{(i)}(t_j+\Delta t_j)^\top \otimes X^{(i)}(t_j+\Delta t_j)^\top] \\ &\approx \frac{1}{\mathcal{K}}\sum_{k=1}^{\mathcal{K}}[X_k^{(i)}(t_j+\Delta t_j)^\top \otimes X_k^{(i)}(t_j+\Delta t_j)^\top]\end{aligned}$$

by the sampled data at terminal time $t_j + \Delta t_j$. In (25), if we get $\mathcal{K}$ sample paths with the data sampled at time t_l $(t_j \le t_l \le t_j + \Delta t_j, l = 1, 2, \cdots, L)$, we calculate $\Delta J^{(i)}$ in $\mathbb{J}^{(i)}$ as

$$\begin{aligned}&\Delta J^{(i)}(t_j, t_j+\Delta t_j) \\ &\approx \frac{1}{\mathcal{K}}\sum_{k=1}^{\mathcal{K}}\Big[\sum_{l=1}^{L} X_k^{(i)}(t_l)^\top \left[Q + 2K^{(i)\top}S + K^{(i)\top}RK^{(i)}\right] X_k^{(i)}(t_l)\Big].\end{aligned}$$

Moreover, we define an operator $\mathrm{vec}^+(P)$ for $P \in \mathcal{S}^n$, which maps P into an N-dimensional vector by stacking the columns corresponding to the diagonal and lower triangular parts of P on top of one another where the off-diagonal terms of P are double. For example,

$$\mathrm{vec}^+\left(\begin{bmatrix} p_{11} & p_{12} & p_{13} \\ p_{12} & p_{22} & p_{23} \\ p_{13} & p_{23} & p_{33}\end{bmatrix}\right) = (p_{11}, 2p_{12}, 2p_{13}, p_{22}, 2p_{23}, p_{33})^\top.$$

Similar to [20], there exists a matrix $\mathcal{T} \in \mathbb{R}^{n^2\times N}$ with $\mathrm{rank}(\mathcal{T}) = N$ such that $\mathrm{vec}(P) = \mathcal{T}\mathrm{vec}^+(P)$ for any $P \in \mathcal{S}^n$. Then equation (25) becomes

$$(\mathbb{X}^{(i)}\mathcal{T})\mathrm{vec}^+(P^{(i+1)}) = \mathbb{J}^{(i)}. \tag{26}$$

To get $\mathrm{vec}^+(P^{(i+1)})$ from (26), one must chose enough trajectories $X^{(i)}$ on intervals $[t_j, t_j + \Delta t_j]$ with $j = 1, 2, \cdots, N$ such that

$$\mathrm{vec}^+(P^{(i+1)}) = (\mathbb{X}^{(i)}\mathcal{T})^{-1}\mathbb{J}^{(i)}. \tag{27}$$

Finally, we obtain $P^{(i+1)}$ by taking the inverse map of $\mathrm{vec}^+(\cdot)$.

B. Numerical Examples

This section presents two numerical examples with dimensions 2 and 5, respectively. Firstly, let $n = 2$ and $m = 1$; set

$$A = \begin{bmatrix} 0.3 & 0.7 \\ -0.9 & 0.5\end{bmatrix}, \ B = \begin{bmatrix} 0.2 \\ 0\end{bmatrix}, \ C = \begin{bmatrix} 0.05 & 0.03 \\ 0.05 & 0.02\end{bmatrix}, \ D = \begin{bmatrix} 0.05 \\ 0.06\end{bmatrix},$$

and $x = (2, 3)^\top$. The coefficients in cost functional are chosen as

$$Q = \begin{bmatrix} 3 & 0 \\ 0 & 2\end{bmatrix}, \quad S = \mathbf{O}, \quad R = 1.25.$$

Algorithm 1 is implemented without all information about system at the initial time. Therefore, we randomly choose $K^{(0)}$ to run the system and observe the state $X(s)$ in the numerical examples. If there is a $K^{(0)}$ such that $X(s)$ tends to a neighborhood of zero as time s goes to infinity, then $K^{(0)}$ can be chosen as an initial stabilizer. The initial stabilizer is selected as $K^{(0)} = (-6, 3)$, which makes the corresponding state trajectory $X = (X_1, X_2)^\top$ be stabilizable, see Fig. 1 (a) for detail. Because there are $N = \frac{n(n+1)}{2} = 3$ independent parameters of P in this example, we select 3 intervals $[0, 1], [1, 2]$ and $[2, 3]$ to reinforce the target function $\Delta J^{(i)}(t_j, t_j + \Delta t_j)$ $(t_j = 0, 1, 2, \Delta t_j = 1)$ defined by (13). Implementing Algorithm 1, we calculate $P^{(i+1)}$ by (27) and obtain

$$P^* = \begin{bmatrix} 61.1422 & -35.7578 \\ -35.7578 & 81.6610\end{bmatrix}$$

after 10 iterations in 5 seconds, please see details in Fig. 1 (b).

We denote the left side of SARE (15) as

$$\mathcal{R}(P) = A^\top P + PA^\top + C^\top PC + Q - (PB + C^\top PD + S^\top)(R + D^\top PD)^{-1}(B^\top P + D^\top PC + S). \tag{28}$$

It is used to measure the distance from P^* to the real solution of SARE. Insert P^* in to (28), we have

$$\mathcal{R}(P^*) = 10^{-4} \cdot \begin{bmatrix} 0.6829 & 0.1212 \\ 0.1212 & -0.7346 \end{bmatrix}$$

and $\|\mathcal{R}(P^*)\| = 1.0175 \times 10^{-4}$. Then the optimal control is $u^* = K^*X^* = -(R + D^\top P^* D)^{-1}(B^\top P^* + D^\top P^* C)X^* = (-8.3854, 4.7642)X^*$. The optimal trajectory $X^* = (X_1^*, X_2^*)^\top$ is presented in Fig. 1 (c).

Now, we compare our method with a model-based approach, which involves two steps: (1) obtain an estimation $\widehat{A}$ of A; (2) solve SARE (15) with $\widehat{A}$ directly by the SDP method in [1]. We use the least-square method to approximate A by the trajectory data $\{x_k\}_{k=0}^{n^2}$ on the time interval $[0, n^2]$ with the following estimation procedure.

1) Select $K = -(D^\top D)^{-1}D^\top C$;
2) Read the data $\{x_k\}_{k=0}^{n^2}$ at time $t_k = \frac{k}{n^2}, k = 0, 1, 2, 3, \cdots, n^2$;
3) Define $y_k = (x_{k+1} - x_k)/(t_{k+1} - t_k)$, and note that $Y = (y_0, \cdots, y_{n^2-1}), \quad X = (x_0, \cdots, x_{n^2-1})$;
4) Estimate $\widehat{A} = YX^\top(XX^\top)^{-1} - BK$.

By the above procedure, we obtain an approximation of A as

$$\widehat{A} = \begin{bmatrix} 0.2987 & 0.7009 \\ -0.9014 & 0.4993 \end{bmatrix}$$

after 11 iterations (shown in Fig. 1 (d)) and use the SDP method in [1] to obtain

$$\widehat{P}^* = \begin{bmatrix} 60.9679 & -35.5549 \\ -35.5549 & 81.2020 \end{bmatrix}$$

in 12 seconds totally. Inserting $\widehat{P}^*$ in to (28), we calculate

$$\mathcal{R}(\widehat{P}^*) = \begin{bmatrix} 0.0642 & -0.0189 \\ -0.0189 & 0.1886 \end{bmatrix}$$

and $\|\mathcal{R}(\widehat{P}^*)\| = 0.1915$. Comparing the proposed partially model-free method in this paper to the above model-based method, the former is more effective than the latter in time consumption and accuracy.

Next, we consider an example with $n = 5$ and $m = 2$, whose coefficient is selected from the example in [1] with $R = I$. To save space, we do not present the parameters of the problem, please see details in [1]. Firstly, we try to find an initial stabilizer by observing the state: running the system with

$$K^{(0)} = \begin{bmatrix} -1.6 & 0.6 & 0.6 & 2 & -1.8 \\ 0.8 & 2.4 & 2.6 & -1.2 & -1.4 \end{bmatrix},$$

we see that the state trajectory $X = (X_1, X_2, X_3, X_4, X_5)^\top$ tends to zero when s grows, see Fig. 1 (e). Then $K^{(0)}$ is chosen as the initial stabilizer. By Algorithm 1, we get

$$P^* = \begin{bmatrix} 0.3684 & 0.3093 & 0.2112 & 0.0272 & -0.3673 \\ 0.3093 & 1.3394 & 1.2835 & -1.0029 & -0.7577 \\ 0.2112 & 1.2835 & 1.6841 & -1.2102 & -0.7843 \\ 0.0272 & -1.0029 & -1.2102 & 2.0572 & 0.0801 \\ -0.3673 & -0.7577 & -0.7843 & 0.0801 & 1.6469 \end{bmatrix}$$

after 10 iterations in 9 seconds. Inserting P^* in to (28), we calculate

$$\mathcal{R}(P^*) = 10^{-3} \cdot \begin{bmatrix} 0.0007 & -0.0579 & -0.0460 & 0.0232 & 0.0110 \\ -0.0579 & 0.1095 & -0.0778 & -0.0024 & -0.0133 \\ -0.0460 & -0.0778 & 0.0031 & 0.0458 & -0.0193 \\ 0.0232 & -0.0024 & 0.0458 & -0.0359 & 0.0118 \\ 0.0110 & -0.0133 & -0.0193 & 0.0118 & 0.0044 \end{bmatrix}$$

and $\|\mathcal{R}(P^*)\| = 1.6091 \times 10^{-4}$. Comparing to the model-based approch, we obtain the estimation of A after 12 iterations as

$$\widehat{A} = \begin{bmatrix} -0.7617 & 0.2551 & -0.9179 & -0.1719 & -0.7017 \\ 1.4727 & -0.6803 & 2.7187 & -0.6058 & -0.8224 \\ 0.8520 & 0.8113 & -1.7078 & -1.5577 & 1.1287 \\ 0.7212 & -0.1878 & 0.0684 & -0.3889 & -0.2338 \\ 0.6363 & -1.0326 & -1.3784 & 0.6573 & -0.2375 \end{bmatrix}$$

and then compute $\widehat{P}^*$ in SARE by SDP

$$\widehat{P}^* = \begin{bmatrix} 0.3675 & 0.3086 & 0.2104 & 0.0271 & -0.3653 \\ 0.3086 & 1.3340 & 1.2753 & -0.9988 & -0.7526 \\ 0.2104 & 1.2753 & 1.6732 & -1.2044 & -0.7787 \\ 0.0271 & -0.9988 & -1.2044 & 2.0505 & 0.0809 \\ -0.3653 & -0.7526 & -0.7787 & 0.0809 & 1.6360 \end{bmatrix}.$$

in 16 seconds totally. Then, we obtain

$$\mathcal{R}(\widehat{P}^*) = \begin{bmatrix} 0.0000 & 0.0001 & 0.0003 & 0.0010 & 0.0002 \\ 0.0001 & 0.0013 & 0.0076 & -0.0007 & -0.0034 \\ 0.0003 & 0.0076 & 0.0095 & -0.0042 & -0.0029 \\ 0.0010 & -0.0007 & -0.0042 & 0.0022 & -0.0029 \\ 0.0002 & -0.0034 & -0.0029 & -0.0029 & 0.0094 \end{bmatrix}$$

and $\|\mathcal{R}(\widehat{P}^*)\| = 1.6883 \times 10^{-2}$. Comparatively, the accuracy of the model-free method is also higher than the model-based method in the 5-dimensional example, which shows that the proposed algorithm in this paper performs better. The optimal control is $u^* = K^*X^*$, where

$$K^* = \begin{bmatrix} -0.3354 & -0.1429 & -0.0372 & 0.1085 & -0.0427 \\ 0.4474 & 0.8542 & 0.9512 & -0.8883 & -0.3935 \end{bmatrix}.$$

The optimal state $X^* = (X_1^*, X_2^*, X_3^*, X_4^*, X_5^*)^\top$ is presented in Fig. 1 (f).

ACKNOWLEDGMENT

The authors would like to thank the associate editor and the anonymous referees for their constructive and insightful comments for improving the quality of this work.

REFERENCES

[1] M. Ait Rami and X. Y. Zhou, "Linear matrix inequalities, Riccati equations, and indefinite stochastic linear quadratic controls", *IEEE Trans. Automat. Contr.*, vol. 45, pp. 1131-1143, 2000.
[2] L. C. III Baird, "Reinforcement learning in continuous time: Advantage updating", *In Proc. of ICNN*, 1994.
[3] L. D. Berkovitz, *Optimal Control Theory*, Springer-Verlag, New York, 1974.
[4] S. J. Bradtke, B. E. Ydestie and A. G. Barto, "Adaptive linear quadratic control using policy iteration", *In: Proc. of ACC*, pp. 3475-3476, 1994.
[5] S. Chen, X. Li and X. Zhou, "Stochastic linear quadratic regulators with indefinite control weight costs", *SIAM J. Control Optimiz.*, vol. 36, pp. 1685-1702, 1998.
[6] X. Chen, G. Qu, Y. Tang, S. Low and N. Li, "Reinforcement learning for decision-making and control in power systems: Tutorial, review, and vision", *arXiv preprint arXiv:2102.01168v3*, 2021.
[7] T. Bian and Z. Jiang, "Reinforcement learning for linear continuous-time systems: an incremental learning approach", *IEEE-CAA J. Automatic.*, vol. 6, pp. 433-440, 2019.
[8] T. E. Duncan, L. Guo and B. Pasik-Duncan, "Adaptive continuous-time linear quadratic Gaussian control", *IEEE Trans. Automat. Contr.*, vol. 44, pp. 1653-1662, 1999.
[9] M. Fazel, R. Ge, S. M. Kakade, and M. Mesbahi, "Global convergence of policy gradient methods for the linear quadratic regulator", *In Proc. Intl Conf. Machine Learning*, pp. 1467-1476, 2018.
[10] W. H. Fleming and M. Nisio, "On stochastic relaxed control for partially observed diffusions", *Nagoya Mathematical Journal*, vol. 93, pp. 71-108, 1984.
[11] Y. Jiang and Z. Jiang, "Global adaptive dynamic programming for continuous-time nonlinear systems", *IEEE Trans. Automat. Contr.*, vol. 60, pp. 2917-2929, 2015.

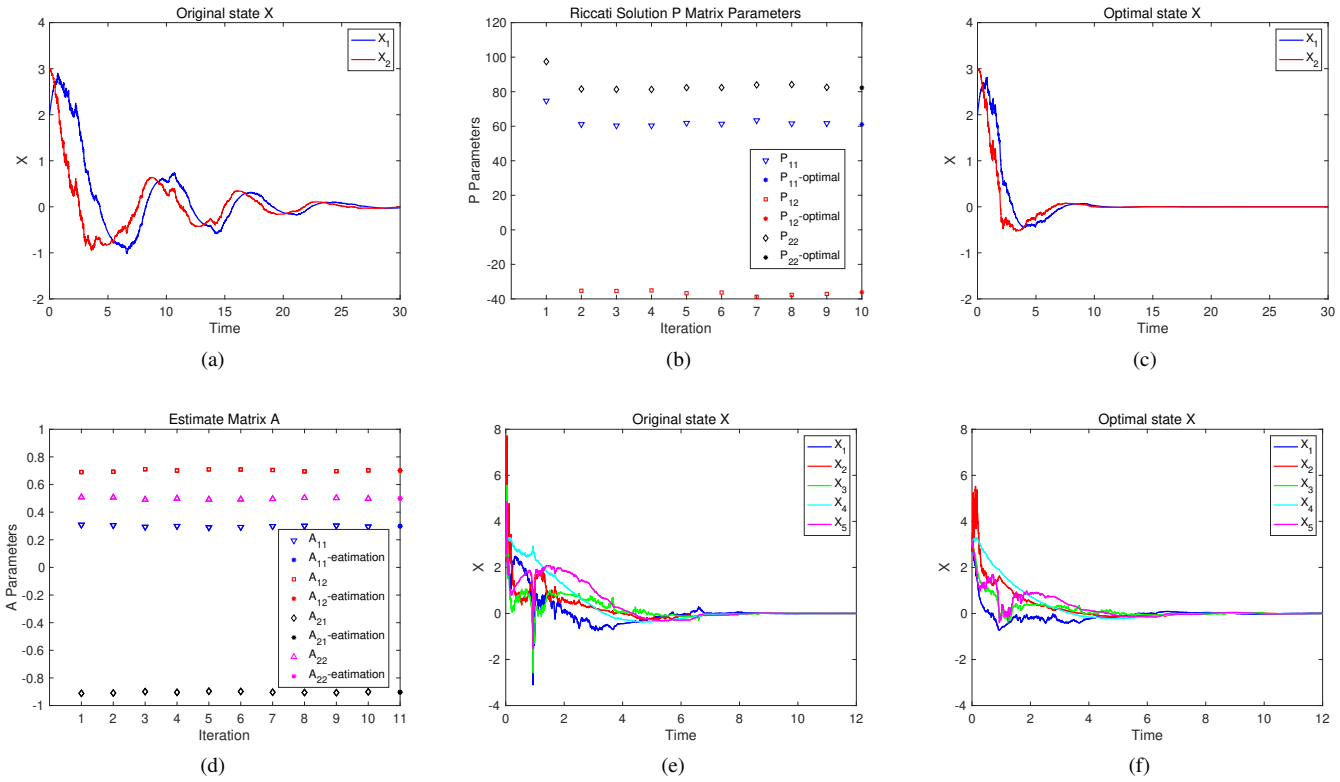

Fig. 1. Simulation results for solutions. (a): System state trajectory $\boldsymbol{X}$ running with the initial stabilizer $\boldsymbol{K}^{(0)}$ in 2-dimensional case; (b): Evolution of $\boldsymbol{P}^*$ parameters in 2-dimensional case; (c): The optimal state trajectory $\boldsymbol{X}^*$ with optimal control $\boldsymbol{u}^* = \boldsymbol{K}^*\boldsymbol{X}^*$ in 2-dimensional case; (d): Evolution of $\widehat{\boldsymbol{A}}$ parameters in 2-dimensional case; (e): System state trajectory $\boldsymbol{X}$ running with the initial stabilizer $\boldsymbol{K}^{(0)}$ in 5-dimensional case; (f): The optimal state trajectory $\boldsymbol{X}^*$ with optimal control $\boldsymbol{u}^* = \boldsymbol{K}^*\boldsymbol{X}^*$ in 5-dimensional case.

[12] B. Kiumarsi, K. G. Vamvoudakis, H. Modares, and F. L. Lewis, "Optimal and autonomous control using reinforcement learning: A survey", *IEEE T. Neur. Net. Lear.*, vol. 29, pp. 2042-2061, 2018.

[13] B. Kiumarsia, B. AlQaudi, H. Modaresa, F. L. Lewis and D. S. Levinee, "Optimal control using adaptive resonance theory and Q-learning", *Neurocomputing*, vol. 361, pp. 119-125, 2019.

[14] J. Lee and R. S. Sutton, "Policy iterations for reinforcement learning problems in continuous time and space-Fundamental theory and methods", *Automatica*, vol. 126, 109421, 2021.

[15] N. Li, X. Li and Z. Yu, "Indefinite mean-field type linear-quadratic stochastic optimal control problems", *Automatica*, vol. 122, 109267, 2020.

[16] B. Luo, D. Liu and H. Wu, "Adaptive constrained optimal control design for data-based nonlinear discrete-time systems with critic-only structure", *IEEE T. Neur. Net. Lear.*, vol. 29, pp. 2099-2111, 2018.

[17] H. Modares, F. L. Lewis and Z. Jiang, "Optimal output-feedback control of unknown continuous-time linear systems using off-policy reinforcement learning", *IEEE T. Cybernetics*, vol. 46, pp. 2401-2410, 2016.

[18] J. M. Mendel and R. W. McLaren, "Reinforcement learning control and pattern recognition systems". In J. M. Mendel, and K. S. Fu, *Adaptive, learning and pattern recognition systems: Theory and applications, New York: Academic Press*, vol. 66, pp. 287-318, 1970.

[19] H. Mohammadi, M. Soltanolkotabi and M. R. Jovanović, "Learning the model-free linear quadratic regulator via random search", *2nd Annual Conference on Learning for Dynamics and Control: Proceedings of Machine Learning Research*, vol. 120, pp. 1-9, 2020.

[20] J. J. Murray, C. J. Cox, G. G. Lendaris and R. Saeks, "Adaptive dynamic programming", *IEEE T. Syst. Man Cy-S*, vol. 32, pp. 140-153, 2002.

[21] K. S. Narendra and L. S. Valavani, "Direct and indirect adaptive control", *IFAC Proceedings Volumes*, vol. 11, pp. 1981-1987, 1978.

[22] S. A. A. Rizvi and Z. Lin, "Output feedback Q-learning control for the discrete-time linear quadratic regulator problem", *IEEE T. Neur. Net. Lear.*, vol. 30, pp. 1523-1536, 2019.

[23] J. Sun and J. Yong, "Stochastic linear quadratic optimal control problems in infinite horizon", *Appl. Math. Optim.*, vol. 78, pp. 145-183, 2018.

[24] R.S. Sutton, A.G. Barto, "Reinforcement learning: An introduction", *MIT Press, Cambridge*, Second Edition, 2018.

[25] R.S. Sutton, A.G. Barto and R.J. Williams, "Reinforcement learning is direct adaptive optimal control", *In: Proc. of ACC*, pp. 2143-2146, 1991.

[26] S. Tang, "Dynamic programming for general linear quadratic optimal stochastic control with random coefficients", *SIAM J. Control Optimiz.*, vol. 53, pp. 1082-1106, 2015.

[27] V. A. Ugrinovskii, "Robust H_∞ control in the presence of stochastic uncertainty", *Int. J. Contr.*, vol. 71, pp. 219-237, 1998.

[28] D. Vrabie, O. Pastravanu, M. Abu-Khalaf and F. L. Lewis, "Adaptive optimal control for continuous-time linear systems based on policy iteration", *Automatica*, vol. 45, pp. 477-484, 2009.

[29] H. Wang, T. Zariphopoulou and X. Y. Zhou, "Reinforcement learning in continuous time and space: A stochastic control approach", *Journal of Machine Learning Research*, vol. 21, pp. 1-34, 2020.

[30] H. Wang and X. Y. Zhou, "Continuous-time meanvariance portfolio selection: A reinforcement learning framework", *Mathematical Finance*, vol. 30, pp. 1273-1308, 2020.

[31] C. J. C. H. Watkins, "Learning from delayed rewards". *Ph.D. thesis. England: University of Cambridge*, 1989.

[32] P. Werbos, "Neural networks for control and system identification", *In: Proc. of CDC*, pp. 260-265, 1989.

[33] W. Ch. Wong and J. H. Lee, "A reinforcement learning-based scheme for direct adaptive optimal control of linear stochastic systems", *Optimal Control Applications and Methods*, vol. 31, pp. 365-374, 2010.

[34] F. A. Yaghmaie and D. J. Braun, "Reinforcement learning for a class of continuous-time input constrained optimal control problems", *Automatica*, vol. 99, pp. 221-227, 2019.

[35] J. Yong and X. Y. Zhou, *Stochastic controls: Hamiltonian systems and HJB equations*, Applications of Mathematics (New York), 43, Springer-Verlag, New York, 1999.

[36] X. Y. Zhou, "On the existence of optimal relaxed controls of stochastic partial differential equations", *SIAM J. Control Optimiz.*, vol. 30, pp. 247-261, 1992.