Model-Free Q-Learning for Infinite-Horizon Stochastic Linear Quadratic Problems with Regime Switching

Xinyue Zhang[*] Na Li[†] Xun Li[‡] Zuo Quan Xu[§]

July 30, 2026

Abstract

This paper addresses infinite-horizon continuous-time stochastic linear quadratic optimal control problems with regime switching. We propose a paradigm shift from model-based design by adopting an adaptive dynamic programming approach, specifically developing on-policy and off-policy Q-learning algorithms that learn the optimal controller solely from online state trajectory data. The theoretical core of our work consists of a complete proof of the equivalence between the on- and off-policy architectures, alongside a rigorous analysis establishing the stability of the closed-loop system and the convergence of the algorithms to the optimal solution. For computational tractability, we implement these algorithms using vectorization and Kronecker product algebra. The theoretical results are corroborated by numerical case studies that clearly demonstrate the operational effectiveness and practical feasibility of the proposed model-free control strategy.

Key words: Q-learning, Infinite-horizon, LQ problems with regime switching, Stochastic optimal control.

1 Introduction

Many practical control systems are subject to abrupt structural changes caused by random environmental fluctuations. Such phenomena are naturally captured by regime switching model, where the evolution of the system modes is governed by an underlying Markov chain. Owing to its ability to model complex, randomly switching dynamics, regime switching model has found widespread application across diverse engineering domains, including power systems [13], robotics [11], and wind energy conversion [3]. Beyond engineering, the framework has also proven instrumental in financial mathematics, for instance, Zhou and Yin [29] developed a continuous-time Markowitz model for portfolio optimization under regime-switching market conditions. The

[*]School of Statistics and Mathematics, Shandong University of Finance and Economics, Jinan 250014, China. Email: xinyuezhang1022@163.com.

[†]Corresponding author. School of Mathematical Sciences, Dalian University of Technology, Dalian 116024, China. Email: lina2025@dlut.edu.cn.

[‡]Department of Applied Mathematics, The Hong Kong Polytechnic University, Hong Kong SAR, China. Email: li.xun@polyu.edu.hk.

[§]Department of Applied Mathematics, The Hong Kong Polytechnic University, Hong Kong SAR, China. Email: maxu@polyu.edu.hk. The work of Na Li is supported in part by the National Natural Science Foundation of China (12571475 and 12171279). The work of Xun Li is supported in part by the Research Grants Council of Hong Kong (15225124), and in part by the PolyU (4-ZZVB). The work of Zuo Quan Xu is supported in part by the National Natural Science Foundation of China (12571517), in part by the Hong Kong RGC (15203423 and 15204622), in part by the PolyU-SDU Joint Research Center on Financial Mathematics, in part by the CAS AMSS-PolyU Joint Laboratory of Applied Mathematics, in part by the Research Centre for Quantitative Finance (1-CE03), and in part by the internal grants from The Hong Kong Polytechnic University.

study of regime switching model thus carries both profound theoretical significance and substantial practical utility, offering a rigorous foundation for tackling challenging stochastic control problems in complex environments.

Over the past two decades, the linear quadratic (LQ) control problem with regime switching has attracted considerable research attention, yielding a rich body of literature (e.g., [4,7,14,28]). The prevailing approach in this domain hinges on solving the coupled algebraic Riccati equations (CAREs) for a unique positive definite solution (e.g., [6, 8, 10, 16, 18, 19, 25]). To illustrate, Li et al. [19] established the equivalence between the well-posedness of indefinite stochastic LQ problems and the solvability of generalized CAREs. Recognizing the inherent nonlinearity and coupling of these equations, Gajic and Borno [8] developed an order-reduction iterative scheme that decouples algebraic Lyapunov equations, guaranteeing convergence of the solution sequence to the CARE solution. Subsequently, Ivanov [10] improved the scheme of [8] by addressing its reliance solely on information from the previous iteration, and proposed an accelerated algorithm that incorporates the most recently updated information within the same iteration. Building on both methods, Wu et al. [25] further advanced the state of the art by introducing an iterative update that simultaneously exploits both previous-step and current-step information. In summary, the vast majority of existing results for LQ problems with regime switching remain firmly rooted in CARE-based solutions, which inherently demand full knowledge of the system dynamics. Yet, in many practical environments, precise dynamic models are either unavailable or prohibitively expensive to obtain, rendering such model-dependent approaches infeasible.

Fortunately, reinforcement learning (RL) is capable of addressing LQ problems when information about the system dynamics is not readily available. Among RL approaches, Q-learning is a superior method that does not require any prior knowledge of the system dynamics. In recent years, Q-learning methods have been widely applied to solving LQ problems for both discrete-time [5, 24, 26] and continuous-time systems [17, 22, 23, 27]. On the one hand, in discrete-time cases, Rizvi and Lin [24] proposed an output feedback-based Q-learning algorithm, which enables the design of LQ controllers directly from input-output data and avoids excitation noise bias. For the situation with a single random parameter sample at each time step, Du et al. [5] adopted the Q-learning approach and proved both convergence and stability of the algorithm. For stochastic systems, Wang et al. [26] transformed the stochastic problem into deterministic one and employed a Q-learning method to solve LQ problems. On the other hand, in continuous-time cases, Palanisamy et al. [22] solved discounted LQ control problems for deterministic systems by employing a parameterized Q-function, while Yang et al. [27] focused on decentralized LQ control for stochastic systems using a decentralized Q-function. Possieri and Sassano [23] employed the Kleinman algorithm in a data-driven manner and proposed an Actor/Critic type Q-learning algorithm that does not require the existence of a known initial stabilizing feedback policy. Furthermore, Jia and Zhou [12] investigated q-learning under an entropy-regularized exploratory diffusion process framework in a stochastic case. Although Q-learning methods have made progress in solving LQ problems with the systems discussed above, research on employing Q-learning methods to solve LQ problems with regime switching remains relatively scarce. Therefore, further research into the theoretical analysis and algorithmic design of Q-learning for stochastic LQ optimal control problems with regime switching, particularly in complex scenarios where information about the system dynamics is unavailable, is of theoretical and practical importance.

Inspired by the aforementioned works, this paper employs a Q-learning method to address infinite-horizon continuous-time stochastic LQ optimal control problems with regime switching, where the drift and diffusion terms in the system dynamics depend on both the state and control. Two proposed algorithms, including on-policy and off-policy Q-learning algorithms, reduce the reliance on information about the system dynamics. Furthermore, we systematically prove the equivalence, stability, and convergence of two algorithms. In addition, the effectiveness and

feasibility of the developed algorithms are demonstrated through numerical experiments. The main contributions of this paper are summarized as follows:

(1) For infinite-horizon continuous-time stochastic LQ optimal control problems with regime switching, we propose on-policy and off-policy Q-learning algorithms. Unlike the work of Wu et al. [25], the proposed Q-learning algorithms only require state trajectory data, and do not require any prior knowledge of the system dynamics.

(2) Due to the uncertainty of Markov chains at next moment, it is impossible to know which Markov chain state will be in at moment $t+\Delta t$ ($t \geq 0$, $\Delta t > 0$). For this reason, we cannot identify which $P_j, j = 1, 2, ..., \mathbf{K}$ will be chosen at the future moment $t + \Delta t$. Meanwhile, all of the $\{P_j\}_{j=1}^{\mathbf{K}}$ are coupled together. By skillfully using system stability properties, we solve the coupled problem over the entire time interval $[t, +\infty)$, thereby significantly reducing computational complexity and decoupling $\{P_j\}_{j=1}^{\mathbf{K}}$.

(3) The stability and convergence of Lyapunov Iteration Scheme are first proved. Based on the equivalence between Q-learning algorithms and Lyapunov Iteration Scheme, we obtain the stability and convergence of the proposed on-policy and off-policy Q-learning algorithms. Compared to the coupling of $\{P_j\}_{j=1}^{\mathbf{K}}$ in Lyapunov Iteration Scheme, Q-learning algorithms can directly solve $\{P_j\}_{j=1}^{\mathbf{K}}$ without requiring the system coefficients.

The remainder of this paper is organized as follows: Section 2 introduces the notation and formulates the system model and the LQ problem with regime switching. Section 3 presents Lyapunov Iteration Scheme, on-policy and off-policy Q-learning algorithms, and proves the equivalence, stability and convergence of the proposed algorithms. Section 4 provides a numerical example to verify the effectiveness of the proposed algorithms. Finally, Section 5 concludes the paper.

2 Notation and Formulation

Let $\mathbb{R}^n$ denote the n-dimensional Euclidean space with Euclidean norm $|\cdot|$. Let $\mathbb{R}^{n\times r}$ be the set of all $(n \times r)$ real matrices. $M^\top$ denotes the transpose of a vector or matrix M. $I_n \in \mathbb{R}^{n\times n}$ denotes the n-dimensional identity matrix. We use $\mathcal{S}^n$, $\mathcal{S}^n_+$ and $\mathcal{S}^n_{++}$ to denote the collections of all symmetric matrices, positive semidefinite matrices, and positive definite matrices in $\mathbb{R}^{n\times n}$, respectively. Moreover, if a matrix $M \in \mathcal{S}^n_{++}$(resp. $M \in \mathcal{S}^n_+$) is positive definite (resp. positive semidefinite), we usually write $M > 0$ (resp. $M \geq 0$). Let $(\Omega, \mathcal{F}, \mathbb{F}, \mathbb{P})$ be a given filtered probability space on which there exists a standard one-dimensional Brownian motion $W(s)$ on $[0, \infty)$, and a continuous-time stationary Markov chain α_s taking values in a finite state space $\mathcal{M} = \{1, 2, \cdots, \mathbf{K}\}$, where $\mathbf{K} \in \mathbb{N}^+$ and $\mathbf{K} > 1$ with the generator $\Pi = (\pi_{kj})_{k,j\in\mathcal{M}}$. Assume that W and α are independent. For $s \geq 0$, we define $\mathcal{F}_s = \sigma\{W(\tau), \alpha_\tau : 0 \leq \tau \leq s\}$ augmented by all $\mathbb{P}$-null sets, and denote $\mathbb{F} = \{\mathcal{F}_s\}_{s\geq 0}$. For $t \geq 0$, we define $L^2_{\mathbb{F}}(t, \infty; \mathbb{R}^n)$ as the Hilbert space of all $\mathbb{R}^n$-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, we use $\otimes$ to denote the Kronecker product. For any matrix $P \in \mathcal{S}^n$, we define

$$\begin{aligned} vec(P) &= [p_{11}, p_{21}, \cdots, p_{n1}, p_{12}, p_{22}, \cdots, p_{n-1,n}, p_{nn}]^\top \in \mathbb{R}^{n^2}, \\ vech(P) &= [p_{11}, 2p_{12}, \cdots, 2p_{1n}, p_{22}, 2p_{23}, \cdots, 2p_{n-1,n}, p_{nn}]^\top \in \mathbb{R}^{\frac{1}{2}n(n+1)}, \end{aligned}$$

where p_{ij} $(i, j = 1, 2, 3, ...)$ is the (i, j)th element of P. According to [20], there exists a matrix $\mathcal{T} \in \mathbb{R}^{n^2\times\frac{1}{2}n(n+1)}$ with $\text{rank}(\mathcal{T}) = \frac{1}{2}n(n + 1)$ such that $vec(P) = \mathcal{T}\, vech(P)$ for any $P \in \mathcal{S}^n$.

Correspondingly, for any $H \in \mathbb{R}^{n\times r}$, we define

$$\bar{H} := \Big(H^\top \otimes H^\top\Big)\,\mathcal{T} \in \mathbb{R}^{r^2\times \frac{1}{2}n(n+1)},$$

which satisfies $(H^\top \otimes H^\top)\,vec(P) = \bar{H}\,vech(P)$. For a vector $x \in \mathbb{R}^n$, we define

$$\bar{x} = \big[x_1^2,\, x_1x_2,\, \ldots,\, x_1x_n,\, x_2^2,\, x_2x_3,\, \ldots,\, x_{n-1}x_n,\, x_n^2\big]^\top \in \mathbb{R}^{\frac{1}{2}n(n+1)},$$

where x_i $(i = 1, 2, 3, ...)$ is the ith element of x.

In this paper, we study a linear stochastic differential equation with regime switching

$$\begin{cases} dX(s) = [A(\alpha_s)X(s) + B(\alpha_s)u(s)]ds + [C(\alpha_s)X(s) + D(\alpha_s)u(s)]dW(s),\ s \in [t,\infty), \\ X(t) = x \in \mathbb{R}^n, \quad \alpha_t = k, \end{cases} \tag{1}$$

where $A(\alpha_s) = A_k$, $B(\alpha_s) = B_k$, $C(\alpha_s) = C_k$, $D(\alpha_s) = D_k$ when $\alpha_s = k$ $(k \in \mathcal{M})$. Here, the matrices A_k, etc. are given with appropriate dimensions. The Markov chain α_s has transition probabilities

$$\mathbf{P}\{\alpha_{s+\Delta s} = j \mid \alpha_s = k\} = \begin{cases} \pi_{kj}\Delta s + o(\Delta s), & \text{if } k \neq j, \\ 1 + \pi_{kk}\Delta s + o(\Delta s), & \text{if } k = j, \end{cases} \tag{2}$$

where $\pi_{kj} \geq 0$ for $k \neq j$ and $\pi_{kk} = -\sum_{j\neq k}\pi_{kj} \leq 0$.

The cost functional is given by

$$\begin{aligned} &J(t, x, k; u(\cdot)) \\ &= \mathbb{E}\Big\{\int_t^\infty \Big[X(s)^\top N(\alpha_s)X(s) + 2u(s)^\top S(\alpha_s)X(s) + u(s)^\top R(\alpha_s)u(s)\Big]ds \;\Big|\; X(t) = x, \alpha_t = k\Big\}, \end{aligned} \tag{3}$$

where $N(\alpha_s) = N_k$, $R(\alpha_s) = R_k$ and $S(\alpha_s) = S_k$ when $\alpha_s = k$ $(k \in \mathcal{M})$ with appropriate dimensions. From now on, we assume that the weighting matrices satisfy the positive definite condition: $R_k > 0$ and $N_k - S_k^\top R_k^{-1} S_k > 0$ for any $k \in \mathcal{M}$.

Definition 2.1. *A control $u(\cdot)$ is called mean-square stabilizing with respect to an initial (t, x, k) if the corresponding state $X(\cdot)$ in system (1) satisfies*

$$\lim_{s\to\infty} \mathbb{E}[X(s)^\top X(s)] = 0,$$

and system (1) is called mean-square stabilizable. Moreover, if the feedback control $u(s) = \sum_{k=1}^{\mathbf{K}} K_k X(s)\chi_{\{\alpha_s=k\}}(s)$ makes system (1) stabilize, then $(K_1, K_2, \cdots, K_{\mathbf{K}})$ is called a stabilizer of system (1).

When system (1) is mean-square stabilizable for a given $(t, x, k) \in [0,\infty)\times\mathbb{R}^n\times\mathcal{M}$, we define the corresponding set of admissible controls as

$$\mathcal{U}_{ad}(t, x, k) = \{u(\cdot) \in L^2_{\mathbb{F}}(t,\infty;\mathbb{R}^r) : u(\cdot) \text{ is stabilizing with respect to } (t, x, k)\}. \tag{4}$$

In this paper, we focus on the following problem, where Problem (LQ-RS) is the abbreviation of the LQ problem with regime switching.

Problem (LQ-RS). For a given initial $(t, x, k) \in [0,\infty)\times\mathbb{R}^n\times\mathcal{M}$, find a control $u^*(\cdot) \in \mathcal{U}_{ad}(t, x, k)$ such that

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

where $V(t,x,k)$ is called the value function of Problem (LQ-RS).

For a given $(t,x,k) \in [0,\infty) \times \mathbb{R}^n \times \mathcal{M}$, Problem (LQ-RS) is called well-posed if $V(t,x,k) > -\infty$. A well-posed problem is called *attainable* at (t,x,k) if there exists a control $u^*(\cdot)$ such that $J(t,x,k;u^*(\cdot)) = V(t,x,k)$. In this case, $u^*(\cdot)$ is called an *optimal control*, and $X^*(\cdot) \equiv X\big(\cdot;t,x,k,u^*(\cdot)\big)$ is called an *optimal state* corresponding to $u^*(\cdot)$.

Assumption 2.1. *System* (1) *is mean-square stabilizable.*

Next, we list three lemmas that are important in our subsequent analysis. First, we present the generalized Itô's formula from [21, Theorem 70].

Lemma 2.2. *Let $b(s,X,k)$ and $\sigma(s,X,k)$ be given $\mathbb{R}^n$-valued, $\mathcal{F}_s$-adapted process, $k \in \mathcal{M}$, and*

$$dX(s) = b(s,X(s),\alpha_s)ds + \sigma(s,X(s),\alpha_s)dW(s).$$

If $\varphi(\cdot,\cdot,k) \in C^{1,2}([0,\infty) \times \mathbb{R}^n)$ for all $k \in \mathcal{M}$, then we have

$$d\varphi(s,X(s),\alpha_s) = \mathcal{L}\varphi(s,X(s),\alpha_s)ds + \varphi_x^\top(s,X(s),\alpha_s)\sigma(s,X(s),\alpha_s)dW(s),$$

where

$$\begin{aligned}\mathcal{L}\varphi(s,x,k) :=\ & \varphi_t(s,x,k) + \varphi_x(s,x,k)^\top b(s,x,k) \\ & + \frac{1}{2}\mathrm{tr}[\sigma(s,x,k)^\top \varphi_{xx}(s,x,k)\sigma(s,x,k)] + \sum_{j=1}^{\mathbf{K}} \pi_{kj}\varphi(s,x,j).\end{aligned}$$

The following lemma illustrates that the stabilizability of system (1) is related to the feasibility of certain coupled Lyapunov inequalities, and not dependent on initial (t,x,k). Please refer to [9, Theorem 3.1] or [19, Lemma 2.5].

Lemma 2.3. *System* (1) *is mean-square stabilizable if and only if there exists matrices $K_1, K_2, \cdots, K_{\mathbf{K}}$ and symmetric matrices $P_1, P_2, \cdots, P_{\mathbf{K}} \in \mathcal{S}_{++}^n$ such that*

$$\begin{aligned}&(A_k + B_kK_k)^\top P_k + P_k(A_k + B_kK_k) \\ &\quad +(C_k + D_kK_k)^\top P_k(C_k + D_kK_k) + \sum_{j=1}^{\mathbf{K}} \pi_{kj}P_j < 0, \quad k \in \mathcal{M}.\end{aligned} \tag{5}$$

In this case, for any $\Lambda_1, \Lambda_2, \cdots, \Lambda_{\mathbf{K}} \in \mathcal{S}_+^n$ $(resp., \mathcal{S}_{++}^n)$, the Lyapunov eqautions

$$\begin{aligned}&(A_k + B_kK_k)^\top P_k + P_k(A_k + B_kK_k) \\ &\quad +(C_k + D_kK_k)^\top P_k(C_k + D_kK_k) + \sum_{j=1}^{\mathbf{K}} \pi_{kj}P_j + \Lambda_k = 0\end{aligned} \tag{6}$$

with $k \in \mathcal{M}$ admit a unique solution $(P_1, P_2, \cdots, P_{\mathbf{K}}) \in (\mathcal{S}_+^n)^{\mathbf{K}}$ $(resp., (\mathcal{S}_{++}^n)^{\mathbf{K}})$.

Referring to Theorems 4.4 and 5.2 in [19], we present the lemma for optimal control of Problem (LQ-RS) by coupled generalized algebraic Riccati equations (CGAREs).

Lemma 2.4. *If there exists a solution $(P_1, P_2, \cdots, P_{\mathbf{K}}) \in (\mathcal{S}_{++}^n)^{\mathbf{K}}$ to the following CGAREs*

$$\begin{aligned}&A_k^\top P_k + P_kA_k + C_k^\top P_kC_k + N_k + \sum_{j=1}^{\mathbf{K}} \pi_{kj}P_j \\ &\quad - (P_kB_k + C_k^\top P_kD_k + S_k^\top)(R_k + D_k^\top P_kD_k)^{-1}(B_k^\top P_k + D_k^\top P_kC_k + S_k) = 0, \quad k \in \mathcal{M},\end{aligned} \tag{7}$$

then the optimal control for Problem (LQ-RS) is

$$u^*(s) = K(\alpha_s)X^*(s) = \sum_{k=1}^{\mathbf{K}} K_k X^*(s)\chi_{\{\alpha_s=k\}}(s) \tag{8}$$

with $K_k = -(R_k + D_k^\top P_k D_k)^{-1}(B_k^\top P_k + D_k^\top P_k C_k + S_k)$*, where* X^* *is the corresponding optimal state with* u^* *of system* (1)*. The value function is*

$$\begin{aligned} &V(t,x,k) = x^\top P_k x \\ &= \mathbb{E}\Big\{ \int_t^\infty X^*(s)^\top \Big[N(\alpha_s) + 2K(\alpha_s)^\top S(\alpha_s) + K(\alpha_s)^\top R(\alpha_s)K(\alpha_s)\Big]X^*(s)ds \;\Big|\; X^*(t) = x, \alpha_t = k\Big\}, \end{aligned} \tag{9}$$

for $k \in \mathcal{M}$.

3 Q-learning method for Problem (LQ-RS)

In this section, we present the main results of this paper. We first introduce Lyapunov Iteration Scheme for Problem (LQ-RS) and prove its stability and convergence. Based on this scheme, we further develop on-policy and off-policy Q-learning algorithms, which can decouple $\{P_j\}_{j=1}^{\mathbf{K}}$ without the system coefficients.

3.1 Lyapunov Iteration Scheme

Now, we present **Lyapunov Iteration Scheme** as follows:

(1) **Initialization:** Select any given stabilizer $(K_1^{(0)}, K_2^{(0)}, \cdots, K_{\mathbf{K}}^{(0)})$ for system (1).

(2) **Lyapunov Recursion:** At each iteration step i $(i = 0, 1, 2, ...)$, solve $P_k^{(i+1)}$ $(k \in \mathcal{M})$ from the equations

$$\begin{aligned} &(A_k + B_k K_k^{(i)})^\top P_k^{(i+1)} + P_k^{(i+1)}(A_k + B_k K_k^{(i)}) + (C_k + D_k K_k^{(i)})^\top P_k^{(i+1)}(C_k + D_k K_k^{(i)}) \\ &\quad + \sum_{j=1}^{\mathbf{K}} \pi_{kj} P_j^{(i+1)} + K_k^{(i)\top} R_k K_k^{(i)} + S_k^\top K_k^{(i)} + K_k^{(i)\top} S_k + N_k = 0, \quad k \in \mathcal{M}. \end{aligned} \tag{10}$$

(3) **Policy Improvement:** Update $K_k^{(i+1)}$ $(k \in \mathcal{M})$ by

$$K_k^{(i+1)} = -(R_k + D_k^\top P_k^{(i+1)} D_k)^{-1}(B_k^\top P_k^{(i+1)} + D_k^\top P_k^{(i+1)} C_k + S_k), \quad k \in \mathcal{M}. \tag{11}$$

In Lyapunov Iteration Scheme, $\{(K_1^{(i)}, K_2^{(i)}, \cdots, K_{\mathbf{K}}^{(i)})\}_{i=0}^{\infty}$ should be the stabilizers of system (1) at each iteration, i.e., it is necessary to require that $(K_1^{(i)}, K_2^{(i)}, \cdots, K_{\mathbf{K}}^{(i)})$ is stepwise stable. Moreover, $\{(P_1^{(i)}, P_2^{(i)}, \cdots, P_{\mathbf{K}}^{(i)})\}_{i=1}^{\infty}$ should be convergent when $i \to \infty$. The following results confirm these.

Theorem 3.1. *Assume Assumption 2.1 holds. Let* $(K_1^{(0)}, K_2^{(0)}, \cdots, K_{\mathbf{K}}^{(0)})$ *be the known initial stabilizer for system* (1)*, then* $\{(K_1^{(i)}, K_2^{(i)}, \cdots, K_{\mathbf{K}}^{(i)})\}_{i=1}^{\infty}$ *updated by* (11) *are stabilizers. Moreover, the solution* $P_k^{(i+1)} \in \mathcal{S}_{++}^n$ $(k \in \mathcal{M})$ *of Lyapunov Recursion* (10) *is unique at each iteration step* i*.*

Proof. Since $K_k^{(0)}$ is a stabilizer for system (1) and

$$\begin{aligned}&K_k^{(0)\top}R_kK_k^{(0)}+S_k^\top K_k^{(0)}+K_k^{(0)\top}S_k+N_k\\&=N_k-S_k^\top R_k^{-1}S_k+(R_kK_k^{(0)}+S_k)^\top R_k^{-1}(R_kK_k^{(0)}+S_k)>0,\quad k\in\mathcal{M},\end{aligned}$$

from Lemma 2.3, there exists a unique solution $P_k^{(1)}\in\mathcal{S}_{++}^n$ of Lyapunov Recursion (10) with $i=0$.

Now, we prove that for each $i\geq 1$ and every $k\in\mathcal{M}$, if $K_k^{(i-1)}$ is a stabilizer and $P_k^{(i)}\in\mathcal{S}_{++}^n$ is the unique solution of Lyapunov Recursion (10), then $K_k^{(i)}=-(R_k+D_k^\top P_k^{(i)}D_k)^{-1}(B_k^\top P_k^{(i)}+D_k^\top P_k^{(i)}C_k+S_k)$ is also a stabilizer and $P_k^{(i+1)}\in\mathcal{S}_{++}^n$ for $k\in\mathcal{M}$. Indeed, a simple calculation shows that

$$\begin{aligned}&(A_k+B_kK_k^{(i)})^\top P_k^{(i)}+P_k^{(i)}(A_k+B_kK_k^{(i)})+(C_k+D_kK_k^{(i)})^\top P_k^{(i)}(C_k+D_kK_k^{(i)})+\sum_{j=1}^{\mathbf{K}}\pi_{kj}P_j^{(i)}\\&=(A_k+B_kK_k^{(i)})^\top P_k^{(i)}+P_k^{(i)}(A_k+B_kK_k^{(i)})+(C_k+D_kK_k^{(i)})^\top P_k^{(i)}(C_k+D_kK_k^{(i)})+\sum_{j=1}^{\mathbf{K}}\pi_{kj}P_j^{(i)}\\&\quad-\Big[(A_k+B_kK_k^{(i-1)})^\top P_k^{(i)}+P_k^{(i)}(A_k+B_kK_k^{(i-1)})+(C_k+D_kK_k^{(i-1)})^\top P_k^{(i)}(C_k+D_kK_k^{(i-1)})\\&\qquad+\sum_{j=1}^{\mathbf{K}}\pi_{kj}P_j^{(i)}+K_k^{(i-1)\top}R_kK_k^{(i-1)}+S_k^\top K_k^{(i-1)}+K_k^{(i-1)\top}S_k+N_k\Big]\\&=-\big[K_k^{(i-1)\top}R_kK_k^{(i-1)}+S_k^\top K_k^{(i-1)}+K_k^{(i-1)\top}S_k+N_k\big]\\&\quad-(K_k^{(i-1)}-K_k^{(i)})^\top D_k^\top P_k^{(i)}D_k(K_k^{(i-1)}-K_k^{(i)})\\&\quad-\Big[(K_k^{(i-1)}-K_k^{(i)})^\top\big[B_k^\top P_k^{(i)}+D_k^\top P_k^{(i)}(C_k+D_kK_k^{(i)})\big]\\&\quad+\big[P_k^{(i)}B_k+(C_k+D_kK_k^{(i)})^\top P_k^{(i)}D_k\big](K_k^{(i-1)}-K_k^{(i)})\Big],\quad k\in\mathcal{M}.\end{aligned}\tag{12}$$

From (11), we have

$$B_k^\top P_k^{(i)}+D_k^\top P_k^{(i)}C_k=-(R_k+D_k^\top P_k^{(i)}D_k)K_k^{(i)}-S_k,\quad k\in\mathcal{M}.\tag{13}$$

Plugging (13) into (12), since $N_k-S_k^\top R_k^{-1}S_k>0$, one gets

$$\begin{aligned}&(A_k+B_kK_k^{(i)})^\top P_k^{(i)}+P_k^{(i)}(A_k+B_kK_k^{(i)})+(C_k+D_kK_k^{(i)})^\top P_k^{(i)}(C_k+D_kK_k^{(i)})+\sum_{j=1}^{\mathbf{K}}\pi_{kj}P_j^{(i)}\\&=-\big[K_k^{(i)\top}R_kK_k^{(i)}+S_k^\top K_k^{(i)}+K_k^{(i)\top}S_k+N_k\big]\\&\quad-(K_k^{(i-1)}-K_k^{(i)})^\top(R_k+D_k^\top P_k^{(i)}D_k)(K_k^{(i-1)}-K_k^{(i)})\\&=-\big[N_k-S_k^\top R_k^{-1}S_k+(R_kK_k^{(i)}+S_k)^\top R_k^{-1}(R_kK_k^{(i)}+S_k)\big]\\&\quad-(K_k^{(i-1)}-K_k^{(i)})^\top(R_k+D_k^\top P_k^{(i)}D_k)(K_k^{(i-1)}-K_k^{(i)})\\&<0,\qquad\qquad k\in\mathcal{M},\end{aligned}\tag{14}$$

so $(K_1^{(i)},K_2^{(i)},\cdots,K_{\mathbf{K}}^{(i)})$ is a stabilizer by Lemma 2.3. Moreover, Lyapunov Recursion (10) admits a unique solution $P_k^{(i+1)}\in\mathcal{S}_{++}^n$ as $K_k^{(i)\top}R_kK_k^{(i)}+S_k^\top K_k^{(i)}+K_k^{(i)\top}S_k+N_k>0,\ k\in\mathcal{M}$. We obtain the conclusion. □

Theorem 3.2. *The iteration* $\{(P_1^{(i)}, P_2^{(i)}, \cdots, P_{\mathbf{K}}^{(i)})\}_{i=1}^{\infty}$ *of Lyapunov Recursion* (10) *converges to the solution* $(P_1, P_2, \cdots, P_{\mathbf{K}}) \in (\mathcal{S}_{++}^n)^{\mathbf{K}}$ *of CGARE*s (7).

Proof. Note $P_k^{(i+1)}$ satisfies Lyapunov Recursion (10). Denote

$$\Delta P_k^{(i+1)} = P_k^{(i)} - P_k^{(i+1)}, \quad \Delta K_k^{(i)} = K_k^{(i-1)} - K_k^{(i)}, \quad k \in \mathcal{M},$$

for $i = 1, 2, \cdots$, and we have

$$\begin{aligned}
0 &= (A_k + B_k K_k^{(i-1)})^\top P_k^{(i)} + P_k^{(i)}(A_k + B_k K_k^{(i-1)}) + (C_k + D_k K_k^{(i-1)})^\top P_k^{(i)}(C_k + D_k K_k^{(i-1)}) \\
&\quad + \sum_{j=1}^{\mathbf{K}} \pi_{kj} P_j^{(i)} + K_k^{(i-1)\top} R_k K_k^{(i-1)} + S_k^\top K_k^{(i-1)} + K_k^{(i-1)\top} S_k \\
&\quad - \big[(A_k + B_k K_k^{(i)})^\top P_k^{(i+1)} + P_k^{(i+1)}(A_k + B_k K_k^{(i)}) + (C_k + D_k K_k^{(i)})^\top P_k^{(i+1)}(C_k + D_k K_k^{(i)}) \\
&\quad\quad + \sum_{j=1}^{\mathbf{K}} \pi_{kj} P_j^{(i+1)} + K_k^{(i)\top} R_k K_k^{(i)} + S_k^\top K_k^{(i)} + K_k^{(i)\top} S_k\big] \\
&= (A_k + B_k K_k^{(i)})^\top \Delta P_k^{(i+1)} + \Delta P_k^{(i+1)}(A_k + B_k K_k^{(i)}) + (C_k + D_k K_k^{(i)})^\top \Delta P_k^{(i+1)}(C_k + D_k K_k^{(i)}) \\
&\quad + \sum_{j=1}^{\mathbf{K}} \pi_{kj} \Delta P_j^{(i+1)} + \Delta K_k^{(i)\top} B_k^\top P_k^{(i)} + P_k^{(i)} B_k \Delta K_k^{(i)} \\
&\quad + (C_k + D_k K_k^{(i-1)})^\top P_k^{(i)}(C_k + D_k K_k^{(i-1)}) - (C_k + D_k K_k^{(i)})^\top P_k^{(i)}(C_k + D_k K_k^{(i)}) \\
&\quad + K_k^{(i-1)\top} R_k K_k^{(i-1)} - K_k^{(i)\top} R_k K_k^{(i)} + S_k^\top \Delta K_k^{(i)} + \Delta K_k^{(i)\top} S_k, \qquad k \in \mathcal{M}.
\end{aligned} \tag{15}$$

From (11), one gets

$$\begin{aligned}
&(C_k + D_k K_k^{(i-1)})^\top P_k^{(i)}(C_k + D_k K_k^{(i-1)}) - (C_k + D_k K_k^{(i)})^\top P_k^{(i)}(C_k + D_k K_k^{(i)}) \\
&= \Delta K_k^{(i)\top} D_k^\top P_k^{(i)} D_k \Delta K_k^{(i)} + \Delta K_k^{(i)\top} D_k^\top P_k^{(i)}(C_k + D_k K_k^{(i)}) \\
&\quad + (C_k + D_k K_k^{(i)})^\top P_k^{(i)} D_k \Delta K_k^{(i)}, \qquad k \in \mathcal{M},
\end{aligned} \tag{16}$$

and

$$\begin{aligned}
&K_k^{(i-1)\top} R_k K_k^{(i-1)} - K_k^{(i)\top} R_k K_k^{(i)} \\
&= \Delta K_k^{(i)\top} R_k \Delta K_k^{(i)} + \Delta K_k^{(i)\top} R_k K_k^{(i)} + K_k^{(i)\top} R_k \Delta K_k^{(i)}, \quad k \in \mathcal{M}.
\end{aligned} \tag{17}$$

Combining (15)-(17), we deduce

$$\begin{aligned}
&(A_k + B_k K_k^{(i)})^\top \Delta P_k^{(i+1)} + \Delta P_k^{(i+1)}(A_k + B_k K_k^{(i)}) + (C_k + D_k K_k^{(i)})^\top \Delta P_k^{(i+1)}(C_k + D_k K_k^{(i)}) \\
&\quad + \sum_{j=1}^{\mathbf{K}} \pi_{kj} \Delta P_j^{(i+1)} + \Delta K_k^{(i)\top}(R_k + D_k^\top P_k^{(i)} D_k)\Delta K_k^{(i)} \\
&\quad + \Delta K_k^{(i)\top}\big[B_k^\top P_k^{(i)} + D_k^\top P_k^{(i)} C_k + S_k + (R_k + D_k^\top P_k^{(i)} D_k) K_k^{(i)}\big] \\
&\quad + \big[B_k^\top P_k^{(i)} + D_k^\top P_k^{(i)} C_k + S_k + (R_k + D_k^\top P_k^{(i)} D_k) K_k^{(i)}\big]^\top \Delta K_k^{(i)} = 0, \quad k \in \mathcal{M}.
\end{aligned} \tag{18}$$

By (11), we have

$$-(R_k + D_k^\top P_k^{(i)} D_k) K_k^{(i)} = B_k^\top P_k^{(i)} + D_k^\top P_k^{(i)} C_k + S_k, \quad k \in \mathcal{M}. \tag{19}$$

Plugging (19) into (18), one gets

$$\begin{aligned}(A_k+B_kK_k^{(i)})^\top\Delta P_k^{(i+1)}+\Delta P_k^{(i+1)}(A_k+B_kK_k^{(i)})+(C_k+D_kK_k^{(i)})^\top\Delta P_k^{(i+1)}(C_k+D_kK_k^{(i)})\\+\sum_{j=1}^{\mathbf{K}}\pi_{kj}\Delta P_j^{(i+1)}+\Delta K_k^{(i)\top}(R_k+D_k^\top P_k^{(i)}D_k)\Delta K_k^{(i)}=0,\quad k\in\mathcal{M}.\end{aligned}\tag{20}$$

Since $(K_1^{(i)},K_2^{(i)},\cdots,K_{\mathbf{K}}^{(i)})$ is a stabilizer of system (1) and

$$\Delta K_k^{(i)\top}(R_k+D_k^\top P_k^{(i)}D_k)\Delta K_k^{(i)}\geq 0,\quad k\in\mathcal{M},$$

from Lemma 2.3, Lyapunov equation (20) admits a unique solution $\Delta P_k^{(i+1)}\geq 0$. Then, $\{(P_1^{(i)},P_2^{(i)},\cdots,P_{\mathbf{K}}^{(i)})\}_{i=1}^\infty$ is monotonically nonincreasing. Since $P_k^{(i)}>0$, and $P_k^{(i)}$ has a low bound, so $\{(P_1^{(i)},P_2^{(i)},\cdots,P_{\mathbf{K}}^{(i)})\}_{i=1}^\infty$ converges to $(P_1,P_2,\cdots,P_{\mathbf{K}})\in(\mathcal{S}_+^n)^{\mathbf{K}}$.

Next, we prove that P_k is the solution of CGAREs (7). When $i\to\infty$, as $P_k^{(i)}\to P_k$, we have $\{K_k^{(i)}\}_{i=1}^\infty$ converges to

$$K_k=-(R_k+D_k^\top P_kD_k)^{-1}(B_k^\top P_k+D_k^\top P_kC_k+S_k),\quad k\in\mathcal{M}.\tag{21}$$

Moreover, $(P_1,P_2,\cdots,P_{\mathbf{K}})$ satisfies

$$\begin{aligned}(A_k+B_kK_k)^\top P_k+P_k(A_k+B_kK_k)+(C_k+D_kK_k)^\top P_k(C_k+D_kK_k)\\+\sum_{j=1}^{\mathbf{K}}\pi_{kj}P_j+K_k^\top R_kK_k+S_k^\top K_k+K_k^\top S_k+N_k=0,\quad k\in\mathcal{M}.\end{aligned}\tag{22}$$

Since $K_k^\top R_kK_k+S_k^\top K_k+K_k^\top S_k+N_k>0$, one gets

$$(A_k+B_kK_k)^\top P_k+P_k(A_k+B_kK_k)+(C_k+D_kK_k)^\top P_k(C_k+D_kK_k)+\sum_{j=1}^{\mathbf{K}}\pi_{kj}P_j<0,\quad k\in\mathcal{M}.$$

By Lemma 2.3, $(K_1,K_2,\cdots,K_{\mathbf{K}})$ is a stabilizer of system (1) and (22) admits a unique solution $(P_1,P_2,\cdots,P_{\mathbf{K}})\in(\mathcal{S}_{++}^n)^{\mathbf{K}}$. Moreover, plugging (21) into (22), (22) becomes CGAREs (7). We complete the proof. □

Although Lyapunov Iteration Scheme provides a constructive route to the optimal solution and establishes both stability and convergence, it has two main shortcomings in the implementation. First, it requires all of the system parameters. Second, it is difficult to solve $P_k^{(i+1)}$ $(k\in\mathcal{M})$ from Lyapunov Recursion (10), because $P_1^{(i+1)},P_2^{(i+1)},\cdots,P_{\mathbf{K}}^{(i+1)}$ are fully coupled. That motivates the development of model-free methods to solve the coupled problem. In the following two subsections, we propose two Q-learning algorithms.

3.2 On-Policy Q-Learning Algorithm for Problem (LQ-RS)

In this subsection, we propose an on-policy Q-learning algorithm which solves $P_k^{(i+1)}$ $(k\in\mathcal{M})$ without any system's information.

Now, we define a Q function as

$$\begin{aligned}Q_k(x,u)&=\begin{bmatrix}x\\u\end{bmatrix}^\top\mathbf{Q}_k\begin{bmatrix}x\\u\end{bmatrix}=\begin{bmatrix}x\\u\end{bmatrix}^\top\begin{bmatrix}Q_{kxx}&Q_{kxu}\\Q_{kux}&Q_{kuu}\end{bmatrix}\begin{bmatrix}x\\u\end{bmatrix}\\&=x^\top\Pi(\mathbf{Q}_k)x+(u-\Gamma(\mathbf{Q}_k)x)^\top(R_k+D_k^\top P_kD_k)(u-\Gamma(\mathbf{Q}_k)x),\quad k\in\mathcal{M},\end{aligned}\tag{23}$$

where $\Pi(\mathbf{Q}_k) = Q_{kxx} - Q_{kxu}(Q_{kuu})^{-1}Q_{kux}$ and $\Gamma(\mathbf{Q}_k) = -(Q_{kuu})^{-1}Q_{kux}$ with

$$\begin{aligned}
Q_{kxx} &:= Q_{xx}(P_k) = P_k + A_k^\top P_k + P_k A_k + C_k^\top P_k C_k + \sum_{j=1}^{\mathbf{K}} \pi_{kj} P_j + N_k, && (24)\\
Q_{kxu} &:= Q_{xu}(P_k) = P_k B_k + C_k^\top P_k D_k + S_k^\top, && (25)\\
Q_{kuu} &:= Q_{uu}(P_k) = R_k + D_k^\top P_k D_k, && (26)\\
Q_{kux} &:= Q_{ux}(P_k) = B_k^\top P_k + D_k^\top P_k C_k + S_k.
\end{aligned}$$

For simplicity, we denote $\Theta(P_k^{(i)}) = \Theta_k^{(i)}$, where $\Theta = Q_{xx}, Q_{xu}, Q_{ux}, Q_{uu}$ and $\Theta_k^{(i)} = Q_{kxx}^{(i)}, Q_{kxu}^{(i)}$, $Q_{kux}^{(i)}, Q_{kuu}^{(i)}$. If P_k ($k \in \mathcal{M}$) is the solution of CGAREs (7) and $u_k = \Gamma(\mathbf{Q}_k)x$ is the optimal solution for Problem (LQ-RS), we have

$$Q_k(x, u_k) = x^\top P_k x, \qquad k \in \mathcal{M}. \tag{27}$$

Then, value function is rewritten as

$$\begin{aligned}
Q_k(x, u_k) = \mathbb{E}\Big\{ \int_t^\infty X(s)^\top \Big[& N(\alpha_s) + 2\Gamma(\mathbf{Q}(\alpha_s))^\top S(\alpha_s) \\
& + \Gamma(\mathbf{Q}(\alpha_s))^\top R(\alpha_s)\Gamma(\mathbf{Q}(\alpha_s)) \Big] X(s)ds \;\Big|\; X(t) = x, \alpha_t = k \Big\},
\end{aligned} \tag{28}$$

where $\Gamma(\mathbf{Q}(\alpha_s)) = \Gamma(\mathbf{Q}_k)$ when $\alpha_s = k$ ($k \in \mathcal{M}$).

Based on (28), we formulate the following on-policy Q-learning algorithm to solve optimal control.

Algorithm 1 On-policy Q-learning algorithm for Problem (LQ-RS)

1: **Initialization:** Select any stabilizer $(\Gamma(\mathbf{Q}_1^{(0)}), \Gamma(\mathbf{Q}_2^{(0)}), \cdots, \Gamma(\mathbf{Q}_{\mathbf{K}}^{(0)}))$ for system (1).
2: Let $i = 0$ and $\varepsilon > 0$.
3: **do** {
4: Running system (1) with $u^{(i)}(s) = \sum_{k=1}^{\mathbf{K}} \Gamma(\mathbf{Q}_k^{(i)}) X^{(i)}(s) \chi_{\{\alpha_s = k\}}(s)$ on $[t, \infty)$.
5: **Policy Evaluation** (Reinforcement): Solve $\mathbf{Q}_k^{(i+1)}$ ($k \in \mathcal{M}$) from the identity

$$\begin{aligned}
\begin{bmatrix} x \\ u_k^{(i)} \end{bmatrix}^\top \mathbf{Q}_k^{(i+1)} \begin{bmatrix} x \\ u_k^{(i)} \end{bmatrix} = \mathbb{E}\Big\{ \int_t^\infty X^{(i)}(s)^\top \Big[& N(\alpha_s) + 2\Gamma(\mathbf{Q}^{(i)}(\alpha_s))^\top S(\alpha_s) \\
& + \Gamma(\mathbf{Q}^{(i)}(\alpha_s))^\top R(\alpha_s)\Gamma(\mathbf{Q}^{(i)}(\alpha_s)) \Big] X^{(i)}(s)ds \;\Big|\; X^{(i)}(t) = x, \alpha_t = k \Big\}, \; k \in \mathcal{M}.
\end{aligned} \tag{29}$$

6: **Policy Improvement** (Update): Update $\Gamma(\mathbf{Q}_k^{(i+1)})$ by

$$\Gamma(\mathbf{Q}_k^{(i+1)}) = -(Q_{kuu}^{(i+1)})^{-1}(Q_{kux}^{(i+1)}), \qquad k \in \mathcal{M}. \tag{30}$$

7: $i \leftarrow i + 1$.
8: } **until** $\|\mathbf{Q}_k^{(i+1)} - \mathbf{Q}_k^{(i)}\| < \varepsilon$ for all $k \in \mathcal{M}$.

By virtue of the introduction of Q-function, Algorithm 1 does not require the knowledge of the system dynamics, which means that it is a totally model-free algorithm. In addition, Algorithm 1 overcomes the difficulty of the randomness of $P_k^{(i+1)}$ ($k \in \mathbf{M}$) at the future time $t + \Delta t$ and decouples $\{P_k^{(i+1)}\}_{k=1}^{\mathbf{K}}$ by system stability properties over the entire time interval $[t, +\infty)$.

Lemma 3.3. *Algorithm 1 is equivalent to Lyapunov Iteration Scheme* (10)-(11).

Proof. We apply generalized Itô's formula to $X^{(i)}(s)^\top P^{(i+1)}(\alpha_s)X^{(i)}(s)$ by Lemma 2.2. Suppose $P^{(i+1)}(\alpha_s)$ is the solution of Lyapunov Recursion (10), and $X^{(i)}(s)$ is the state in (1). We have

$$\begin{aligned}
&d\Big[X^{(i)}(s)^\top P^{(i+1)}(\alpha_s)X^{(i)}(s)\Big]\\
&=\Big\{X^{(i)}(s)^\top\Big[\big(A(\alpha_s)+B(\alpha_s)K^{(i)}(\alpha_s)\big)^\top P^{(i+1)}(\alpha_s)+P^{(i+1)}(\alpha_s)\big(A(\alpha_s)+B(\alpha_s)K^{(i)}(\alpha_s)\big)\\
&\quad+\big(C(\alpha_s)+D(\alpha_s)K^{(i)}(\alpha_s)\big)^\top P^{(i+1)}(\alpha_s)\big(C(\alpha_s)+D(\alpha_s)K^{(i)}(\alpha_s)\big)\\
&\quad+\sum_{j=1}^{\mathbf{K}}\pi_{\alpha_s j}P_j^{(i+1)}\Big]X^{(i)}(s)\Big\}ds+\{...\}\,dW(s).
\end{aligned}\tag{31}$$

Integrating (31) on $[t,\infty)$ and taking expectations on both sides, one gets

$$\begin{aligned}
&x^\top P_k^{(i+1)}x\\
&=-\mathbb{E}\Big\{\int_t^\infty X^{(i)}(s)^\top\Big[\big(A(\alpha_s)+B(\alpha_s)K^{(i)}(\alpha_s)\big)^\top P^{(i+1)}(\alpha_s)+P^{(i+1)}(\alpha_s)\big(A(\alpha_s)+B(\alpha_s)K^{(i)}(\alpha_s)\big)\\
&\quad+\big(C(\alpha_s)+D(\alpha_s)K^{(i)}(\alpha_s)\big)^\top P^{(i+1)}(\alpha_s)\big(C(\alpha_s)+D(\alpha_s)K^{(i)}(\alpha_s)\big)\\
&\quad+\sum_{j=1}^{\mathbf{K}}\pi_{\alpha_s j}P_j^{(i+1)}\Big]X^{(i)}(s)ds\Big|X^{(i)}(t)=x,\alpha_t=k\Big\}\\
&=-\mathbb{E}\Big\{\sum_{k=1}^{\mathbf{K}}\int_t^\infty X^{(i)}(s)^\top\Big[\big(A_k+B_kK_k^{(i)}\big)^\top P_k^{(i+1)}+P_k^{(i+1)}\big(A_k+B_kK_k^{(i)}\big)+\sum_{j=1}^{\mathbf{K}}\pi_{kj}P_j^{(i+1)}\\
&\quad+\big(C_k+D_kK_k^{(i)}\big)^\top P_k^{(i+1)}\big(C_k+D_kK_k^{(i)}\big)\Big]X^{(i)}(s)\chi_{\{\alpha_s=k\}}(s)ds\Big|X^{(i)}(t)=x,\alpha_t=k\Big\}.
\end{aligned}\tag{32}$$

Taking Lyapunov Recursion (10) into (32) yields

$$\begin{aligned}
x^\top P_k^{(i+1)}x=\mathbb{E}\Big\{\int_t^\infty X^{(i)}(s)^\top\Big[&N(\alpha_s)+2K^{(i)}(\alpha_s)^\top S(\alpha_s)\\
&+K^{(i)}(\alpha_s)^\top R(\alpha_s)K^{(i)}(\alpha_s)\Big]X^{(i)}(s)ds\Big|X^{(i)}(t)=x,\alpha_t=k\Big\},\quad k\in\mathcal{M}.
\end{aligned}\tag{33}$$

On the other hand, if $P_k^{(i+1)}\in\mathcal{S}_{++}^n$ $(k\in\mathcal{M})$ is the solution of (33), for any $t>0$, integrating (31) on $[t,t+\Delta t]$ and taking expectations on both sides, one gets

$$\begin{aligned}
&\mathbb{E}\big[X^{(i)}(t+\Delta t)^\top P^{(i+1)}(\alpha_{t+\Delta t})X^{(i)}(t+\Delta t)\big|X^{(i)}(t)=x,\alpha_t=k\big]-x^\top P_k^{(i+1)}x\\
&=\mathbb{E}\Big\{\int_t^{t+\Delta t}X^{(i)}(s)^\top\Big[\big(A(\alpha_s)+B(\alpha_s)K^{(i)}(\alpha_s)\big)^\top P^{(i+1)}(\alpha_s)+P^{(i+1)}(\alpha_s)\big(A(\alpha_s)+B(\alpha_s)K^{(i)}(\alpha_s)\big)\\
&\quad+\big(C(\alpha_s)+D(\alpha_s)K^{(i)}(\alpha_s)\big)^\top P^{(i+1)}(\alpha_s)\big(C(\alpha_s)+D(\alpha_s)K^{(i)}(\alpha_s)\big)\\
&\quad+\sum_{j=1}^{\mathbf{K}}\pi_{\alpha_s j}P_j^{(i+1)}\Big]X^{(i)}(s)ds\Big|X^{(i)}(t)=x,\alpha_t=k\Big\},\qquad k\in\mathcal{M}.
\end{aligned}\tag{34}$$

From (33), we obtain

$$\begin{aligned} x^\top P_k^{(i+1)} x = \mathbb{E}\Big\{ \int_t^{t+\Delta t} & X^{(i)}(s)^\top \Big[N(\alpha_s) + 2K^{(i)}(\alpha_s)^\top S(\alpha_s) \\ & + K^{(i)}(\alpha_s)^\top R(\alpha_s) K^{(i)}(\alpha_s) \Big] X^{(i)}(s) ds \Big| X^{(i)}(t) = x, \alpha_t = k \Big\} \\ & + \mathbb{E}\big[X^{(i)}(t+\Delta t)^\top P^{(i+1)}(\alpha_{t+\Delta t}) X^{(i)}(t+\Delta t) \big| X^{(i)}(t) = x, \alpha_t = k \big], \quad k \in \mathcal{M}. \end{aligned} \tag{35}$$

Combining (34) and (35), we have, for $k \in \mathcal{M}$,

$$\begin{aligned} \mathbb{E}\Big\{ \int_t^{t+\Delta t} & X^{(i)}(s)^\top \Big[\big(A(\alpha_s) + B(\alpha_s) K^{(i)}(\alpha_s)\big)^\top P^{(i+1)}(\alpha_s) + P^{(i+1)}(\alpha_s) \big(A(\alpha_s) + B(\alpha_s) K^{(i)}(\alpha_s)\big) \\ & + \big(C(\alpha_s) + D(\alpha_s) K^{(i)}(\alpha_s)\big)^\top P^{(i+1)}(\alpha_s) \big(C(\alpha_s) + D(\alpha_s) K^{(i)}(\alpha_s)\big) + \sum_{j=1}^{\mathbf{K}} \pi_{\alpha_s j} P_j^{(i+1)} \\ & + N(\alpha_s) + 2K^{(i)}(\alpha_s)^\top S(\alpha_s) + K^{(i)}(\alpha_s)^\top R(\alpha_s) K^{(i)}(\alpha_s) \Big] X^{(i)}(s) ds \Big| X^{(i)}(t) = x, \alpha_t = k \Big\} = 0. \end{aligned} \tag{36}$$

Dividing Δt on both sides of (36), let $\Delta t \to 0$, one gets

$$\begin{aligned} x^\top \Big[& (A_k + B_k K_k^{(i)})^\top P_k^{(i+1)} + P_k^{(i+1)} (A_k + B_k K_k^{(i)}) + (C_k + D_k K_k^{(i)})^\top P_k^{(i+1)} (C_k + D_k K_k^{(i)}) \\ & + \sum_{j=1}^{\mathbf{K}} \pi_{kj} P_j^{(i+1)} + K_k^{(i)\top} R_k K_k^{(i)} + S_k^\top K_k^{(i)} + K_k^{(i)\top} S_k + N_k \Big] x = 0, \quad k \in \mathcal{M}, \end{aligned} \tag{37}$$

which holds true for any $x \in \mathbb{R}^n$, and $P_k^{(i+1)} \in \mathcal{S}_{++}^n$ $(k \in \mathcal{M})$ satisfies Lyapunov Recursion (10). So, (33) is equivalent to Lyapunov Recursion (10).

Rewriting (37) with $u_k^{(i)} = K_k^{(i)} x$ $(k \in \mathcal{M})$, yields

$$\begin{bmatrix} x \\ u_k^{(i)} \end{bmatrix}^\top \begin{bmatrix} Q_{kxx}^{(i+1)} - P_k^{(i+1)} & Q_{kxu}^{(i+1)} \\ Q_{kux}^{(i+1)} & Q_{kuu}^{(i+1)} \end{bmatrix} \begin{bmatrix} x \\ u_k^{(i)} \end{bmatrix} = 0, \quad k \in \mathcal{M}. \tag{38}$$

Adding (38) to (33), we obtain

$$\begin{aligned} & \begin{bmatrix} x \\ u_k^{(i)} \end{bmatrix}^\top \begin{bmatrix} Q_{kxx}^{(i+1)} - P_k^{(i+1)} & Q_{kxu}^{(i+1)} \\ Q_{kux}^{(i+1)} & Q_{kuu}^{(i+1)} \end{bmatrix} \begin{bmatrix} x \\ u_k^{(i)} \end{bmatrix} + \begin{bmatrix} x \\ u_k^{(i)} \end{bmatrix}^\top \begin{bmatrix} P_k^{(i+1)} & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x \\ u_k^{(i)} \end{bmatrix} \\ = \mathbb{E}\Big\{ & \int_t^\infty X^{(i)}(s)^\top \Big[N(\alpha_s) + 2\Gamma(\mathbf{Q}^{(i)}(\alpha_s))^\top S(\alpha_s) \\ & + \Gamma(\mathbf{Q}^{(i)}(\alpha_s))^\top R(\alpha_s) \Gamma(\mathbf{Q}^{(i)}(\alpha_s)) \Big] X^{(i)}(s) ds \Big| X^{(i)}(t) = x, \alpha_t = k \Big\}, \quad k \in \mathcal{M}, \end{aligned}$$

which confirms (29). Since (29) is equivalent to (33), and (33) is equivalent to Lyapunov Recursion (10), then (29) is equivalent to Lyapunov Recursion (10).

Moreover, it is easy to verify that

$$\begin{aligned} \Gamma(\mathbf{Q}_k^{(i+1)}) &= -(Q_{kuu}^{(i+1)})^{-1} Q_{kux}^{(i+1)} \\ &= -\Big(R_k + D_k^\top P_k^{(i+1)} D_k \Big)^{-1} \Big(B_k^\top P_k^{(i+1)} + D_k^\top P_k^{(i+1)} C_k + S_k \Big) \\ &= K_k^{(i+1)}, \ k \in \mathcal{M}. \end{aligned} \tag{39}$$

We complete the proof. □

Based on the equivalence, we show that Algorithm 1 also has stabilization and convergence properties.

Theorem 3.4. *Assume Assumption 2.1 holds. Let* $(\Gamma(\mathbf{Q}_1^{(0)}), \Gamma(\mathbf{Q}_2^{(0)}), ..., \Gamma(\mathbf{Q}_{\mathbf{K}}^{(0)}))$ *be the known initial stabilizer for system* (1)*, then all the control gains* $\{\Gamma(\mathbf{Q}_k^{(i)})\}_{i=1}^{\infty}$ $(k \in \mathcal{M})$ *updated by* (30) *are stabilizers. Moreover,* $\{\mathbf{Q}_k^{(i+1)}\}_{i=0}^{\infty}$ $(k \in \mathcal{M})$ *in* (29) *converges to* $\mathbf{Q}_k$ $(k \in \mathcal{M})$ *and* $\{\Gamma(\mathbf{Q}_k^{(i)})\}_{i=1}^{\infty}$ $(k \in \mathcal{M})$ *converges to* $\Gamma(\mathbf{Q}_k)$ $(k \in \mathcal{M})$*. The optimal control is* $u^* = \sum_{k=1}^{\mathbf{K}} \Gamma(\mathbf{Q}_k) X^* \times \chi_{\{\alpha_s = k\}}(s)$*, where* X^* *is the corresponding optimal state with* u^* *of system* (1)*.*

Proof. From Theorems 3.1 and 3.2, combining Lemma 3.3, we obtain the conclusion. □

Next, we show how to implement Algorithm 1. To solve the parameters of $\mathbf{Q}_k^{(i+1)}$ $(k \in \mathcal{M})$ in (29) and $\Gamma(\mathbf{Q}_k^{(i+1)})$ $(k \in \mathcal{M})$ in (30), from [15], we randomly choose $\mathbf{H}$ initial states $x_h \in \mathbb{R}^n$ and $\mathbf{K}$ initial states $k \in \mathcal{M}$ to generate the corresponding state trajectories $X_{h,k}^{(i)}(s) = X^{(i)}(s; t, x_h, k)$ over the horizon $[t, \infty)$ with $h = 1, 2, \ldots, \mathbf{H}$. In addition, we calculate the conditional expectation of $\mathbf{N}$ sample paths $X_{h,k,\nu}^{(i)}$ $(\nu = 1, 2, ..., \mathbf{N})$ under the known initial (t, x_h, k). Moreover, we collect trajectory samples at times t_l $(t_l \geq 0)$ and adopt a step size $\delta_{t_l} = t_{l+1} - t_l$, where $l = 1, 2, \ldots, \mathbf{L}$, and $\mathbf{L}$ is sufficiently large. We denote

$$L_{xu,k}^{(i)} = \begin{bmatrix} \bar{x}_1 & 2x_1^\top \otimes {u_{1,k}^{(i)}}^\top & \bar{u}_{1,k}^{(i)} \\ \bar{x}_2 & 2x_2^\top \otimes {u_{2,k}^{(i)}}^\top & \bar{u}_{2,k}^{(i)} \\ \vdots & \vdots & \vdots \\ \bar{x}_{\mathbf{H}} & 2x_{\mathbf{H}}^\top \otimes {u_{\mathbf{H},k}^{(i)}}^\top & \bar{u}_{\mathbf{H},k}^{(i)} \end{bmatrix}, \quad k \in \mathcal{M},$$

where $u_{h,k}^{(i)} = \Gamma(\mathbf{Q}_k^{(i)}) x_h + \omega$ and ω is a persistent excitation signal. We denote

$$\Psi_k^{(i)} = \begin{bmatrix} \frac{1}{\mathbf{N}} \sum_{\nu=1}^{\mathbf{N}} \left[\sum_{l=1}^{\mathbf{L}} \left(X_{1,k,\nu}^{(i)}(t_l)\right)^\top \Sigma^{(i)}(\alpha_{t_l}) X_{1,k,\nu}^{(i)}(t_l) \delta_{t_l} \right] \\ \frac{1}{\mathbf{N}} \sum_{\nu=1}^{\mathbf{N}} \left[\sum_{l=1}^{\mathbf{L}} \left(X_{2,k,\nu}^{(i)}(t_l)\right)^\top \Sigma^{(i)}(\alpha_{t_l}) X_{2,k,\nu}^{(i)}(t_l) \delta_{t_l} \right] \\ \vdots \\ \frac{1}{\mathbf{N}} \sum_{\nu=1}^{\mathbf{N}} \left[\sum_{l=1}^{\mathbf{L}} \left(X_{\mathbf{H},k,\nu}^{(i)}(t_l)\right)^\top \Sigma^{(i)}(\alpha_{t_l}) X_{\mathbf{H},k,\nu}^{(i)}(t_l) \delta_{t_l} \right] \end{bmatrix}, \quad k \in \mathcal{M},$$

where $\Sigma^{(i)}(\alpha_{t_l}) = N(\alpha_{t_l}) + 2K^{(i)}(\alpha_{t_l})^\top S(\alpha_{t_l}) + \left(K^{(i)}(\alpha_{t_l})\right)^\top R(\alpha_{t_l}) K^{(i)}(\alpha_{t_l})$. We note that the number of initial states $\mathbf{H}$ must satisfy $\mathbf{H} \geq \frac{1}{2}[n(n+1) + 2nr + r(r+1)]$ to solve $Q_{kxx}^{(i+1)}$, $Q_{kux}^{(i+1)}$ and $Q_{kuu}^{(i+1)}$ with $k \in \mathcal{M}$.

Algorithm 1 is required to satisfy a persistent excitation (PE) condition (Bradtke et al. [2], Al-Tamimi et al. [1]), which is exactly the following assumption. The PE signal ω is introduced in $u_{h,k}^{(i)} = \Gamma(\mathbf{Q}_k^{(i)}) x_h + \omega$ to guarantee that $L_{xu,k}^{(i)}$ has full column rank, which ensures that $\mathbf{Q}_k^{(i+1)}$ $(k \in \mathcal{M})$ can be uniquely solved.

Assumption 3.1. *In order to guarantee that Algorithm 1 can be implemented online at each step,* $L_{xu,k}^{(i)}$ *is required to have full column rank, we assume there exists* $\mathbf{H}_0 > 0$ *such that for all* $\mathbf{H} \geq \mathbf{H}_0$*,* $\operatorname{rank}(L_{xu,k}^{(i)}) = \frac{1}{2}[n(n+1) + 2nr + r(r+1)]$.

Under Assumption 3.1, using least-squares method, we have

$$\begin{bmatrix} vech(Q_{kxx}^{(i+1)}) \\ vec(Q_{kxu}^{(i+1)}) \\ vech(Q_{kuu}^{(i+1)}) \end{bmatrix} = \left({L_{xu,k}^{(i)}}^\top L_{xu,k}^{(i)} \right)^{-1} {L_{xu,k}^{(i)}}^\top \Psi_k^{(i)}, \quad k \in \mathcal{M}. \tag{40}$$

So, (29) can be solved by (40). We also can obtain (30).

3.3 Off-Policy Q-Learning Algorithm for Problem (LQ-RS)

In Subsection 3.2, we introduce the on-policy Q-learning algorithm for Problem (LQ-RS), which generates a new trajectory data pair $(X^{(i)}(s), u^{(i)}(s))$ for each iteration, resulting in low data utilization efficiency and increased running time. We now present an off-policy Q-learning algorithm for Problem (LQ-RS), which reuses trajectory data to improve data utilization efficiency and reduce running time.

We consider using off-policy trajectory data pair $(X(s), u(s))$ to calculate $\mathbf{Q}_k^{(i+1)}$ and $\Gamma(\mathbf{Q}_k^{(i+1)})$ for $k \in \mathcal{M}$. We demonstrate the following off-policy Q-learning algorithm for Problem (LQ-RS).

Algorithm 2 Off-policy Q-learning algorithm for Problem (LQ-RS)

1: **Initialization:** Select any stabilizer $(\Gamma(\mathbf{Q}_1^{(0)}), \Gamma(\mathbf{Q}_2^{(0)}), \cdots, \Gamma(\mathbf{Q}_{\mathbf{K}}^{(0)}))$ for system (1). Collect stabilizing control input trajectory $u(s)$ and obtain state trajectory $X(s)$ by running system (1) on $[t, \infty)$.
2: Let $i = 0$ and $\varepsilon > 0$.
3: **do** {
4: **Policy Evaluation** (Reinforcement): Solve $\mathbf{Q}_k^{(i+1)}$ $(k \in \mathcal{M})$ from the identity

$$\begin{aligned}
&\begin{bmatrix} x \\ u_k^{(i)} \end{bmatrix}^\top \mathbf{Q}_k^{(i+1)} \begin{bmatrix} x \\ u_k^{(i)} \end{bmatrix} + \mathbb{E}\Big\{ \int_t^\infty 2\Big(u(s) - \Gamma(\mathbf{Q}^{(i)}(\alpha_s))X(s)\Big)^\top Q_{ux}^{(i+1)}(\alpha_s)X(s) \\
&\quad + \Big(u(s) - \Gamma(\mathbf{Q}^{(i)}(\alpha_s))X(s)\Big)^\top Q_{uu}^{(i+1)}(\alpha_s)\Big(u(s) + \Gamma(\mathbf{Q}^{(i)}(\alpha_s))X(s)\Big)ds \Big| X(t) = x, \alpha_t = k \Big\} \\
&= \mathbb{E}\Big\{ \int_t^\infty 2u(s)^\top S(\alpha_s)X(s) + u(s)^\top R(\alpha_s)u(s) + X(s)^\top N(\alpha_s)X(s)ds \Big| X(t) = x, \alpha_t = k \Big\},
\end{aligned} \tag{41}$$

where $Q_{ux}^{(i+1)}(\alpha_s) = Q_{kux}^{(i+1)}$, $Q_{uu}^{(i+1)}(\alpha_s) = Q_{kuu}^{(i+1)}$ when $\alpha_s = k$ $(k \in \mathcal{M})$.
5: **Policy Improvement** (Update): Update $\Gamma(\mathbf{Q}_k^{(i+1)})$ with formula (30).
6: $i \leftarrow i + 1$.
7: } **until** $\|\mathbf{Q}_k^{(i+1)} - \mathbf{Q}_k^{(i)}\| < \varepsilon$ for all $k \in \mathcal{M}$.

Theorem 3.5. *Assume Assumption 2.1 holds. Let* $(\Gamma(\mathbf{Q}_1^{(0)}), \Gamma(\mathbf{Q}_2^{(0)}), ..., \Gamma(\mathbf{Q}_{\mathbf{K}}^{(0)}))$ *be the known initial stabilizer for system* (1)*, then all the control gains* $\{\Gamma(\mathbf{Q}_k^{(i)})\}_{i=1}^\infty$ $(k \in \mathcal{M})$ *updated by* (30) *are stabilizers. Moreover,* $\{\mathbf{Q}_k^{(i+1)}\}_{i=0}^\infty$ $(k \in \mathcal{M})$ *in* (41) *converges to* $\mathbf{Q}_k$ $(k \in \mathcal{M})$ *and* $\{\Gamma(\mathbf{Q}_k^{(i)})\}_{i=1}^\infty$ $(k \in \mathcal{M})$ *converges to* $\Gamma(\mathbf{Q}_k)$ $(k \in \mathcal{M})$*. The optimal control is* $u^* = \sum_{k=1}^{\mathbf{K}} \Gamma(\mathbf{Q}_k)X^* \times \chi_{\{\alpha_s = k\}}(s)$*, where* X^* *is the corresponding optimal state with* u^* *of system* (1)*.*

Proof. First, we prove that Algorithm 2 is equivalent to Lyapunov Iteration Scheme. We rewrite (1) as

$$\begin{aligned}
dX(s) = &\Big[A(\alpha_s)X(s) + B(\alpha_s)K^{(i)}(\alpha_s)X(s) + B(\alpha_s)\big(u(s) - K^{(i)}(\alpha_s)X(s)\big)\Big]ds \\
&+ \Big[C(\alpha_s)X(s) + D(\alpha_s)K^{(i)}(\alpha_s)X(s) + D(\alpha_s)\big(u(s) - K^{(i)}(\alpha_s)X(s)\big)\Big]dW(s).
\end{aligned} \tag{42}$$

We apply generalized Itô's formula to $X(s)^\top P^{(i+1)}(\alpha_s)X(s)$ by Lemma 2.2, where $P^{(i+1)}(\alpha_s)$ is

the solution of Lyapunov Recursion (10), and $X(s)$ is the state in (1). We have

$$
\begin{aligned}
&d\Big[X(s)^\top P^{(i+1)}(\alpha_s)X(s)\Big] \\
&= X(s)^\top\Big[P^{(i+1)}(\alpha_s)\big(A(\alpha_s)+B(\alpha_s)K^{(i)}(\alpha_s)\big)+(A(\alpha_s)+B(\alpha_s)K^{(i)}(\alpha_s))^\top P^{(i+1)}(\alpha_s)\Big]X(s)ds \\
&\quad + 2X(s)^\top P^{(i+1)}(\alpha_s)B(\alpha_s)(u(s)-K^{(i)}(\alpha_s)X(s))ds + \Big[(C(\alpha_s)+D(\alpha_s)K^{(i)}(\alpha_s))X(s) \\
&\quad + D(\alpha_s)(u(s)-K^{(i)}(\alpha_s)X(s))\Big]^\top P^{(i+1)}(\alpha_s)\Big[(C(\alpha_s)+D(\alpha_s)K^{(i)}(\alpha_s))X(s) \\
&\quad + D(\alpha_s)(u(s)-K^{(i)}(\alpha_s)X(s))\Big]ds + X(s)^\top\sum_{j=1}^{\mathbf{K}}\pi_{\alpha_s j}P_j^{(i+1)}X(s)ds + \{\ldots\}dW(s). \qquad (43)
\end{aligned}
$$

Then, integrating both sides of (43) and taking conditional expectation from t to ∞, we obtain

$$
\begin{aligned}
&x^\top P_k^{(i+1)}x \\
&= -\mathbb{E}\Big\{\sum_{k=1}^{\mathbf{K}}\int_t^\infty\Big[X(s)^\top\Big[P_k^{(i+1)}(A_k+B_kK_k^{(i)})+(A_k+B_kK_k^{(i)})^\top P_k^{(i+1)} \\
&\quad + (C_k+D_kK_k^{(i)})^\top P_k^{(i+1)}(C_k+D_kK_k^{(i)})+\sum_{j=1}^{\mathbf{K}}\pi_{kj}P_j^{(i+1)}\Big]X(s) \\
&\quad + 2X(s)^\top P_k^{(i+1)}B_k\Big(u(s)-K_k^{(i)}X(s)\Big)+2X(s)^\top\Big(C_k+D_kK_k^{(i)}\Big)^\top P_k^{(i+1)}D_k\Big(u(s)-K_k^{(i)}X(s)\Big) \\
&\quad + \Big(u(s)-K_k^{(i)}X(s)\Big)^\top D_k^\top P_k^{(i+1)}D_k\Big(u(s)-K_k^{(i)}X(s)\Big)\Big]\chi_{\{\alpha_s=k\}}(s)ds\Big|X(t)=x,\alpha_t=k\Big\}, \\
&\qquad\qquad k\in\mathcal{M}. \qquad (44)
\end{aligned}
$$

Substituting (10) into (44), we obtain

$$
\begin{aligned}
&x^\top P_k^{(i+1)}x+\mathbb{E}\Big\{\int_t^\infty\Big[2\Big(u(s)-K^{(i)}(\alpha_s)X(s)\Big)^\top\Big(B(\alpha_s)^\top P^{(i+1)}(\alpha_s) \\
&\quad + D(\alpha_s)^\top P^{(i+1)}(\alpha_s)C(\alpha_s)\Big)X(s)+\Big(u(s)-K^{(i)}(\alpha_s)X(s)\Big)^\top D(\alpha_s)^\top P^{(i+1)}(\alpha_s)D(\alpha_s)\Big(u(s) \\
&\quad + K^{(i)}(\alpha_s)X(s)\Big)\Big]ds\Big|X(t)=x,\alpha_t=k\Big\} \\
&= \mathbb{E}\Big\{\int_t^\infty X(s)^\top\Big[N(\alpha_s)+K^{(i)}(\alpha_s)^\top R(\alpha_s)K^{(i)}(\alpha_s) \\
&\quad + 2S(\alpha_s)^\top K^{(i)}(\alpha_s)\Big]X(s)ds\Big|X(t)=x,\alpha_t=k\Big\}, k\in\mathcal{M}. \qquad (45)
\end{aligned}
$$

Adding (38) to (45) and using $u_k^{(i)} = \Gamma(\mathbf{Q}_k^{(i)})x$ $(k \in \mathcal{M})$, we obtain

$$\begin{aligned}
&\begin{bmatrix} x \\ u_k^{(i)} \end{bmatrix}^\top \mathbf{Q}_k^{(i+1)} \begin{bmatrix} x \\ u_k^{(i)} \end{bmatrix} + \mathbb{E}\Big\{ \int_t^\infty \Big[2\Big(u(s) - \Gamma(\mathbf{Q}^{(i)}(\alpha_s))X(s)\Big)^\top \Big(Q_{ux}^{(i+1)}(\alpha_s) - S(\alpha_s)\Big)X(s) \\
&+ \Big(u(s) - \Gamma(\mathbf{Q}^{(i)}(\alpha_s))X(s)\Big)^\top \Big(Q_{uu}^{(i+1)}(\alpha_s) - R(\alpha_s)\Big)\Big(u(s) \\
&+ \Gamma(\mathbf{Q}^{(i)}(\alpha_s))X(s)\Big)\Big] ds \Big| X(t) = x, \alpha_t = k \Big\} \\
&= \mathbb{E}\Big\{ \int_t^\infty X(s)^\top \Big[N(\alpha_s) + \Gamma(\mathbf{Q}^{(i)}(\alpha_s))^\top R(\alpha_s)\Gamma(\mathbf{Q}^{(i)}(\alpha_s)) \\
&\qquad + 2S(\alpha_s)^\top \Gamma(\mathbf{Q}^{(i)}(\alpha_s))\Big] X(s) ds \Big| X(t) = x, \alpha_t = k \Big\}, \qquad k \in \mathcal{M}. \qquad (46)
\end{aligned}$$

Deleting the same term $X(s)^\top \big[2S(\alpha_s)^\top \Gamma(\mathbf{Q}^{(i)}(\alpha_s)) + \Gamma(\mathbf{Q}^{(i)}(\alpha_s))^\top R(\alpha_s)\Gamma(\mathbf{Q}^{(i)}(\alpha_s))\big]X(s)$ from both sides of (46), we have (41). Conversely, if $P_k^{(i+1)} \in \mathcal{S}_{++}^n$ $(k \in \mathcal{M})$ is the solution to (41), then similarly to the proof of Lemma 3.3, it is readily verified that (10) holds. Moreover, (11) is equivalent to (30) from Lemma 3.3. Therefore, Algorithm 2 is equivalent to Lyapunov Iteration Scheme.

Then, from Theorems 3.1 and 3.2, we obtain the conclusion. □

Next, we show the implementation of Algorithm 2. To solve the parameters of $\mathbf{Q}_k^{(i+1)}$ $(k \in \mathcal{M})$ in (41) of Algorithm 2 and $\Gamma(\mathbf{Q}_k^{(i+1)})$ $(k \in \mathcal{M})$ in (30), we generate the state trajectory $X_{h,k}(s) = X(s; t, x_h, k)$, under the corresponding control input trajectory $u_{h,k}(s) = u(s; t, x_h, k)$ over horizon $[t, \infty)$ with $h = 1, 2, \ldots, \mathbf{H}$. In addition, we take $u_{h,k}^{(i)} = \Gamma(\mathbf{Q}_k^{(i)})x_h$ $(k \in \mathcal{M})$ in Algorithm 2. We calculate the conditional expectation of $\mathbf{N}$ sample paths $X_{h,k,\nu}$ and $u_{h,k,\nu}$ $(\nu = 1, 2, ..., \mathbf{N})$. Moreover, we adopt δ_t step size, and the times at which the Markov chain state k is equal to j are denoted as t_l^j $(0 \le t_l^j)$, where $l = 1, 2, \ldots, \mathbf{L}$, and $\mathbf{L}$ is sufficiently large. We denote

$$I_{xx} = \begin{bmatrix} \bar{x}_1 \\ \bar{x}_2 \\ \vdots \\ \bar{x}_{\mathbf{H}} \end{bmatrix}, \qquad \delta_{ux,k}^{(i)} = \begin{bmatrix} u_{1,k}^{(i)\top} \otimes x_1^\top \\ u_{2,k}^{(i)\top} \otimes x_2^\top \\ \vdots \\ u_{\mathbf{H},k}^{(i)\top} \otimes x_{\mathbf{H}}^\top \end{bmatrix}, \qquad \delta_{uu,k}^{(i)} = \begin{bmatrix} \bar{u}_{1,k}^{(i)} \\ \bar{u}_{2,k}^{(i)} \\ \vdots \\ \bar{u}_{\mathbf{H},k}^{(i)} \end{bmatrix},$$

$$I_{uu}^{kj} = \begin{bmatrix} \frac{1}{\mathbf{N}} \sum_{\nu=1}^{\mathbf{N}} \Big[\sum_{l=1}^{\mathbf{L}} \bar{u}_{1,k,\nu}(t_l^j)\delta_t\Big] \\ \frac{1}{\mathbf{N}} \sum_{\nu=1}^{\mathbf{N}} \Big[\sum_{l=1}^{\mathbf{L}} \bar{u}_{2,k,\nu}(t_l^j)\delta_t\Big] \\ \vdots \\ \frac{1}{\mathbf{N}} \sum_{\nu=1}^{\mathbf{N}} \Big[\sum_{l=1}^{\mathbf{L}} \bar{u}_{\mathbf{H},k,\nu}(t_l^j)\delta_t\Big] \end{bmatrix}, \quad I_{XX}^{kj} = \begin{bmatrix} \frac{1}{\mathbf{N}} \sum_{\nu=1}^{\mathbf{N}} \Big[\sum_{l=1}^{\mathbf{L}} X_{1,k,\nu}(t_l^j) \otimes X_{1,k,\nu}(t_l^j)\delta_t\Big] \\ \frac{1}{\mathbf{N}} \sum_{\nu=1}^{\mathbf{N}} \Big[\sum_{l=1}^{\mathbf{L}} X_{2,k,\nu}(t_l^j) \otimes X_{2,k,\nu}(t_l^j)\delta_t\Big] \\ \vdots \\ \frac{1}{\mathbf{N}} \sum_{\nu=1}^{\mathbf{N}} \Big[\sum_{l=1}^{\mathbf{L}} X_{\mathbf{H},k,\nu}(t_l^j) \otimes X_{\mathbf{H},k,\nu}(t_l^j)\delta_t\Big] \end{bmatrix},$$

$$I_{Xu}^{kj} = \begin{bmatrix} \frac{1}{\mathbf{N}} \sum_{\nu=1}^{\mathbf{N}} \Big[\sum_{l=1}^{\mathbf{L}} X_{1,k,\nu}(t_l^j) \otimes u_{1,k,\nu}(t_l^j)\delta_t\Big] \\ \frac{1}{\mathbf{N}} \sum_{\nu=1}^{\mathbf{N}} \Big[\sum_{l=1}^{\mathbf{L}} X_{2,k,\nu}(t_l^j) \otimes u_{2,k,\nu}(t_l^j)\delta_t\Big] \\ \vdots \\ \frac{1}{\mathbf{N}} \sum_{\nu=1}^{\mathbf{N}} \Big[\sum_{l=1}^{\mathbf{L}} X_{\mathbf{H},k,\nu}(t_l^j) \otimes u_{\mathbf{H},k,\nu}(t_l^j)\delta_t\Big] \end{bmatrix}, \quad \gamma_k = \begin{bmatrix} \frac{1}{\mathbf{N}} \sum_{\nu=1}^{\mathbf{N}} \Big[\sum_{l=1}^{\mathbf{L}} \Xi_{1,k,\nu}(\alpha_{t_l})\delta_t\Big] \\ \frac{1}{\mathbf{N}} \sum_{\nu=1}^{\mathbf{N}} \Big[\sum_{l=1}^{\mathbf{L}} \Xi_{2,k,\nu}(\alpha_{t_l})\delta_t\Big] \\ \vdots \\ \frac{1}{\mathbf{N}} \sum_{\nu=1}^{\mathbf{N}} \Big[\sum_{l=1}^{\mathbf{L}} \Xi_{\mathbf{H},k,\nu}(\alpha_{t_l})\delta_t\Big] \end{bmatrix}$$

with

$$\begin{aligned}
\Xi_{h,k,l}(\alpha_{t_l}) = {} & X_{h,k,\nu}(t_l)^\top N(\alpha_{t_l}) X_{h,k,\nu}(t_l) + 2u_{h,k,\nu}(t_l)^\top S(\alpha_{t_l}) X_{h,k,\nu}(t_l) \\
& + u_{h,k,\nu}(t_l)^\top R(\alpha_{t_l}) u_{h,k,\nu}(t_l), \qquad h = 1, 2, ..., \mathbf{H},
\end{aligned}$$

where $k,\ j \in \mathcal{M}$. We denote

$$\Phi^{(i)} = \left[I_{\mathbf{K}} \otimes I_{xx},\ \ \Theta^{(i)},\ \ \zeta^{(i)}\right], \quad \eta = \left[\gamma_1,\ \ \gamma_2,\ \ \cdots,\ \ \gamma_{\mathbf{K}}\right]^\top,$$

where $I_{\mathbf{K}}$ is the $\mathbf{K} \times \mathbf{K}$ identity matrix, and the block matrices $\Theta^{(i)}$ and $\zeta^{(i)}$ are defined as

$$\Theta^{(i)} = \begin{bmatrix} \Theta_{11}^{(i)} & \Theta_{12}^{(i)} & \cdots & \Theta_{1\mathbf{K}}^{(i)} \\ \Theta_{21}^{(i)} & \Theta_{22}^{(i)} & \cdots & \Theta_{2\mathbf{K}}^{(i)} \\ \vdots & \vdots & \ddots & \vdots \\ \Theta_{\mathbf{K}1}^{(i)} & \Theta_{\mathbf{K}2}^{(i)} & \cdots & \Theta_{\mathbf{K}\mathbf{K}}^{(i)} \end{bmatrix}, \quad \zeta^{(i)} = \begin{bmatrix} \zeta_{11}^{(i)} & \zeta_{12}^{(i)} & \cdots & \zeta_{1\mathbf{K}}^{(i)} \\ \zeta_{21}^{(i)} & \zeta_{22}^{(i)} & \cdots & \zeta_{2\mathbf{K}}^{(i)} \\ \vdots & \vdots & \ddots & \vdots \\ \zeta_{\mathbf{K}1}^{(i)} & \zeta_{\mathbf{K}2}^{(i)} & \cdots & \zeta_{\mathbf{K}\mathbf{K}}^{(i)} \end{bmatrix}$$

with their block elements given by

$$\Theta_{kj}^{(i)} = \begin{cases} 2\delta_{ux,k}^{(i)} + 2I_{Xu}^{kk} - 2I_{XX}^{kk}(I_n \otimes \Gamma(\mathbf{Q}_k^{(i)})), & \text{if } k = j, \\ 2I_{Xu}^{kj} - 2I_{XX}^{kj}(I_n \otimes \Gamma(\mathbf{Q}_j^{(i)})), & \text{if } k \neq j, \end{cases}$$

and

$$\zeta_{kj}^{(i)} = \begin{cases} \delta_{uu,k}^{(i)} + I_{uu}^{kk} - I_{XX}^{kk}\bar{\Gamma}(\mathbf{Q}_k^{(i)}), & \text{if } k = j, \\ I_{uu}^{kj} - I_{XX}^{kj}\bar{\Gamma}(\mathbf{Q}_j^{(i)}), & \text{if } k \neq j. \end{cases}$$

We note that $\mathbf{H}$ is required to satisfy $\mathbf{H} \geq \frac{1}{2}[n(n+1) + 2nr + r(r+1)]$ in order to solve $Q_{kxx}^{(i+1)}$, $Q_{kux}^{(i+1)}$ and $Q_{kuu}^{(i+1)}$ with $k \in \mathcal{M}$.

Assumption 3.2. *In order to guarantee that Algorithm 2 can be implemented online at each step, $\Phi^{(i)}$ are required to have full column rank, we assume there exists $\mathbf{H}_0 > 0$ such that for all $\mathbf{H} \geq \mathbf{H}_0$,* $\operatorname{rank}(\Phi^{(i)}) = \frac{\mathbf{K}}{2}[n(n+1) + 2nr + r(r+1)]$.

Under Assumption 3.2, using least-squares method, we have

$$\left[\xi_{xx}^\top,\ \ \xi_{ux}^\top,\ \ \xi_{uu}^\top\right]^\top = \left({\Phi^{(i)}}^\top \Phi^{(i)}\right)^{-1} {\Phi^{(i)}}^\top \eta, \tag{47}$$

where

$$\begin{aligned} \xi_{xx} &= [vech(Q_{1xx}^{(i+1)}), vech(Q_{2xx}^{(i+1)}), \cdots, vech(Q_{\mathbf{K}xx}^{(i+1)})], \\ \xi_{ux} &= [vec(Q_{1ux}^{(i+1)}), vec(Q_{2ux}^{(i+1)}), \cdots, vec(Q_{\mathbf{K}ux}^{(i+1)})], \\ \xi_{uu} &= [vech(Q_{1uu}^{(i+1)}), vech(Q_{2uu}^{(i+1)}), \cdots, vech(Q_{\mathbf{K}uu}^{(i+1)})]. \end{aligned}$$

Therefore, (41) can be solved by (47). We also can obtain (30).

4 Numerical Example

In this section, we provide a simulation example to illustrate the proposed algorithms. We consider a two-dimensional linear stochastic system with two regimes. By setting $n = 2$, $r = 1$ and $\mathbf{K} = 2$, the parameter matrices of system (1) are given as follows

$$A_1 = \begin{bmatrix} -0.5 & 1.0 \\ 0.0 & -0.3 \end{bmatrix}, \quad B_1 = \begin{bmatrix} 0.0 \\ 1.0 \end{bmatrix}, \quad C_1 = \begin{bmatrix} 0.1 & 0.0 \\ 0.0 & 0.1 \end{bmatrix}, \quad D_1 = \begin{bmatrix} 0.05 \\ 0.05 \end{bmatrix},$$

$$A_2 = \begin{bmatrix} -0.4 & 0.8 \\ 0.0 & -0.6 \end{bmatrix}, \quad B_2 = \begin{bmatrix} 1.0 \\ 0.0 \end{bmatrix}, \quad C_2 = \begin{bmatrix} 0.1 & 0.05 \\ 0.05 & 0.1 \end{bmatrix}, \quad D_2 = \begin{bmatrix} 0.05 \\ 0.05 \end{bmatrix},$$

and the generator matrix is

$$\Pi = \begin{bmatrix} -2 & 2 \\ 2 & -2 \end{bmatrix}.$$

The coefficients in cost functional are given as

$$N_1 = \begin{bmatrix} 0.4 & 0.05 \\ 0.05 & 0.2 \end{bmatrix}, \quad S_1 = \begin{bmatrix} 0.1 & 0.03 \end{bmatrix}, \quad R_1 = 0.12,$$

$$N_2 = \begin{bmatrix} 0.3 & 0.04 \\ 0.04 & 0.5 \end{bmatrix}, \quad S_2 = \begin{bmatrix} 0.05 & 0.07 \end{bmatrix}, \quad R_2 = 0.08.$$

We find

$$\Gamma(\mathbf{Q}_1^{(0)}) = \begin{bmatrix} -4.41 & -0.69 \end{bmatrix}, \quad \Gamma(\mathbf{Q}_2^{(0)}) = \begin{bmatrix} -1.15 & 3.02 \end{bmatrix}$$

can drive the trajectories tend to zero when time s increases. Therefore, we choose them as the initial stabilizer. To satisfy Assumptions 3.1 and 3.2, we randomly choose more than $\frac{1}{2}[2 \times (2+1) + 2 \times 2 \times 1 + 1 \times (1+1)] = 6$ initial state vectors for each k. The components of each initial state vector are independently sampled from the uniform distribution on $[0, 10]$. Here we set $\mathbf{H} = 200$ and $\mathbf{N} = 50$ in this example.

(1) Algorithm 1 is implemented as in (29) and (30). At every iteration, we obtain $X^{(i)} = [X_{(1)}^{(i)}, X_{(2)}^{(i)}]^\top$ and run system (1) with $u_k^{(i)} = \Gamma(\mathbf{Q}_k^{(i)})x_h + \omega$, where the PE signal ω is a Gaussian white noise process with $\omega \sim \mathcal{N}(0, 8)$. Fig. 1a shows values of $\|\mathbf{Q}_k^{(i+1)} - \mathbf{Q}_k^{(i)}\|$ $(k = 1, 2)$ in each iteration, illustrating the variation in the differences between $\mathbf{Q}_k^{(i+1)}$ and $\mathbf{Q}_k^{(i)}$. We obtain that the sequence $\{\mathbf{Q}_k^{(i)}\}_{i=1}^{\infty}$ and $\{\Gamma(\mathbf{Q}_k^{(i)})\}_{i=1}^{\infty}$ $(k = 1, 2)$ converge separately to

$$\mathbf{Q}_1^* = \begin{bmatrix} 0.2826 & 0.2008 & 0.1206 \\ 0.2008 & 0.4668 & 0.1918 \\ 0.1206 & 0.1918 & 0.1209 \end{bmatrix}, \quad \mathbf{Q}_2^* = \begin{bmatrix} 0.4441 & 0.1622 & 0.1665 \\ 0.1622 & 0.3017 & 0.0771 \\ 0.1665 & 0.0771 & 0.0809 \end{bmatrix},$$

$$\Gamma(\mathbf{Q}_1^*) = \begin{bmatrix} -0.9972 & -1.5860 \end{bmatrix}, \quad \Gamma(\mathbf{Q}_2^*) = \begin{bmatrix} -2.0576 & -0.9527 \end{bmatrix}.$$

To check whether $\mathbf{Q}_k^*$ $(k = 1, 2)$ is the solution of CGAREs, we use (24), (25) and (26) to solve for the parameter P_k^*. We obtain

$$P_1^* = \begin{bmatrix} 0.1748 & 0.0196 \\ 0.0196 & 0.1609 \end{bmatrix}, \quad P_2^* = \begin{bmatrix} 0.1153 & 0.0043 \\ 0.0043 & 0.2418 \end{bmatrix}.$$

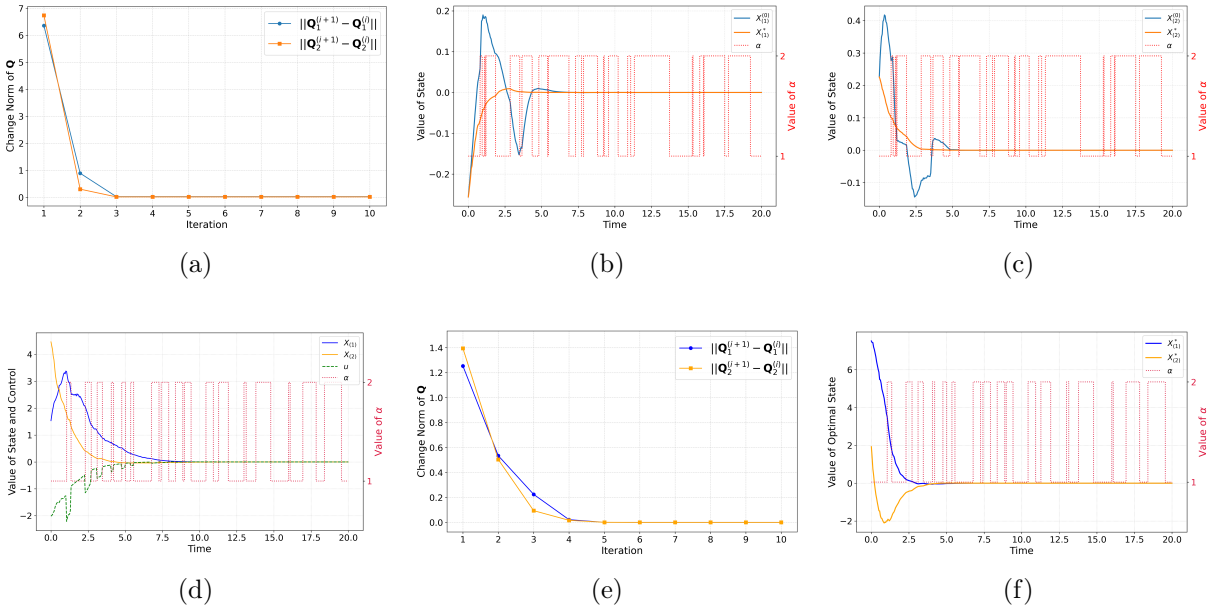
We define the left-hand sides of (7) as

$$\begin{aligned} \mathcal{R}(P_k) = {} & A_k^\top P_k + P_k A_k + C_k^\top P_k C_k + N_k + \sum_{j=1}^{\mathbf{K}} \pi_{kj} P_j \\ & - (P_k B_k + C_k^\top P_k D_k + S_k^\top)(R_k + D_k^\top P_k D_k)^{-1}(B_k^\top P_k + D_k^\top P_k C_k + S_k), \quad k \in \mathcal{M}. \end{aligned} \tag{48}$$

By substituting P_k^* into (48), we obtain $||\mathcal{R}(P_1^*)|| = 1.8990 \times 10^{-2}$ and $||\mathcal{R}(P_2^*)|| = 1.9535 \times 10^{-2}$. System (1) is simulated separately with $\Gamma(\mathbf{Q}_k^{(0)})$ and $\Gamma(\mathbf{Q}_k^*)$ $(k = 1, 2)$ to obtain the state trajectories $X^{(0)} = [X_{(1)}^{(0)}, X_{(2)}^{(0)}]^\top$ and $X^* = [X_{(1)}^*, X_{(2)}^*]^\top$. As shown in Fig. 1b and Fig. 1c, the optimal state trajectory X^* converges to zero much faster than the initial state trajectory $X^{(0)}$ under the Markov chain α.

(2) Algorithm 2 is implemented as in (41) and (30). The system state trajectory $X = [X_{(1)}, X_{(2)}]^\top$ is generated by simulating system (1) under a stabilizing control input u, as shown in Fig. 1d along with the Markov chain trajectory α. Fig. 1e shows values of $\|\mathbf{Q}_k^{(i+1)} - \mathbf{Q}_k^{(i)}\|$ $(k = 1, 2)$ in each iteration, illustrating the variation in the differences between $\mathbf{Q}_k^{(i+1)}$ and $\mathbf{Q}_k^{(i)}$. We obtain that the sequence $\{\mathbf{Q}_k^{(i)}\}_{i=1}^{\infty}$ and $\{\Gamma(\mathbf{Q}_k^{(i)})\}_{i=1}^{\infty}$ $(k = 1, 2)$ converge separately to

$$\mathbf{Q}_1^* = \begin{bmatrix} 0.2833 & 0.2022 & 0.1197 \\ 0.2022 & 0.4591 & 0.1886 \\ 0.1197 & 0.1886 & 0.1209 \end{bmatrix}, \quad \mathbf{Q}_2^* = \begin{bmatrix} 0.4430 & 0.1578 & 0.1651 \\ 0.1578 & 0.3089 & 0.0778 \\ 0.1651 & 0.0778 & 0.0809 \end{bmatrix},$$



(a) (b) (c) (d) (e) (f)

Figure 1: Simulation results. (a)-(c) show the results for Algorithm 1. (a) shows the convergence of $\mathbf{Q}_k$ for $k = 1, 2$. (b) compares $X_{(1)}^{(0)}$ and $X_{(1)}^*$. (c) compares $X_{(2)}^{(0)}$ and $X_{(2)}^*$. (d)-(f) show the results for Algorithm 2. (d) shows the trajectories of $X_{(1)}$, $X_{(2)}$, u, and α. (e) shows the convergence of $\mathbf{Q}_k$ for $k = 1, 2$. (f) shows the trajectories of $X_{(1)}^*$, $X_{(2)}^*$, and α.

$$\Gamma(\mathbf{Q}_1^*) = \begin{bmatrix} -0.9898 & -1.5712 \end{bmatrix}, \quad \Gamma(\mathbf{Q}_2^*) = \begin{bmatrix} -2.0406 & -0.9617 \end{bmatrix}.$$

To check whether $\mathbf{Q}_k^*$ $(k = 1, 2)$ is the solution of CGAREs, we use (24), (25) and (26) to solve for the parameter P_k^*. We obtain

$$P_1^* = \begin{bmatrix} 0.1731 & 0.0187 \\ 0.0187 & 0.1591 \end{bmatrix}, \quad P_2^* = \begin{bmatrix} 0.1139 & 0.0063 \\ 0.0063 & 0.2373 \end{bmatrix}.$$

By substituting P_k^* into (48), we obtain $||\mathcal{R}(P_1^*)|| = 1.0681 \times 10^{-2}$ and $||\mathcal{R}(P_2^*)|| = 1.3227 \times 10^{-2}$. System (1) is simulated with $\Gamma(\mathbf{Q}_k^*)$ $(k = 1, 2)$ to obtain the optimal state trajectory $X^* = [X_{(1)}^*, X_{(2)}^*]^\top$ as shown in Fig. 1f, along with the Markov chain trajectory α.

5 Conclusion

In this paper, we develop a Q-learning framework for infinite-horizon continuous-time stochastic LQ optimal control problems with regime switching, effectively eliminating the need for explicit system dynamics knowledge. By exploiting the intrinsic stability properties of the system, we solve the coupled problem in infinite-horizon and propose both on-policy and off-policy model-free algorithms that learn optimal control gains directly from state trajectory data. Theoretical analysis establishes the equivalence, stability, and convergence of the proposed algorithms, while numerical experiments validate their practical efficacy. Some future work can be extended to the control problem with nonlinear system, N players game with large population system, and so on.

References

[1] A. Al-Tamimi, F. L. Lewis, M. Abu-Khalaf, *Model-free Q-learning designs for linear discrete-time zero-sum games with application to H-infinity control*. Automatica, vol. 43, no.3, pp. 473-481, 2007.

[2] S. J. Bradtke, B. E. Ydstie, A. G. Barto, *Adaptive linear quadratic control using policy iteration*. In Proceedings of 1994 American Control Conference-ACC'94, vol. 3, pp. 3475-3479, 1994.

[3] P. Cheng, H. Wang, V. Stojanovic, F. Liu, S. He, K. Shi, *Dissipativity-based finite-time asynchronous output feedback control for wind turbine systems via a hidden Markov model*, International Journal of Systems Science, vol. 53, no. 15, 3177-3189, 2022.

[4] V. Dragan and T. Morozan, *The Linear quadratic optimization problems for a class of linear stochastic systems with multiplicative white noise and Markovian jumping*, IEEE Transactions on Automatic Control, vol. 49, no. 5, pp. 665-675, 2004.

[5] K. Du, Q. X. Meng, F. Zhang, *A Q-Learning Algorithm for Discrete-Time Linear-Quadratic Control with Random Parameters of Unknown Distribution: Convergence and Stabilization*, SIAM Journal on Control and Optimization, vol. 60, no. 4, pp. 1991-2015, 2022.

[6] S. Dong, L. Liu, G. Feng, M. Liu, Z. G. Wu, R. Zheng, *Cooperative output regulation quadratic control for discrete-time heterogeneous multiagent Markov jump systems*. IEEE Transactions on Cybernetics, vol. 52, no. 9, pp.9882-9892, 2021.

[7] M. D. Fragoso and J. Baczynski, *Optimal control for continuous-time LQ problems with infinite Markov jump parameters*, SIAM Journal on Control and Optimization, vol. 40, no. 1, pp.270-297, 2001.

[8] Z. Gajic and I. Borno, *Lyapunov Iteration Schemes for optimal control of jump linear systems at steady state*, IEEE Transactions on Automatic Control, vol. 40, pp. 1971-1975, 1995.

[9] L. E. Ghaoui and M. Ait Rami, *Robust state-feedback stabilization of jump linear systems via LMIs*, International Journal of Robust and Nonlinear Control, vol. 6, pp. 1015-1022, 1996.

[10] I.G. Ivanov, *On some iterations for optimal control of jump linear equations*, Nonlinear Analysis: Theory, Methods & Applications, vol. 69, pp. 4012-4024, 2008.

[11] B. Jiang, H. R. Karimi, S. Yang, C. Gao, Y. Kao, *Observer-based adaptive sliding mode control for nonlinear stochastic Markov jump systems via T-S fuzzy modeling: Applications to robot arm model*, IEEE Transactions on Industrial Electronics, vol. 68, no. 1, pp. 466-477, 2020.

[12] Y. W. Jia and X. Y. Zhou, *q-Learning in continuous time*. Journal of Machine Learning Research, vol. 24, no. 161, pp. 1-61, 2023.

[13] S. Kuppusamy, Y. H. Joo, H. S. Kim, *Asynchronous control for discrete-time hidden Markov jump power systems*, IEEE Transactions on Cybernetics, vol. 52, no. 9, 9943-9948, 2021.

[14] N. Li, S. Y. Lv, Z. Wu, J. Xiong, *Sufficient stochastic maximum principle for mean-field control problems with regime switching in an infinite horizon*. Science China Information Sciences, vol. 68, no. 11, 2025.

[15] N. Li, X. Li, Z. Q. Xu, *Policy Iteration Reinforcement Learning Method for Continuous-Time Linear-Quadratic Mean-Field Control Problems*, IEEE Transactions on Automatic Control, vol. 70, no. 4, 2025.

[16] Z. Li, Y. Zhang, A. G. Wu, *An inversion-free iterative algorithm for Riccati matrix equations in discrete-time Markov jump systems*. IEEE Transactions on Automatic Control, vol. 67, no. 9, pp. 4754-4761, 2022.

[17] J. Y. Lee, J. B. Park, Y. H. Choi, *Integral Q-learning and explorized policy iteration for adaptive optimal control of continuous-time linear systems*, Automatica, vol. 48, no. 11, pp. 2850-2859, 2012.

[18] X. Li, and X. Y. Zhou, *Indefinite stochastic LQ controls with Markovian jumps in a finite time horizon*. Communications in Information and Systems, vol. 2, no. 3, pp. 265-282, 2002.

[19] X. Li, X. Y. Zhou and M. Ait Rami, *Indefinite Stochastic LQ Control with Markovian Jumps in Infinite Time Horizon*, Journal of Global Optimization, vol. 27, pp. 149-175, 2003.

[20] J. J. Murray, C. J. Cox, G. G. Lendaris, R. Saeks, *Adaptive dynamic programming*, IEEE transactions on systems, man, and cybernetics, Part C (Applications and Reviews), vol. 32, no. 2, pp. 140-153, 2002.

[21] P. E. Protter, *Stochastic Integration and Differential equations*, Springer Berlin Heidelberg, 2005.

[22] M. Palanisamy, H. Modares, F. L. Lewis, M. Aurangzeb, *Continuous-Time Q-Learning for Infinite-Horizon Discounted Cost Linear Quadratic Regulator Problems*, IEEE Transactions on Cybernetics, vol. 45, no. 2, 2015.

[23] C. Possieri and M. Sassano, *Q-Learning for Continuous-Time Linear Systems: A Data-Driven Implementation of the Kleinman Algorithm*, IEEE Transactions on Systems, Man, and Cybernetics: Systems, vol. 52, no. 10, 2022.

[24] S. A. A. Rizvi and Z. L. Lin, *Output Feedback Q-Learning Control for the Discrete-Time Linear Quadratic Regulator Problem*, IEEE Transactions on Neural Networks and Learning Systems, vol. 30, no. 5, 2019.

[25] A. G. Wu, H. J. Sun, Y. Zhang, *A novel iterative algorithm for solving coupled Riccati equations*, Applied Mathematics and Computation, 364, Article 124645, 2020.

[26] T. Wang, H. G. Zhang, Y. H. Luo, *Stochastic linear quadratic optimal control for model-free discrete-time systems based on Q-learning algorithm*, Neurocomputing, vol. 312, 2018.

[27] W. J. Yang, W. Wang, J. J. Xu, J. W. Xia, *Q-Learning for Finite-Horizon Decentralized Linear-Quadratic Optimal Control Problem*, Optimal Control Applications and Methods, vol. 46, no. 6, pp. 2626-2635, 2025.

[28] X. Zhang, X. Li, J. Xiong, *Open-loop and closed-loop solvabilities for stochastic linear quadratic optimal control problems of Markovian regime switching system*, ESAIM: Control, Optimisation and Calculus of Variations, vol.27, no.69, 2021.

[29] X. Y. Zhou and G. Yin, *Markowitz's Mean-Variance Portfolio Selection With Regime Switching: A Continuous-Time Model*, SIAM Journal on Control and Optimization, vol. 42, no. 4, pp. 1466-1482, 2003.