Data-driven policy iteration algorithm for continuous-time stochastic linear-quadratic optimal control problems

Heng Zhang1 | Na Li2

1School of Control Science and Engineering, Shandong University, Jinan, China
2School of Statistics and Mathematics, Shandong University of Finance and Economics, Jinan, China

Abstract
This paper studies a continuous-time stochastic linear-quadratic (SLQ) optimal control problem on infinite-horizon. Combining the Kronecker product theory with an existing policy iteration algorithm, a data-driven policy iteration algorithm is proposed to solve the problem. In contrast to most existing methods that need all information of system coefficients, the proposed algorithm eliminates the requirement of three system matrices by utilizing data of a stochastic system. More specifically, this algorithm uses the collected data to iteratively approximate the optimal control and a solution of the stochastic algebraic Riccati equation (SARE) corresponding to the SLQ optimal control problem. The convergence analysis of the obtained algorithm is given rigorously, and a simulation example is provided to illustrate the effectiveness and applicability of the algorithm.

KEYWORDS
data-driven, policy iteration, stochastic algebraic Riccati equation, stochastic linear-quadratic optimal control problem

1 | INTRODUCTION
The linear-quadratic (LQ) optimal control problem originated by Kalman [1] is of great importance in the field of optimal control. The SLQ optimal control problem, pioneered by Wonham [2], has been widely considered in previous literature [3–9]. It is well known that a conventional way to work out the SLQ optimal control problem is to solve the corresponding SARE. However, since the SARE is always nonlinear, it is hard to get a clear expression of its analytical solution.

Over the past two decades, researchers have turned to investigate numerical solutions to the corresponding SARE of their problems. For example, Wu et al. [10] proposed two iterative algorithms to solve an SARE arising in SLQ optimal control problems subject to state-dependent noise. Feng and Anderson [11] developed an iterative strategy to study a class of state-perturbed SARE in LQ zero-sum games. Ait Rami and Zhou [12] adopted the theory of linear matrix inequality to solve an SARE arising in a continuous-time infinite-horizon indefinite SLQ problems. In literature mentioned above, all parameters of their systems should be used to solve the corresponding SARE. However, the system coefficients may not be completely known in the real world, especially in applications such as finance and engineering. Therefore, it is valuable to solve the SARE with partially model-free systems, that is, with partial information of the system coefficient matrices.

Recently, the techniques of adaptive dynamic programming (ADP) [13] and reinforcement learning [14]...
have been widely used to tackle control problems with model-free or partially model-free systems. For example, about deterministic problems, Zhao and Zhang [15] applied the method of Q-learning to solve a discrete-time optimal control problem with unknown system coefficients. Al-Tamimi et al. [16] obtained an optimal strategy for a class of linear model-free zero-sum games by the method of Q-learning. Liu et al. [17] adopted the ADP algorithm to tackle a discrete-time model-free nonlinear optimal control problem. Vrabie et al. [18] introduced a policy iteration algorithm to investigate partially model-free LQ optimal control problems. By virtue of ADP, Jiang and Jiang [19] studied a kind of deterministic continuous-time LQ problems with completely unknown system matrices. Liu et al. [20] used an ADP algorithm to tackle an unknown nonlinear control problem. Based on the ADP approach, Wang et al. [21] got an optimal output feedback control for model-free continuous-time nonlinear systems with actuator saturation. Chai et al. [22] established a novel deep reinforcement learning-based algorithm to work out a mobile robot control problem in unknown environment. For other advanced developments of learning-based control problems, please see [23, 24] and references therein. As for the stochastic case, Chen and Wang [25] obtained an optimal control for a class of model-free discrete-time stochastic systems by using the theory of ADP. Tan et al. [26] constructed a Q-learning method to tackle a delayed discrete-time SLQ problem with unknown system coefficients. Without knowing the information of drift term, Duncan et al. [27] obtained an adaptive linear-quadratic Gaussian control for a class of linear systems where the diffusion term does not rely on the control and state. Without using system matrix $A$ (see Equation 1 for the system dynamics), Li et al. [28] proposed a partially model-free policy iteration method to solve a kind of continuous-time SLQ problem on infinite horizon.

Inspired by the above work, especially [19, 28], this paper aims to tackle the infinite-horizon continuous-time SLQ optimal control problem with three unknown system parameters. With the help of ADP technique and Kronecker product representation, a novel data-driven algorithm is proposed. The main features of this paper are highlighted below. (i) A data-driven policy iteration algorithm is developed to solve the SLQ optimal control problem, where only partial system coefficient matrices are assumed known. Different from many existing literature such as [12], we use less system matrix information. (ii) In contrast to the algorithm obtained in Li et al. [28] which does not need the information of system matrix $A$ (see Equation 1), as a continuation, we further develop a novel design process, which does not require three system matrices $A$, $B$, and $C$, to tackle the problem. (iii) The proposed algorithm is applied to a numerical example. The corresponding numerical results demonstrate that our algorithm successfully finds a solution with high accuracy by using only partial system matrix information.

The rest of this paper is organized as follows. In Section 2, the SLQ optimal control problem is introduced, and some preliminaries are given. In Section 3, the data-driven algorithm along with its convergence analysis is developed in detail. Section 4 provides a numerical example to validate the algorithm. Finally, some concluding remarks and outlooks are given in Section 5.

### 1.1 Notations

We denote by $\mathbb{R}$ the set of real numbers and by $\mathbb{Z}^+$ the set of non-negative integers. The collection of all $p \times q$ real matrices is denoted by $\mathbb{R}^{p \times q}$. $\mathbb{R}^p$ represents the Euclidean space with dimension $p$, and $\| \cdot \|$ is the Euclidean norm. For simplicity, we denote zero matrix (or vector) by 0. $\text{diag}(l)$ denotes a square diagonal matrix with the elements of vector $l$ on the main diagonal. $M^T$ is the transpose of a vector or matrix $M$. $I_p \in \mathbb{R}^{p \times p}$ denotes the $p$-dimensional identity matrix. We use $S^p_0$, $S^p_+$, and $S^p_{++}$ to denote the collection of all symmetric matrices, positive semidefinite matrices, and positive definite matrices in $\mathbb{R}^{p \times p}$, respectively. Moreover, if a matrix $E \in S^p_{++}$ (resp. $E \in S^p_+$) is positive definite (resp. positive semidefinite), we usually write $E > 0$ (resp. $\geq 0$). If matrices $E \in S^p_0$, $F \in S^p_0$, then we write $E \geq F$ (resp. $E > F$) if $E - F \geq 0$ (resp. $E - F > 0$). $(\Omega, \mathcal{F}, \{\mathcal{F}_t\}_{t \geq 0}, P)$ is a filtered probability space that satisfies usual conditions, on which a one-dimensional standard Brownian motion $W(\cdot)$ is defined. We define space $L^2_p(\mathbb{R}^n)$ as

$$L^2_p(\mathbb{R}^n) \triangleq \left\{ u(\cdot) : [0, +\infty) \times \Omega \rightarrow \mathbb{R}^n \mid u(\cdot) \text{ is } \mathcal{F}_t\text{-adapted,} \right\}$$

Furthermore, $\varnothing$ denotes the Kronecker product. For any matrix $F$, $\text{vec}(F)$ is a vectorization map from the matrix $F$ into a column vector of proper size, which stacks the columns of $F$ on top of one another, that is,

$$\text{vec} \begin{bmatrix} f_{11} & f_{12} \\ f_{21} & f_{22} \\ f_{31} & f_{32} \end{bmatrix} \triangleq (f_{11}, f_{21}, f_{31}, f_{12}, f_{22}, f_{32})^T.$$
where $p_{ij}, i, j = 1, 2, 3, \ldots$, is the $(i,j)$th element of $P$ and $x_i, i = 1, 2, 3, \ldots$, is the $i$th element of $X$.

2 | PROBLEM FORMULATION AND SOME PRELIMINARIES

In this section, the SLQ optimal control problem and some preliminaries will be presented. Moreover, some assumptions are given to ensure the well-posedness of the problem.

Consider a stochastic linear system

$$
\begin{align*}
\frac{dX(s)}{dt} &= [AX(s) + Bv(s)]ds + [CX(s) + Dv(s)]dW(s), s \in [0, \infty), \\
X(0) &= x_0,
\end{align*}
$$

(1)

where $x_0 \in \mathbb{R}^n$, $X(\cdot) \in \mathbb{R}^n$ and $v(\cdot) \in \mathbb{R}^m$. The system coefficients $A$, $B$, $C$, and $D$ are given constant matrices of proper sizes. The performance index adopted in this paper is

$$
J(v(\cdot)) = \mathbb{E} \int_0^\infty [X(s)^TQX(s) + 2v(s)^TX(s) + v(s)^TRv(s)]ds,
$$

(2)

where $Q \in \mathbb{R}^{n \times n}$, $S \in \mathbb{R}^{m \times n}$ and $R \in \mathbb{R}^{m \times m}$ are constant matrices.

Now, we give the definition of $L^2$-stabilizability, which is indispensable for the well-posedness of our problems.

**Definition 1.** System (1) is said to be $L^2$-stabilizable if there exists a matrix $K \in \mathbb{R}^{m \times n}$ such that, for any initial state $x_0$, the solution of

$$
\begin{align*}
\frac{dX(s)}{dt} &= (A + BK)X(s)ds + (C + DK)X(s)dW(s), s \in [0, \infty), \\
X(0) &= x_0,
\end{align*}
$$

(3)

satisfies $\lim \mathbb{E}[X(s)^TX(s)] = 0$. In this case, the feedback control $v(\cdot) = KX(\cdot)$ is called stabilizing and the matrix $K$ is called a stabilizer of system (1).

**Assumption A1.** System (1) is $L^2$-stabilizable.

We define

$$
\mathcal{V}_{ad} \triangleq \left\{ v(\cdot) \in L^2_{ad}(\mathbb{R}^m) | v(\cdot) \text{ is stabilizing} \right\}
$$

as an admissible control set. The SLQ optimal control problem is given as follows:

**Problem (SLQ).** For given $x_0 \in \mathbb{R}^n$, we want to find an optimal control $v^*(\cdot) \in \mathcal{V}_{ad}$ such that

$$
J(v^*(\cdot)) = \inf_{v(\cdot) \in \mathcal{V}_{ad}} J(v(\cdot)).
$$

When $\inf_{v(\cdot) \in \mathcal{V}_{ad}} J(v(\cdot)) > -\infty$ is satisfied for any $x_0 \in \mathbb{R}^n$, Problem (SLQ) is well-posed. Moreover, the control $v^*(\cdot)$ that achieves $\inf_{v(\cdot) \in \mathcal{V}_{ad}} J(v(\cdot))$ is called optimal control, and the corresponding trajectory $X^*(\cdot)$ is called optimal trajectory.

Based on the main results of [28], we introduce the following assumption.

**Assumption A2.** $R > 0$ and $Q - S^TR^{-1}S > 0$.

Therefore, for any $x_0 \in \mathbb{R}^n$, Problem (SLQ) is a well-posed problem under Assumptions A1–A2.

3 | DATA-DRIVEN ALGORITHM FOR PROBLEM (SLQ)

In this section, we will introduce a data-driven algorithm to solve Problem (SLQ), which does not need coefficient matrices $A$, $B$, and $C$.

Before giving the algorithm, we first present an iterative method to solve Problem (SLQ). For the proof, please see Lemma 2.3 and Theorems 2.1–2.2 in [28].

**Lemma 1.** Suppose $K_0$ is a stabilizer of system (1) and $P_{i+1} \in \mathcal{S}^n_{++}, i = 0, 1, 2, \ldots$, are the solution of

$$
\begin{align*}
P_{i+1}(A + BK_1) + (A + BK_1)^TP_{i+1} \\
+ (C + DK_1)^TP_{i+1}(C + DK_1) \\
+ K_1^TRK_1 + S^TK_1 + K_1^TS + Q = 0,
\end{align*}
$$

(4)

where $K_{i+1}, i = 0, 1, 2, \ldots$, are updated by

$$
K_{i+1} = -(R + D^TP_{i+1}D)^{-1}(B^TP_{i+1} + D^TP_{i+1}C + S),
$$

(5)

then we have

(i) every element of $\{K_i\}_{i=1}^\infty$ is a stabilizer of system (1);

(ii) $P^* \geq P_{i+1} \geq P_i, i = 1, 2, 3, \ldots$;

(iii) $\lim_{i \to \infty} P_i = P^*, \lim_{i \to \infty} K_i = K^*$, where $K^* = -(R + D^TPD)^{-1}(D^TPC + B^TP^* + S)$ and $P^*$ is the solution to the SARE

$$
PA + A^TP + C^TPC + Q - (C^TPD + PB + S^T)(R + D^TPD)^{-1}(D^TPC + B^TP + S) = 0.
$$

(6)

Moreover, the optimal control of Problem (SLQ) is

$$
v^*(\cdot) = K^*X^*(\cdot).
$$

Though Lemma 1 presents an approximation method to solve SARE (6), solving $P_{i+1}$ and $K_{i+1}$ from Equations (4) and (5) require all information of the system coefficient.
matrices. As noted in the previous section, it is hard to obtain all information of the system parameters in the real world. In the sequel, we will propose a data-driven algorithm to solve them with partial knowledge of system (1).

In order to get our algorithm, system (1) is rewritten as

\[
\begin{align*}
\mathbf{d}(X(s)) &= [A_iX(s) + B_i(v(s) - K_iX(s))] dW(s), \quad s \in [0, \infty), \\
X(0) &= x_0,
\end{align*}
\]

where \( A_i \triangleq A + B_i K_i \) and \( C_i \triangleq C + D_i K_i \). Then (4) can be transformed to

\[
A_i^T P_{i+1} + P_{i+1} A_i + C_i^T P_{i+1} C_i + Q_i = 0, \tag{8}
\]

where \( Q_i \triangleq S_i^T K_i + K_i^T R_i K_i + K_i^T S_i + Q_i \).

Now, we give the next lemma to illustrate some relationship between \( P_{i+1} \) and \( K_{i+1} \), \( i = 0, 1, 2, \ldots \), generated from (4) and (5).

**Lemma 2.** For any \( K_i \), \( i = 0, 1, 2, \ldots, P_{i+1} \) and \( K_{i+1} \) generated from (4) and (5) satisfy the following equation

\[
\mathbb{E} \left[ X(t + \Delta t)^T P_{i+1} X(t + \Delta t) - X(t)^T P_{i+1} X(t) \right] \\
+ 2 \mathbb{E} \int_t^{t+\Delta t} \left( v(s) - K_i X(s) \right)^T M_{i+1} X(s) ds \\
- \mathbb{E} \int_t^{t+\Delta t} v(s)^T D_i^T P_{i+1} Dv(s) ds \\
+ \mathbb{E} \int_t^{t+\Delta t} X(s)^T K_i^T D_i^T P_{i+1} D K_i X(s) ds \\
= - \mathbb{E} \int_t^{t+\Delta t} X(s)^T Q_i X(s) ds \\
- 2 \mathbb{E} \int_t^{t+\Delta t} \left( v(s) - K_i X(s) \right)^T S X(s) ds,
\]

where \( M_{i+1} \triangleq (R + D_i^T P_{i+1} D) K_{i+1} \), \( X(\cdot) \) is the trajectory of system (7) and \( \Delta t \) is a predefined positive number.

**Proof.** Keeping in mind that \( X(\cdot) \) is the solution of system (7), then Ito’s formula implies

\[
d \left( X(s)^T P_{i+1} X(s) \right) = \mathbb{E} \left[ X(s)^T \left[ A_i^T P_{i+1} + P_{i+1} A_i + C_i^T P_{i+1} C_i \right] X(s) \right] \\
+ 2 \mathbb{E} \int_t^{t+\Delta t} \left( v(s) - K_i X(s) \right)^T B_i^T P_{i+1} Dv(s) ds \\
+ 2 \mathbb{E} \int_t^{t+\Delta t} \left( v(s) - K_i X(s) \right)^T D_i^T P_{i+1} Dv(s) ds,
\]

where the second equality is due to \( C_i \triangleq C + D_i K_i \). Moreover, it follows from (5) and (8) that

\[
B_i^T P_{i+1} + D_i^T P_{i+1} C_i = -(R + D_i^T P_{i+1} D) K_{i+1}.
\]

Inserting them into (10), one gets

\[
d \left( X(s)^T P_{i+1} X(s) \right) \\
= - \mathbb{E} \left[ X(s)^T Q_i X(s) \right] ds - \mathbb{E} \left( \left( v(s) - K_i X(s) \right)^T \right) \left\{ R + D_i^T P_{i+1} D \right\} K_{i+1} X(s) ds \\
- \mathbb{E} \left[ \left( v(s) - K_i X(s) \right)^T S X(s) \right] ds \quad \text{for all } s \geq t.
\]

Thus, integrating from \( t \) to \( t + \Delta t \) and taking expectation \( \mathbb{E} \) on both sides of (11), we get (9). The proof is completed.

By Kronecker product theory, if \( D, E, \) and \( F \) are matrices of proper sizes, \( P \) is any symmetric matrix, and \( \theta \) is any column vector of proper size, we have

\[
\text{vec}(DEF) = (F^T \otimes D) \text{vec}(E), \quad E^T \otimes F^T = (E \otimes F)^T.
\]

\[
\theta^T P \theta = \text{vec}(\theta^T P \theta) = (\theta^T \otimes \theta^T) \text{vec}(P) = \delta^T \text{vech}(P).
\]

Thus, in (9), noting that \( Dv(s) \) and \( DK_i X(s) \) are two column vectors and \( P_{i+1} \in \mathbb{S}^n \), one gets

\[
\mathbb{E} \left[ \left( v(s) - K_i X(s) \right)^T D_i^T P_{i+1} Dv(s) \right] = -\mathbb{E} \left( \left( v(s) - K_i X(s) \right)^T S X(s) \right) ds = \delta^T \text{vech}(P_{i+1})
\]

Similarly, from (9) and the above notations, for any \( l \in \mathbb{Z}^+ \), we derive

\[
\mathbb{E} \left[ \left( v(s) - K_l X(s) \right)^T D_i^T P_{i+1} Dv(s) \right] = -\mathbb{E} \left( \left( v(s) - K_l X(s) \right)^T S X(s) \right) ds = \delta^T \text{vech}(P_{i+1}).
\]
To rewrite (12) in a more compact form, we define matrices
\[\begin{align*}
\eta & \triangleq \mathbb{E} \left[ \int_{t_l}^{t_1} X(s) \otimes v^T(s) ds \right], \\
\delta_{x_0} & \triangleq \mathbb{E} \left[ \int_{t_l}^{t_1} X(s) \otimes X(s) ds \right], \\
\delta_{x} & \triangleq \mathbb{E} \left[ \int_{t_l}^{t_1} X(s) \otimes X(s) ds \right], \\
\delta_{v} & \triangleq \mathbb{E} \left[ \int_{t_l}^{t_1} \mathbb{E} \left[ \int_{t_l}^{t_1} Dv(s) ds \right] \right]. \\
\delta_{vk} & \triangleq \mathbb{E} \left[ \int_{t_l}^{t_1} Dv(s) ds \right].
\end{align*}\]

where \(0 \leq t_0 < t_1 < t_2 < \cdots < t_l\).

To rewrite (12) in a more compact form, we define matrices \(\eta_{xx} \in \mathbb{R}^{lx_{mn}(l+1)/2}, \delta_{x_0} \in \mathbb{R}^{lx_{mn}}, \delta_{x} \in \mathbb{R}^{lx_{mn}}, \delta_{v} \in \mathbb{R}^{lx_{mn}},\) and \(\delta_{vk} \in \mathbb{R}^{lx_{mn}(l+1)/2}\) as follows

\[\begin{align*}
\eta_{xx} & \triangleq \mathbb{E} \left[ x(t_1) - x(t_0), \ldots, x(t_l) - x(t_1), x(t_l) - x(t_{l-1}) \right]^T, \\
\delta_{x_0} & \triangleq \mathbb{E} \left[ \int_{t_l}^{t_1} X(s) \otimes X(s) ds \right], \\
\delta_{x} & \triangleq \mathbb{E} \left[ \int_{t_l}^{t_1} X(s) \otimes X(s) ds \right], \\
\delta_{v} & \triangleq \mathbb{E} \left[ \int_{t_l}^{t_1} \mathbb{E} \left[ \int_{t_l}^{t_1} Dv(s) ds \right] \right]. \\
\delta_{vk} & \triangleq \mathbb{E} \left[ \int_{t_l}^{t_1} Dv(s) ds \right].
\end{align*}\]

With these symbols, (12) implies

\[\begin{align*}
\mathbb{V} \left( \begin{array}{c}
\text{vec}(P_{l+1}) \\
\text{vec}(M_{l+1})
\end{array} \right) = \mathbb{I}_l,
\end{align*}\]

where \(\mathbb{V} \in \mathbb{R}^{\left( mn + \frac{n(n+1)}{2} \right) \times l} \) and \(\mathbb{I}_l \in \mathbb{R}^l\) are defined as

Multiplying \(\mathbb{V}_i^T\) on both sides of (13), we have

\[\begin{align*}
\mathbb{V}_i^T \mathbb{V}_i \left( \begin{array}{c}
\text{vec}(P_{l+1}) \\
\text{vec}(M_{l+1})
\end{array} \right) = \mathbb{V}_i^T \mathbb{I}_l, \quad \forall i \in \mathbb{Z}^+.
\end{align*}\]

If \(\mathbb{V}_i\), \(\forall i \in \mathbb{Z}^+\), has full column rank, (14) can be solved by

\[\begin{align*}
\left( \text{vec}(P_{l+1}) \right) = (\mathbb{V}_i^T \mathbb{V}_i)^{-1} \mathbb{V}_i ^T \mathbb{I}_l, \quad \forall i \in \mathbb{Z}^+.
\end{align*}\]

If \(\mathbb{V}_i\), \(\forall i \in \mathbb{Z}^+\), has full column rank, it follows from Lemma 2 and the above procedure that \(P_{l+1}\) and \(K_{l+1}\) generated from (4) and (5) satisfy (15). Note that (15) does not need coefficient matrices \(A, B, C\); thus, if we can solve \(P_{l+1}\) and \(K_{l+1}\), \(\forall i \in \mathbb{Z}^+\), from (15), we obtain a partially model-free algorithm.

Then, we give a rank condition in the next lemma, under which matrices \(\mathbb{V}_i\), \(\forall i \in \mathbb{Z}^+\), have full column rank.

**Lemma 3.** If there exists an \(l_0 \in \mathbb{Z}^+\), such that

\[\text{rank}((\delta_{x_0}, \delta_x)) = mn + \frac{n(n+1)}{2},\]

for all \(l \geq l_0\), then matrices \(\mathbb{V}_i\), \(\forall i \in \mathbb{Z}^+\), have full column rank.

**Proof.** Given \(i \in \mathbb{Z}^+\), this proof is equivalent to proving that

\[\mathbb{V}_i N = 0\]

has only the solution \(N = 0\).
Now, we prove it by contradiction. Assume \( N = [\text{vech}(F)^T, \text{vec}(G)^T]^T \in \mathbb{R}^{mn+n^{mn+1}} \) is a nonzero column vector, where \( \text{vech}(F) \in \mathbb{R}^{n^{mn+1}} \) and \( \text{vec}(G) \in \mathbb{R}^{mn} \). Applying Itô’s formula to \( X(s)^T F X(s) \), one gets

\[
\mathbb{E} \left[ X(t + \Delta t)^T F X(t + \Delta t) - X(t)^T F X(t) \right] = \mathbb{E} \int_t^{t+\Delta t} \left( A^T F + F A_i + C_i^T F C_i \right) X(s) ds + 2\mathbb{E} \int_t^{t+\Delta t} (v(s) - K_i X(s))^T B^T F X(s) ds + 2\mathbb{E} \int_t^{t+\Delta t} (v(s) - K_i X(s))^T D^T F C_i X(s) ds + \mathbb{E} \int_t^{t+\Delta t} (v(s) - K_i X(s))^T D^T F D(v(s) - K_i X(s)) ds,
\]

where \( X(\cdot) \) is the trajectory of system (7) with control \( v(\cdot) \).

By (9), (18), and the definition of \( \mathcal{V}_i \), we have

\[
\mathcal{V}_i N = \delta_{xx} \text{vech}(\mathcal{Y}) + \delta_{xv} \text{vec}(\mathcal{L}),
\]

where

\[
\mathcal{Y} = A^T F + F A_i + C_i^T F C_i - K_i^T (B^T F + D^T F C_i + G - D^T F D K_i) - (F B + C_i^T F D + G - K_i^T D^T F D) K_i,
\]

\[
\mathcal{L} = 2B^T F + 2D^T F C_i + 2G - 2D^T F D K_i.
\]

Noting that \( \mathcal{Y} \) is symmetric, we derive

\[
\delta_{xx} \text{vech}(\mathcal{Y}) = \delta_{xv} \text{vech}(\mathcal{L}),
\]

where \( \delta_\tau \in \mathbb{R}^{[n^{n+1}]} \) is defined as

\[
\delta_\tau = \begin{bmatrix} \int_{t_0}^{t_1} \bar{X}(s) ds, \int_{t_2}^{t_1} \bar{X}(s) ds, \cdots, \int_{t_{l-1}}^{t_l} \bar{X}(s) ds \end{bmatrix}^T.
\]

Then (17) and (19) imply

\[
[\delta_\tau, \delta_{xv}] \begin{pmatrix} \text{vech}(\mathcal{Y}) \\ \text{vec}(\mathcal{L}) \end{pmatrix} = 0.
\]

It follows from (20), (21), \( \mathcal{Y} = 0 \), and \( \mathcal{L} = 0 \) that

\[
A^T F + F A_i + C_i^T F C_i = 0. \tag{23}
\]

Further, since \( K_i, i \in \mathbb{Z}^+ \), is a stabilizer, we can easily see from Definition 1 that the trajectory of

\[
\begin{cases} 
\mathbb{E} \int_0^t x(s)^T F x(s) ds + C_i x(s) dw(s), \\
x(0) = x_0 \neq 0,
\end{cases}
\]

satisfies \( \lim_{t \to +\infty} \mathbb{E} [x(s)^T x(s)] = 0 \).

For any \( t > 0 \), applying Itô’s formula to \( d(x(s)^T F x(s)) \), we get

\[
\mathbb{E} [x(t)^T F x(t)] - x_0^T F x_0 = -\mathbb{E} \int_0^t x(s)^T (A^T F + F A_i + C_i^T F C_i) x(s) ds,
\]

where \( x(\cdot) \) is governed by (24).

Letting \( t \) go to positive infinity, (23) and (25) yield that \( x_0^T F x_0 = 0 \). Notice that \( x_0 \) can be any nonzero element in \( \mathbb{R}^n \); thus, we know \( F = 0 \). Then (21) and \( \mathcal{L} = 0 \) imply that \( G = 0 \). Consequently, this contradicts with \( N \neq 0 \), which completes the proof.

Using notations defined above, the data-driven algorithm is shown in Algorithm 1.

**Algorithm 1 Data-driven policy iteration algorithm.**

1. Let \( i = 0 \) and choose a stabilizer \( K_0 \) for system (1).
2. Employ \( v(\cdot) = K_0 X(\cdot) + e(\cdot) \) as the control and compute \( \eta_{xx}, \delta_{xx}, \delta_{xv} \) and \( \delta_{xw} \).
3. Calculate \( \delta_{\delta_{xx}} \).
4. Solve \( P_{i+1} \) and \( M_{i+1} \) from (15).
5. \( K_{i+1} = (R + D^T P_{i+1} D)^{-1} M_{i+1} \).
6. \( i \leftarrow i + 1 \).
7. **Until** \( |P_i - P_{i+1}| < \varepsilon \), where \( \varepsilon > 0 \) is a threshold.

**Remark 1.** In Algorithm 1, \( e(\cdot) \) is called the exploration noise. By utilizing the exploration noise, the persistent excitation condition [19, 29,30] can be met, and thus, rank condition (16) in Lemma 3 is satisfied. To tackle some practical ADP and machine learning problems, researchers usually choose exploration noises such as exponentially decreasing noise [31], the random noise generated from the normal distribution [30], the sum of sinusoidal signals [19], and random noise [16]. During the simulation in Section 4, the exploration noise is selected as a noise generated by Gaussian distribution.

Finally, we present the convergence analysis of Algorithm 1.
Theorem 1. When rank condition (16) is guaranteed, \( \{K_i\}_{i=1}^{\infty} \) and \( \{P_i\}_{i=1}^{\infty} \) defined in Algorithm 1 converge to \( K^* \) and \( P^* \), respectively.

Proof. Given \( K_i, \forall i \in \mathbb{Z}^+ \), it follows from Lemma 2 that \( (P_{i+1}, M_{i+1}) \) generated from iteration (4) and (5) satisfy (15). Moreover, it can be seen from Lemma 3 that (15) has a unique solution if rank condition (16) holds.

Therefore, if condition (16) is satisfied, the solution of equation (15) is equivalent to the solution of iterations (4) and (5). Then Lemma 1 implies the convergence of Algorithm 1. This completes the proof.

4 NUMERICAL EXAMPLE

In this section, we give a simulation example to illustrate the data-driven partially model-free algorithm. The system parameters of system (1) are given as follows

\[
A = \begin{bmatrix}
0 & -0.6 \\
0.6 & -0.3
\end{bmatrix}, \quad B = \begin{bmatrix} 0.05 \end{bmatrix},
\]

\[
C = \begin{bmatrix}
-0.02 & 0.03 \\
-0.05 & 0.02
\end{bmatrix}, \quad D = \begin{bmatrix} 0.001 & 0.03 \end{bmatrix},
\]

and the initial state is \( x_0 = [0.3, -0.2]^T \). The coefficients in cost functional (2) are chosen as \( Q = \text{diag}(2, 1) \), \( S = [0.1, 0.2] \) and \( R = 0.1 \).

Let \( K_0 = [0, 0] \) and \( \Delta t = 0.1 \) s; that is, the value of \( l \) in equation (12) is \( l = 4/\Delta t = 4/0.1 = 40 \). The state/input data collected within 4 s and the coefficient matrix \( D \) are used to implement Algorithm 1. The mathematical expectations are calculated by Monte Carlo with 10^4 samples. At the first iteration step, we set \( P_0 = 0 \) to check the stopping criterion \( \varepsilon = 10^{-5} \).

By applying the data-driven algorithm, we can obtain two approximation matrices \( \tilde{P}^* \) and \( \tilde{K}^* \) as shown below:

\[
\tilde{P}^* = \begin{bmatrix} 4.2947 & -0.9340 \\
-0.9340 & 2.7567
\end{bmatrix}, \quad \tilde{K}^* = \begin{bmatrix} -2.9482 & 2.1263 \end{bmatrix}.
\]

The convergence of these matrices is plotted in Figures 1 and 2, where \( P_{ij} \) and \( \tilde{P}_{ij}^* \), \( i, j = 1, 2 \), is the \((i, j)\)th element of \( P \) and \( \tilde{P}^* \), and \( K_i \) and \( \tilde{K}_i^* \), \( i = 1, 2 \), is the \(i\)th element of \( K \) and \( \tilde{K}^* \), respectively. Moreover, in order to check the errors between \( (\tilde{P}^*, \tilde{K}^*) \) and the values \((P^*, K^*)\) described by Lemma 1, we run the algorithm in Lemma 1 and obtain

\[ v(\cdot) = \tilde{K}^* X(\cdot). \]
\[
P^* = \begin{bmatrix} 4.2933 & -0.9317 \\ -0.9317 & 2.7631 \end{bmatrix}, \quad K^* = \begin{bmatrix} -2.9455 & 2.1284 \end{bmatrix}.
\]

It is easy to see from the above matrices that Algorithm 1 gives a solution pair close to \((P^*, K^*)\). Noteworthily, the proposed algorithm uses less system matrix information than the method in Lemma 1. Furthermore, an optimal trajectory governed by \(v(\cdot) = K^* X(\cdot)\) is plotted in Figure 3, which means that \(K^*\) is indeed a stabilizer. The above simulation results imply that the algorithm proposed in this paper may be an effective method in solving infinite-horizon SLQ optimal control problems with partial knowledge of system parameters.

5 | CONCLUSIONS

This article is concerned with a continuous-time SLQ optimal control problem. A data-driven policy iteration algorithm is developed to tackle the problem. By utilizing the collected data, this algorithm removes the requirement of three system matrices \(A, B,\) and \(C\). A detailed numerical study indicates that the developed algorithm may serve as a promising tool for solving the SLQ problem.

There are still many issues worth studying in the future. For example, topics such as designing data-driven algorithms for SLQ problems with model uncertainty [32], indefinite weighting matrix [33, 34], or mean-field [35] are very interesting. Applications of data-driven algorithms for autonomous vehicles [36] and spacecrafts [37] are also of vital importance. These problems are left for further investigation.

AUTHOR CONTRIBUTIONS

Heng Zhang: Methodology; writing—original draft. Na Li: Methodology; writing—review and editing.

ACKNOWLEDGMENTS

The work of Zhang was supported by the National Natural Science Foundation of China (Grant Nos. 61925306 and 61821004), the National Key R&D Program of China (Grant No. 2022YFA1006103), and the Natural Science Foundation of Shandong Province (Grant No. ZR2020ZD24). The work of Li was supported by the National Natural Science Foundation of China (Grant Nos. 12171279 and 11801317), the Natural Science Foundation of Shandong Province (Grant No. ZR2019MA013), and the Colleges and Universities Youth Innovation Technology Program of Shandong Province (Grant No. 2019KJ011). The authors also thank Professor Guangchen Wang and Miss Yu Wang for their insightful comments on improving the quality of this work.

CONFLICT OF INTEREST STATEMENT

The authors declare no potential conflict of interests.

ORCID

Heng Zhang https://orcid.org/0000-0003-2508-1137

REFERENCES

1. R. E. Kalman, Contributions to the theory of optimal control, Bol. Soc. Mat. Mex. 5 (1960), 102–119.
2. W. M. Wonham, On a matrix Riccati equation of stochastic control, SIAM J. Control 6 (1968), 681–697.
3. J. Yong and X. Zhou, Stochastic control: Hamiltonian systems and HJB equations, Springer-Verlag, New York, 1999.
4. Y. Huang, W. Zhang, and H. Zhang, Infinite horizon linear quadratic optimal control for discrete-time stochastic systems, Asian J. Control 10 (2008), 608–615.
5. J. Sun, X. Li, and J. Yong, Open-loop and closed-loop solvabilities for stochastic linear quadratic optimal control problems, SIAM J. Control Optim. 54 (2016), 2274–2308.
6. J. Sun and J. Yong, Stochastic linear quadratic optimal control problems in infinite horizon, Appl. Math. Optim. 78 (2018), 145–183.
7. Z. Yu, Linear-quadratic optimal control and nonzero-sum differential game of forward-backward stochastic system, Asian J. Control 14 (2012), 173–185.
8. T. Song and B. Liu, Discrete-time mean-field stochastic linear-quadratic optimal control problem with finite horizon, Asian J. Control 23 (2021), 979–989.
9. H. Zhang and Z. Yan, Backward stochastic optimal control with mixed deterministic controller and random controller and its applications in linear-quadratic control, Appl. Math. Comput. 369 (2020), 1–11.
10. A. Wu, H. Sun, and Y. Zhang, Two iterative algorithms for stochastic algebraic Riccati matrix equations, Appl. Math. Comput. 339 (2018), 410–421.
11. Y. Feng and B. D. O. Anderson, An iterative algorithm to solve state-perturbed stochastic algebraic Riccati equations in LQ zero-sum games, Syst. Control Lett. 59 (2010), 50–56.
12. M. Ait Rami and X. Zhou, Linear matrix inequalities, Riccati equations, and indefinite stochastic linear quadratic controls, IEEE Trans. Automat. Control 45 (2000), 1131–1143.
13. P. J. Werbos, Beyond regression: new tools for prediction and analysis in the behavioural sciences, Harvard University, Cambridge, MA, 1974.
14. R. S. Sutton and A. G. Barto, Reinforcement learning: an introduction, 2nd ed., MIT Press, Cambridge, 2018.
15. J. Zhao and C. Zhang, Finite-horizon optimal control of discrete-time linear systems with completely unknown dynamics using Q-learning, J. Ind. Manage. Optim. 17 (2021), 1471–1488.
16. A. Al-Tamimi, F. L. Lewis, and M. Abu-Khalaf, Model-free Q-learning designs for linear discrete-time zero-sum games with application to H-infinity control, Automatica 43 (2007), 473–481.
17. D. Liu, D. Wang, and X. Yang, An iterative adaptive dynamic programming algorithm for optimal control of unknown discrete-time nonlinear systems with constrained inputs, Info. Sci. 220 (2013), 331–342.
18. D. Vrabie, O. Pastravanu, M. Abu-Khalaf, and F. L. Lewis, Adaptive optimal control for continuous-time linear systems based on policy iteration, Automatica 45 (2009), 477–484.
19. Y. Jiang and Z. Jiang, Computational adaptive optimal control for continuous-time linear systems with completely unknown dynamics, Automatica 48 (2012), 2699–2704.
20. D. Liu, Y. Huang, D. Wang, and Q. Wei, Neural-network-observer-based optimal control for unknown nonlinear systems using adaptive dynamic programming, Internat. J. Control 86 (2013), 1554–1566.
21. T. Wang, S. Sui, and S. Tong, Data-based adaptive neural network optimal output feedback control for nonlinear systems with actuator saturation, Neurocomputing 247 (2017), 192–201.
22. R. Chai, H. Niu, J. Carrasco, F. Arvin, H. Yin, and B. Lennox, Design and experimental validation of deep reinforcement learning-based fast trajectory planning and control for mobile robot in unknown environment, IEEE Trans. Neural Netw. Learn. Syst. Early Access (2022).
23. R. Chai, A. Tsourdos, A. Savvaris, S. Chai, Y. Xia, and C. P. Chen, Design and implementation of deep neural network-based control for automatic parking maneuver process, IEEE Trans. Neural Netw. Learn. Syst. 33 (2022), 1400–1413.
24. R. Chai, D. Liu, T. Liu, A. Tsourdos, Y. Xia, and S. Chai, Deep learning-based trajectory planning and control for autonomous ground vehicle parking maneuver, IEEE Trans. Automat. Sci. Eng., 20 (2023), 1633–1647.
25. X. Chen and F. Wang, Neural-network-based stochastic linear quadratic optimal tracking control scheme for unknown discrete-time systems using adaptive dynamic programming, Control Theory Technol. 19 (2021), 315–327.
26. X. Tan, Y. Li, and Y. Liu, Stochastic linear quadratic optimal tracking control for discrete-time systems with delays based on Q-learning algorithm, AIMS Math. 8 (2023), 10249–10265.
27. T. E. Duncan, L. Guo, and B. Pasik-Duncan, Adaptive continuous-time linear quadratic Gaussian control, IEEE Trans. Automat. Control 44 (1999), 1653–1662.
28. N. Li, X. Li, J. Peng, and Z. Q. Xu, Stochastic linear quadratic optimal control problem: a reinforcement learning method, IEEE Trans. Automat. Control 67 (2022), 5009–5016.
29. S. J. Bradtke, B. E. Ydstie, and A. G. Barto, Adaptive linear quadratic control using policy iteration, Proceedings of the 1994 American control conference, Baltimore, MD USA, 1994, pp. 3475–3479.
30. S. J. Bradtke, Reinforcement learning applied to linear quadratic regulation, Proceedings of the 5th International Conference on Neural Information Processing Systems, San Francisco, CA, USA, 1992, pp. 295–302.
31. K. G. Vamvoudakis and F. L. Lewis, Multi-player non-zero-sum games: online adaptive learning solution of coupled Hamilton-Jacobi-equations, Automatica 47 (2011), 1556–1569.
32. R. Chai, A. Tsourdos, H. Gao, S. Chai, and Y. Xia, Attitude tracking control for reentry vehicles using centralized robust model predictive control, Automatica 145 (2022), 110561.
33. C. Peng and W. Zhang, Multicriteria optimization problems of finite horizon stochastic cooperative linear-quadratic difference games, Sci. China Inf. Sci. 65 (2022), 172203.
34. W. Zhang and C. Peng, Indefinite mean-field stochastic cooperative linear-quadratic dynamic difference game with its application to the network security model, IEEE Trans. Cybern. 52 (2022), 11805–11818.
35. C. Peng and W. Zhang, Pareto optimality in infinite horizon mean-field stochastic cooperative linear-quadratic difference games, IEEE Trans. Automat. Control 68 (2023), 4113–4126.
36. R. Chai, A. Tsourdos, S. Chai, Y. Xia, A. Savvaris, and C. P. Chen, Multi-phase overtaking maneuver planning for autonomous ground vehicles via a desensitized trajectory optimization approach, IEEE Trans. Ind. Informat. 19 (2023), 74–87.
37. R. Chai, A. Tsourdos, H. Gao, Y. Xia, and S. Chai, Dual-loop tube-based robust model predictive attitude tracking control for spacecraft with system constraints and additive disturbances, IEEE Trans. Ind. Electron. 69 (2022), 4022–4033.

AUTHOR BIOGRAPHIES

Heng Zhang received the BS degree in mathematics and applied mathematics from Shandong Normal University, Jinan, China, in 2019. He is currently working toward the PhD degree at the School of Control Science and Engineering, Shandong University, Jinan, China. His current research interests include stochastic control, reinforcement learning, adaptive dynamic programming, and their applications.

Na Li received her BS degree in mathematics and applied mathematics from the Department of Mathematics, Linyi University, Linyi, China, in 2005. She received the MSc degree in mathematics from the Department of Mathematics, Zhejiang University, Hangzhou, China, in 2007, and the PhD degree in probability theory and mathematical statistics from the School of Mathematics, Shandong University, Jinan, China, in 2015. She is currently a professor in the School of Statistics and Mathematics at Shandong University of Finance and Economics. Her research interests include stochastic control theory, forward and backward stochastic differential equation, reinforcement learning and mathematical finance. She has published papers in journals such as SIAM Journal on Control and Optimization, IEEE Transactions on Automatic Control, and Automatica.

How to cite this article: H. Zhang and N. Li, Data-driven policy iteration algorithm for continuous-time stochastic linear-quadratic optimal control problems, Asian J. Control 26 (2024), 481–489, DOI 10.1002/asjc.3223.