Model-Free λ-Policy Iteration for Discrete-Time Linear Quadratic Regulation

Yongliang Yang, *Member, IEEE*, Bahare Kiumarsi, *Member, IEEE*, Hamidreza Modares, *Senior Member, IEEE*, and Chengzhong Xu, *Fellow, IEEE*

***Abstract*— This article presents a model-free λ-policy iteration (λ-PI) for the discrete-time linear quadratic regulation (LQR) problem. To solve the algebraic Riccati equation arising from solving the LQR in an iterative manner, we define two novel matrix operators, named the weighted Bellman operator and the composite Bellman operator. Then, the λ-PI algorithm is first designed as a recursion with the weighted Bellman operator, and its equivalent formulation as a fixed-point iteration with the composite Bellman operator is shown. The contraction and monotonic properties of the composite Bellman operator guarantee the convergence of the λ-PI algorithm. In contrast to the PI algorithm, the λ-PI does not require an admissible initial policy, and the convergence rate outperforms the value iteration (VI) algorithm. Model-free extension of the λ-PI algorithm is developed using the off-policy reinforcement learning technique. It is also shown that the off-policy variants of the λ-PI algorithm are robust against the probing noise. Finally, simulation examples are conducted to validate the efficacy of the λ-PI algorithm.**

***Index Terms*— Algebraic Riccati equation (ARE), fixed-point theory, off-policy reinforcement learning (RL), optimal control.**

I. Introduction

DYNAMIC programming (DP) theory, developed by Bellman [1], is both a mathematical programming approach and an algorithmic coding method that simplifies a complex problem by breaking it down into subproblems in a recursive manner. DP has found numerous applications in many fields, ranging from investment planning, scheduling problems, and routing problems, to name a few. In control theory, DP technique, resting on Bellman's principle of optimality, leads to solving the Hamilton–Jacobi–Bellman partial differential equation, which provides the necessary and sufficient condition to the optimal feedback control policy [2]. This is in contrast to the Pontryagin maximum principle, which only provides the necessary condition of the optimal feedback law and generally results in open-loop control solutions [3]. It is well known that the exact DP method suffers from the "curse of dimensionality" and might fail to produce the solution in polynomial time due to the computational limitation for high-dimensional problems [4]. To facilitate the decision-making problem, approximate/adaptive DP (ADP) algorithms are developed to target large-scale systems for which exact methods become infeasible. Typical iterative approximation approaches, including value iteration (VI) [5]–[7] and policy iteration (PI) [8]–[10], have been widely developed for various optimal control problems, including optimal regulation problem [11]–[14], differential games [15], [16], stochastic systems [17], and tracking problem [18], to name a few. However, for PI, the initial iterative policy is required to be admissible, which might not be available when the system dynamics is completely unknown. On the other hand, VI does not require an initial stabilizing policy and can be implemented by simple recursion [19], which makes the convergence of VI slower than PI [20], [21]. λ-policy iteration (λ-PI) algorithm, a balance between VI and PI, brings the best of both worlds together and offers advantages of relaxed initial condition requirement in contrast to PI algorithm and faster convergence speed compared to VI algorithm. The existing λ-PI algorithm is designed for decision-making optimization for Markov decision process, of which the complete system knowledge is required [22]. To the authors' knowledge, the λ-PI algorithm has not been leveraged to solve the linear quadratic regulation (LQR) problem for the discrete-time linear time-invariant (LTI) system. For the LQR problem of LTI systems in discrete time, in this article, we define the weighted Bellman operator and the composite Bellman operator to reformulate the λ-PI algorithm as a fixed-point iteration. It is shown that the composite Bellman operator has contraction and monotonic properties. On this basis, the iterative learning performance convergence is guaranteed.

Manuscript received 25 November 2020; revised 9 May 2021; accepted 17 July 2021. Date of publication 11 August 2021; date of current version 6 February 2023. The work of Chengzhong Xu was supported by the National Key Research and Development Program of China under Grant 2019YFB2102100. This work was supported in part by the National Natural Science Foundation of China under Grant 61903028, in part by the Science and Technology Development Fund (FDCT), Macao, under Grant 0015/2019/AKP, in part by the University of Macau (UM) Macao Talent Programme under Grant UMMTP-2019-02, and in part by the Guangdong-Hong Kong-Macao Joint Laboratory of Human-Machine Intelligence-Synergy Systems under Grant 2019B121205007. *(Corresponding author: Chengzhong Xu.)*

Yongliang Yang was with the State Key Laboratory of IoTSC, University of Macau, Taipa, Macau 999078, China. He is now with the Key Laboratory of Knowledge Automation for Industrial Processes of Ministry of Education, School of Automation and Electrical Engineering, University of Science and Technology Beijing, Beijing 100083, China (e-mail: yangyongliang@ieee.org).

Bahare Kiumarsi is with the Department of Electrical and Computer Engineering, Michigan State University, East Lansing, MI 48824 USA (e-mail: kiumarsi@msu.edu).

Hamidreza Modares is with the Department of Mechanical Engineering, Michigan State University, East Lansing, MI 48824 USA (e-mail: modaresh@msu.edu).

Chengzhong Xu is with the State Key Laboratory of IoTSC, University of Macau, Taipa, Macau (e-mail: czxu@um.edu.mo).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TNNLS.2021.3098985.

Digital Object Identifier 10.1109/TNNLS.2021.3098985

Existing results on λ-PI require the model knowledge for both value function update and policy improvement [22]. For real-world applications, the exact model of the system dynamics is difficult to be obtained. Neural-network-based identifiers have been combined with the actor–critic reinforcement learning (RL) algorithm to approximate the unknown system dynamics [23]. However, a small bias in the identified dynamics can have a huge bias in the resulting control policy and, thus, the control performance. Off-policy RL methods [24], a data-driven technique inspired by biological systems, have been introduced to implement model-free optimal control design recently for both continuous-time [25], [26] and discrete-time systems [27], [28]. In the off-policy RL method, online learning consists of two phases: data collection phase and learning phase. A behavior policy for data collection phase is applied to the system that can be different from the policy under evaluation at each iteration [29], which brings more flexibility in the controller implementation and learning procedure. The collected data are then leveraged in the learning phase to learn about as many policies as required. Applications of off-policy RL algorithms have been successfully made in the H_2 optimal regulation problem [25], [30], H_∞ control problem [27], robust stabilization problem [28], [31], [32], and intermittent feedback design [33], to name a few. In this article, a novel off-policy variant of the λ-PI algorithm is presented using only online collected data, where the complete knowledge of system dynamics is no longer required. In addition, performance convergence to the optimum is investigated using the fixed-point theory.

In this article, an iterative λ-PI algorithm is developed for solving the LQR problem of discrete-time linear systems with completely unknown dynamics. First, the matrix Bellman operator and Riccati operator are defined, which are fundamental in the convergence and performance analysis. Then, the λ-PI algorithm is formulated as the fixed-point iteration with a composite Bellman operator, which is shown to be a contraction mapping. Convergence of the λ-PI algorithm is discussed within the fixed-point framework in detail. For the implementation purpose, on- and off-policy equivalent variants of the λ-PI algorithm are derived. The main contributions of this article can be summarized as follows.

1) λ-PI algorithm is developed to solve the algebraic Riccati equation (ARE) for the discrete-time LQR problem for the first time in this article. Compared with the PI and VI algorithms in terms of the convergence rate and control performance, it is shown that the λ-PI algorithm is an intermediate between the PI and VI algorithms.
2) On- and off-policy variants of the λ-PI algorithm are developed with comparison in terms of robustness against the probing noise. It is shown that the off-policy RL-based λ-PI implementation does not have estimation bias in the presence of probing noise as opposed to the on-policy variants.

The remainder of this article is as follows. In Section II, the optimal control problem of the linear dynamical system in discrete time is presented, where the Bellman and Riccati matrix operators are defined for policy performance analysis. In Section II-C, the application of the Bellman and Riccati matrix operators in VI and PI algorithms is briefly reviewed. In Section III, the λ-PI algorithm solving the discrete-time linear quadratic regulator problem is developed with theoretical discussions on the convergence and performance analysis. In Section IV, the off-policy RL-based data-driven implementation variants are developed with discussion on the robustness against the probing noise in the behavior noise. Simulation results are shown in Section V and concluding remarks are made in Section VI.

Notations: For symmetric matrices P_1 and P_2, $P_1 \succ (\succeq) P_2$ if $P_1 - P_2$ is positive (semi)definite. For a square matrix M, the spectral radius of $M \in \mathbb{R}^{n\times n}$ is the largest absolute value of its eigenvalues $\lambda(M)$ for $i = 1, \ldots, n$ and denoted as $\rho(M) = \max_{i=1,\ldots,n}\{|\lambda_i(M)|\}$. For arbitrary $M \in \mathbb{R}^{n\times n}$, the matrix M is Schur stable if $\|M\|_2 < 1$, where $\|M\|_2$ represents the spectral norm for matrix M.

II. Problem Formulation

Consider the following linear discrete-time dynamical system:

$$x_{k+1} = Ax_k + Bu_k \tag{1}$$

where $x_k \in \mathbb{R}^n$ and $u_k \in \mathbb{R}^m$ are the state and control input, respectively, at time k.

Assumption 1: The pair (A, B) is Schur stabilizable. □

From the system dynamics (1), one can observe that the state transition depends on the control input u_k, which is further generated by a specific feedback control policy $\mu(\cdot) : \mathbb{R}^n \to \mathbb{R}^m$, i.e., $u_k = \mu(x_k)$. Therefore, as shown later in Section IV, different policies $\mu^{\text{on}}(x_k)$ and $\mu^{\text{off}}(x_k)$ might result in different state transition even starting from the same state x_k, i.e.,

$$\begin{aligned} x_{k+1}^{\text{on}} &= Ax_k + B\mu^{\text{on}}(x_k) \\ x_{k+1}^{\text{off}} &= Ax_k + B\mu^{\text{off}}(x_k). \end{aligned}$$

A. Riccati Equation and Riccati Operator

Consider the infinite horizon performance index

$$J(\mu(\cdot); x_k) = \sum_{j=k}^{\infty} r(x_j, u_j) \tag{2}$$

where $u_k = \mu(x_k)$ and $r(x_k, u_k) = x_k^{\mathrm{T}} Q x_k + u_k^{\mathrm{T}} R u_k$ with $Q \succeq 0$ and $R \succ 0$. For admissible policy $\mu(x_k)$, the value function $V(x_k)$, which is equivalent to the performance index $J(\mu(\cdot); x_k)$ [9], [34], can be obtained by solving the Bellman equation as

$$V(x_k) = x_k^{\mathrm{T}} Q x_k + \mu^{\mathrm{T}}(x_k) R \mu(x_k) + V(x_{k+1}) \tag{3}$$

for $x_k \in \mathbb{R}^n$. Based on the linear quadratic optimal control theory [2], for arbitrary admissible linear feedback policy $\mu(x_k) = Kx_k$, the value function is quadratic in the state, i.e., $V(x_k) = x_k^{\mathrm{T}} P x_k$. Then, the Bellman equation (3) can be rewritten as [35], [36]

$$\begin{aligned} x_k^{\mathrm{T}} P x_k = {} & x_k^{\mathrm{T}} Q x_k + x_k^{\mathrm{T}} K^{\mathrm{T}} R K x_k \\ & + x_k^{\mathrm{T}} (A + BK)^{\mathrm{T}} P (A + BK) x_k. \end{aligned} \tag{4}$$

Define the Hamiltonian as [2]

$$\mathcal{H}(x_k, u_k, V(x_{k+1})) = x_k^{\mathrm{T}} Q x_k + u_k^{\mathrm{T}} R u_k + V(x_{k+1}). \quad (5)$$

Based on Bellman's optimality principle [2], the optimal value function is defined as

$$V^*(x_k) = \min_{u_k} \mathcal{H}\big(x_k, u_k, V^*(x_{k+1})\big). \quad (6)$$

Based on the stationarity condition for the optimality [2], i.e.,

$$\frac{\partial \mathcal{H}(x_k, u_k, V^*(x_{k+1}))}{\partial u_k} = 0$$

the optimal feedback policy can be determined as

$$\mu^*(x_k) = \arg\min_{u_k} \mathcal{H}\big(x_k, u_k, V^*(x_{k+1})\big) = K x_k$$

with the optimal gain

$$K = -\big(R + B^{\mathrm{T}} P B\big)^{-1} B^{\mathrm{T}} P A.$$

Inserting the optimal control into the Bellman equation (4) yields the necessary and sufficient condition to the optimal control [2]

$$P = A^{\mathrm{T}} P A + Q - A^{\mathrm{T}} P B\big(R + B^{\mathrm{T}} P B\big)^{-1} B^{\mathrm{T}} P A \quad (7)$$

where P is the solution to the discrete-time ARE (7).

Definition 1 (Riccati Operator): Define the Riccati operator $\mathrm{Ric}(\cdot)$ on the value function $V(\cdot)$ as

$$\mathcal{R}(V(\cdot); x_k) = \frac{\partial V(x_{k+1})}{\partial x_{k+1}} + x_k^{\mathrm{T}} Q x_k - \frac{1}{4}\left[\frac{\partial V(x_{k+1})}{\partial x_{k+1}}\right]^{\mathrm{T}} R^{-1} B R^{-1} \frac{\partial V(x_{k+1})}{\partial x_{k+1}}.$$

Using the linear quadratic theory, the Riccati operator can be denoted as the following matrix operator, i.e.,

$$\mathcal{R}(P) \triangleq A^{\mathrm{T}} P A + Q - A^{\mathrm{T}} P B\big(R + B^{\mathrm{T}} P B\big)^{-1} B^{\mathrm{T}} P A \quad (8)$$

where P is a positive semidefinite matrix. □

Based on Definition 1, the ARE (7) can be equivalently expressed as the following fixed-point equation:

$$P = \mathcal{R}(P). \quad (9)$$

B. Lyapunov Equation and Bellman Operator

Definition 2 (Bellman Operator): Define the Bellman operator of the value function as

$$\mathcal{B}_\mu(V(x_k)) = \mu^{\mathrm{T}}(x_k) R \mu(x_k) + x_k^{\mathrm{T}} Q x_k + V(x_{k+1})$$

where $x_{k+1} = A x_k + B\mu(x_k)$. Based on the linear optimal control theory, the Bellman operator for linear systems can be expressed as

$$\begin{aligned} \mathcal{B}_K(P) &= K^{\mathrm{T}} R K + Q + (A + BK)^{\mathrm{T}} P (A + BK), \\ &= K^{\mathrm{T}} R K + Q + A_K^{\mathrm{T}} P A_K \\ A_K &= A + BK \end{aligned} \quad (10)$$

where $A_K = A + BK$ and $\mu(x_k) = K x_k$. □

Lemma 1: Consider the following set as:

$$\Omega \triangleq \left\{ P \succ 0 \,\middle|\, \rho\Big(A - B\big(R + B^{\mathrm{T}} P B\big)^{-1} B^{\mathrm{T}} P A\Big) < 1 \right\}. \quad (11)$$

Then, $P \in \Omega$ provided that

$$\min_K \mathcal{H}(x_k, u_k, V(x_{k+1})) \le V(x_k) \ \forall x_k \in \mathbb{R}^n \quad (12)$$

where $u_k = K x_k$, $V(x_k) = x_k^{\mathrm{T}} P x_k$, and $x_{k+1} = A x_k + B u_k$.

Proof: Based on the optimal control theory [2], one has

$$\arg\min_K \mathcal{H}(x_k, u_k, V(x_{k+1}))) = -\big(R + B^{\mathrm{T}} P B\big)^{-1} B^{\mathrm{T}} P A. \quad (13)$$

Then, condition (12) implies that

$$K^{\mathrm{T}} R K + Q + (A + BK)^{\mathrm{T}} P (A + BK) \le P. \quad (14)$$

for $K = -\big(R + B^{\mathrm{T}} P B\big)^{-1} B^{\mathrm{T}} P A$. In the following, we consider two cases.

1) If (12) holds as an equation, then (14) also holds as an equation. Inserting $K = -\big(R + B^{\mathrm{T}} P B\big)^{-1} B^{\mathrm{T}} P A$ into (14) yields the ARE (7). From optimal control theory, under Assumption 1, $P \in \Omega$ with P satisfying ARE (7).
2) Consider the case when (12) holds as a strict inequality with "<." Then, (14) also holds with "<." This implies that the closed-loop system

$$x_{k+1} = \Big[A - B\big(R + B^{\mathrm{T}} P B\big)^{-1} B^{\mathrm{T}} P A\Big] x_k \quad (15)$$

is Schur stable with $V(x_k) = x_k^{\mathrm{T}} P x_k$ being the Lyapunov function. Finally, from the stability theory for the discrete-time dynamical linear system, one can conclude that $\rho\Big(A - B\big(R + B^{\mathrm{T}} P B\big)^{-1} B^{\mathrm{T}} P A\Big) < 1$.

To summarize, the conclusion of Lemma 1 holds. ■

Lemma 2: Under Assumption 1, the Riccati operator defined in (8) has the following properties.

1) *Contraction Property:* The Riccati operator is a contract mapping in the sense that

$$\|\mathcal{R}(P_1) - \mathcal{R}(P_2)\| < \|P_1 - P_2\| \ \forall P_1, P_2 \in \Omega \quad (16)$$

where Ω is defined in (11).
2) *Monotonic Property:* The Riccati operator is monotonic in the sense that

$$\mathcal{R}(P_1) \succ (\succeq) \mathcal{R}(P_2), \quad \text{if } P_1 \succ (\succeq) P_2 \quad (17)$$

where $P_1, P_2 \in \Omega$.

Proof: First, based on the matrix inversion lemma [37], for positive definite matrix P, one has

$$\big(P^{-1} + B R^{-1} B^{\mathrm{T}}\big)^{-1} = P - P B\big(R + B^{\mathrm{T}} P B\big)^{-1} B^{\mathrm{T}} P. \quad (18)$$

Then, based on the linear quadratic optimal control theory [38], the Riccati operator can be rewritten as

$$\mathcal{R}(P) = A^{\mathrm{T}}\big(P^{-1} + B R^{-1} B^{\mathrm{T}}\big)^{-1} A + Q. \quad (19)$$

The term $R(P_1) - R(P_2)$ can be written as (20) (see the bottom of the next page). Note that $P_j \in \Omega$, for $j = 1$ and 2. Then, from Lemma 1, one has

$$\Big\| A - B\big(R + B^{\mathrm{T}} P_j B\big)^{-1} B^{\mathrm{T}} P_j A \Big\| < 1, \ j = 1, 2. \quad (21)$$

Therefore, combining (20) and (21) yields (16).

2) See [39]. This completes the proof. ■

C. Matrix Operator in Iterative ADP

1) Policy Evaluation: The policy evaluation aims to find the value function corresponding to a given admissible policy, which is widely discussed in [9] and [40]. Based Definition 2, the policy evaluation is a fixed-point matrix equation of P, i.e., $P = \mathcal{B}_K(P)$. This fact is discussed as follows.

Lemma 3 (Fixed-Point Iteration With the Bellman Operator): Consider the admissible policy $\mu(x_k) = Kx_k$ for $\forall x_k \in \mathbb{R}$, and the kernel matrix sequence $\{P^j\}_{j=1}^{\infty}$ generated by $P^{j+1} = \mathcal{B}_K(P^j)$ with positive definite initial kernel matrix P^0. Then, the kernel matrix sequence $\{P^j\}_{j=1}^{\infty}$ converges to the solution to the Bellman equation (4).

Proof: From Definition 2, for $n = 1, 2, \ldots$, we define $\mathcal{B}_K^{(n+1)}(P) = \mathcal{B}_K\left(\mathcal{B}_K^{(n)}(P)\right)$. Then, the recursion of the Bellman operator can be expressed as $P^{j+1} = \mathcal{B}_K(P^j) = \mathcal{B}_K^{(j+1)}(P^0)$. Note that the policy $\mu(x_k) = Kx_k$ is admissible, and then, $A_K = A + BK$ is Schur stable and one can obtain $\lim_{n\to\infty} x_k^{\mathrm{T}}(A_K^{\mathrm{T}})^n P^0 (A_K)^n x_k = 0$ and $\lim_{n\to\infty} x_k^{\mathrm{T}} P^n x_k = \sum_{j=k}^{\infty}\left[x_j^{\mathrm{T}} Q x_j + u_j^{\mathrm{T}} R u_j\right]$, $\forall x_k \in \mathbb{R}^n$. Therefore, P^{∞} satisfies

$$\begin{aligned} x_k^{\mathrm{T}} P^{\infty} x_k &= x_k^{\mathrm{T}} Q x_k + u_k^{\mathrm{T}} R u_k + \sum_{j=k+1}^{\infty}\left[x_j^{\mathrm{T}} Q x_j + u_j^{\mathrm{T}} R u_j\right] \\ &= x_k^{\mathrm{T}} Q x_k + u_k^{\mathrm{T}} R u_k + x_{k+1}^{\mathrm{T}} P^{\infty} x_{k+1} \end{aligned} \tag{22}$$

which is consistent with the Bellman equation (4). This completes the proof. □ ■

2) PI Algorithm: PI algorithm is a successive approximation of both the optimal policy and optimal value function. The property of PI algorithm is investigated as follows.

Lemma 4 [41] (Stability and Convergence of PI Algorithm): Consider the kernel matrix sequence $\{P_i\}_{i=0}^{\infty}$ and the feedback gain sequence $\{K_i\}_{i=0}^{\infty}$ generated by

$$V_i(x_k) = x_k^{\mathrm{T}} Q x_k + \mu_i^{\mathrm{T}}(x_k) R \mu_i(x_k) + V_i(x_{k+1}) \tag{23}$$

$$\mu_{i+1}(x_k) = \arg\min_{\mu(\cdot)} \mathcal{H}(x_k, \mu(x_k), V_i(x_k)) \tag{24}$$

with $V_i(x_k) = x_k^{\mathrm{T}} P_i x_k$, $\mu_i(x_k) = K_i x_k$, and initial admissible policy $\mu_0(x_k) = K_0 x_k$. Then, the following conditions hold.

1) $P_0 \succ P_1 \succ P_2 \succ, \ldots, \succ P_{\infty}$.
2) $\lim_{i\to\infty} P_i = P^*$, where P^* is the unique solution to ARE (9).
3) $\rho(A_{K_i}) < 1$ for $\forall i$, where $A_{K_i} = A + BK_i$. ■

3) Value Iteration: Compared to the PI algorithm, in each iteration of VI algorithm, the iterative value function does not need to satisfy the Bellman equation. Instead, a one-step recursion is involved, referred to as value function update, given by

$$V_{i+1}(x_k) = \min_{u(\cdot)} \mathcal{H}(x_k, \mu(x_k), V_i(x_k)). \tag{25}$$

With the iterative value function, the iterative policy can be accordingly updated as

$$\mu_i(x_k) = \arg\min_{\mu(\cdot)} \mathcal{H}(x_k, \mu(x_k), V_i(x_k)). \tag{26}$$

With the Bellman operator definition in (10), the value function update (25) can be rewritten as $P_{i+1} = \mathcal{B}_{K_i}(P_i)$. From (26), one has $K_i = -\left(R + B^{\mathrm{T}} P_i B\right)^{-1} B^{\mathrm{T}} P_i A$. Then, the value function update (25) can be also expressed in terms of the Riccati operator as $P_{i+1} = \mathcal{R}(P_i)$. The convergence of the VI algorithm has been investigated in the following lemma.

Lemma 5 [39] (Convergence of VI Algorithm): Consider the kernel matrix sequence $\{P_i\}_{i=1}^{\infty}$ generated as $P_{i+1} = \mathcal{R}(P_i)$ with positive definite initial kernel matrix P_0. Then, the following conditions hold.

1) $P_0 \prec P_1 \prec P_2 \prec, \ldots, \prec P_{\infty}$ *if* $P_0 \prec P^*$.
2) $P_0 \succ P_1 \succ P_2 \succ, \ldots, \succ P_{\infty}$ *if* $P_0 \succ P^*$.
3) $\lim_{i\to\infty} P_i = P^*$.

Here, P^* is the unique solution to ARE (9). ■

Remark 1: As shown later in Section III, the condition (12) plays an important role in the theoretical discussion on the main results in this article. It is worthwhile to highlight the following facts.

1) In the PI algorithm, the iteration starts with an initial admissible policy in order to guarantee the performance convergence toward the optimum [9], [30]. In contrast, as shown later in Section III for λ-PI algorithm, the iteration begins with an arbitrary value function. In addition, in order to guarantee the convergence of the λ-PI algorithm, the initial value function should satisfy condition (12), as investigated in Theorem 2.
2) If the condition (12) holds as an equation, then P is the solution to the ARE. In addition, the existence of positive definite matrix P satisfying ARE (7) can be guaranteed by Assumption 1 [42].
3) If the condition (12) holds as a strict inequality, based on (18), premultiplying and postmultiplying $M = P^{-1}$ to condition (12) yields

$$MA^{\mathrm{T}}\left(M + BR^{-1}B^{\mathrm{T}}\right)^{-1} AM - P < -MQM < 0. \tag{27}$$

Applying the Schur complement to (27), one can obtain the linear matrix inequality (LMI) condition on the

$$\begin{aligned} \mathcal{R}(P_1) - \mathcal{R}(P_2) &= A^{\mathrm{T}}\left[\left(P_1^{-1} + BR^{-1}B^{\mathrm{T}}\right)^{-1} - \left(P_2^{-1} + BR^{-1}B^{\mathrm{T}}\right)^{-1}\right]A \\ &= A^{\mathrm{T}}\left\{\left(P_1^{-1} + BR^{-1}B^{\mathrm{T}}\right)^{-1}\left[\left(P_2^{-1} + BR^{-1}B^{\mathrm{T}}\right) - \left(P_1^{-1} + BR^{-1}B^{\mathrm{T}}\right)\right]\left(P_2^{-1} + BR^{-1}B^{\mathrm{T}}\right)^{-1}\right\}A \\ &= A^{\mathrm{T}}\left[P_1 - P_1 B\left(R + B^{\mathrm{T}} P_1 B\right)^{-1} B^{\mathrm{T}} P_1\right] P_1^{-1}(P_1 - P_2) P_2^{-1}\left[P_2 - P_2 B\left(R + B^{\mathrm{T}} P_2 B\right)^{-1} B^{\mathrm{T}} P_2\right]A \\ &= \left[A^{\mathrm{T}} - A^{\mathrm{T}} P_1 B\left(R + B^{\mathrm{T}} P_1 B\right)^{-1} B^{\mathrm{T}}\right](P_1 - P_2)\left[A - B\left(R + B^{\mathrm{T}} P_2 B\right)^{-1} B^{\mathrm{T}} P_2 A\right] \end{aligned} \tag{20}$$

Algorithm 1 λ-PI Algorithm

1: Set $i = 0$ and begin with P_0 satisfying condition (12);
2: (Policy Update) update the iterative feedback gain as

$$K_{i+1} = -\left(R + B^{\mathrm{T}} P_i B\right)^{-1} B^{\mathrm{T}} P_i A, \tag{29}$$

3: (Value Update) update the iterative kernel matrix as

$$P_{i+1} = \mathcal{B}^{\lambda}_{K_{i+1}}(P_i), \tag{30}$$

4: Stop if $\|P_i - P_{i+1}\| \le \varepsilon$; otherwise, set $i = i + 1$ and go to step 1.

matrix $M = P^{-1}$ as

$$\begin{bmatrix} -M & MA^{\mathrm{T}} \\ AM & -M - BR^{-1}B^{\mathrm{T}} \end{bmatrix} < 0 \tag{28}$$

where the existence of M can be guaranteed by Assumption 1 [43]. Therefore, finding positive definite matrix P satisfying (12) is equivalent to solving LMI (28). □

Remark 2: From Definition 3, one can observe that the weighted Bellman operator is weighted by the geometric weight $(1-\lambda)\lambda^{j-1}$. The parameter λ is confined within (0, 1). From (32) to (35) in Lemma 6, one can observe that as $\lambda \to 0$, the weighted Bellman operator is the same as the Riccati operator and the λ-PI algorithm reduces to the VI algorithm. On the other hand, as $\lambda \to 1$, the weighted Bellman operator is the same as the Bellman operator and the λ-PI algorithm reduces to the PI algorithm. □

III. λ-PI Algorithm: Theoretical Discussion

A. Fixed-Point Iteration for λ-PI

In this section, we first define novel matrix operators, named weighted Bellman operator and composite Bellman operator. Then, the λ-PI algorithm is defined based on the weighted Bellman operator. Moreover, we show that the λ-PI algorithm can be written as a fixed-point iteration with the composite Bellman operator.

Definition 3: Consider a given positive simidefinite matrix P, and denote $K = \arg\min_K \mathcal{H}(x_k, u_k, V(x_{k+1})))$ with $u_k = Kx_k$. Let the kernel matrix sequence $\{P_j\}_{j=1}^{\infty}$ be defined as $P_{i+1} = \mathcal{B}_K(P_i)$ with $P_0 = P$. The weighted Bellman operator is defined as $\mathcal{B}^{\lambda}_K(P) = \sum_{j=1}^{\infty}(1-\lambda)\lambda^{j-1}P_j$ with $\lambda \in (0, 1)$. □

Based on Definition 3, the λ-PI algorithm can be presented in Algorithm 1.

Definition 4: Consider system (1) with given performance index Q and R defined in (2) and matrix $P \succ 0$. The composite Bellman operator of a matrix M is defined as $\Lambda_P(M) = (1-\lambda)\mathcal{B}_K(P) + \lambda\mathcal{B}_K(M)$, where $K = -\left(R + B^{\mathrm{T}}PB\right)^{-1}B^{\mathrm{T}}PA$ and $\lambda \in (0, 1)$. □

The relation between the weighted Bellman operator and the composite Bellman operator is investigated as follows.

Lemma 6 (Fixed-Point Formulation for the λ-PI Algorithm): Consider the kernel matrix sequence $\{P_i\}_{i=0}^{\infty}$ and the control feedback gain sequence $\{K_i\}_{i=1}^{\infty}$ generated by (30) and (29) in Algorithm 1 using the weighted Bellman operator. Then, P_{i+1} also satisfies the fixed-point equation

$$P_{i+1} = \Lambda_{P_i}(P_{i+1}) \tag{31}$$

with the composite Bellman operator $\Lambda_{P_i}(\cdot)$ parameterized by P_i.

Proof: Based on Definition 3, the iterative kernel matrix P_{i+1} can be expressed as

$$P_{i+1} = \sum_{\kappa=1}^{\infty} \bar{P}_i^{\kappa} \tag{32}$$

where $\bar{P}_i^0 = P_i$ and $\bar{P}_i^{\kappa}$ for $\kappa = 1, 2, \ldots$ is defined in (33) (see at the bottom of the next page). First, note that

$$\bar{P}_i^1 = (1-\lambda)\mathcal{B}_{K_{i+1}}\left(\bar{P}_i^0\right) = (1-\lambda)\mathcal{B}_{K_{i+1}}(P_i). \tag{34}$$

In addition, one has

$$\begin{aligned}
\bar{P}_i^2 &= \tilde{P}_i^2 + \hat{P}_i^2,\ \tilde{P}_i^2 = (1-\lambda)\lambda\left(K_{i+1}^{\mathrm{T}}RK_{i+1} + Q\right)\\
\bar{P}_i^3 &= \tilde{P}_i^3 + \hat{P}_i^3,\ \tilde{P}_i^3 = (1-\lambda)\lambda^2\left(K_{i+1}^{\mathrm{T}}RK_{i+1} + Q\right)\\
&\vdots\\
\bar{P}_i^n &= \tilde{P}_i^n + \hat{P}_i^n,\ \tilde{P}_i^n = (1-\lambda)\lambda^{n-1}\left(K_{i+1}^{\mathrm{T}}RK_{i+1} + Q\right)
\end{aligned} \tag{35}$$

where $\hat{P}_i^{\kappa}$ is defined in (36), as shown at the bottom of the next page, for $\kappa = 2, 3, \ldots$

Then,

$$\begin{aligned}
\sum_{\kappa=2}^{\infty}\tilde{P}_i^{\kappa} &= \sum_{\kappa=2}^{\infty}(1-\lambda)\lambda^{\kappa-1}\left(K_{i+1}^{\mathrm{T}}RK_{i+1} + Q\right)\\
&= \lambda\left(K_{i+1}^{\mathrm{T}}RK_{i+1} + Q\right)\\
\sum_{\kappa=2}^{\infty}\hat{P}_i^{\kappa} &= \lambda A_{K_{i+1}}^{\mathrm{T}}P_{i+1}A_{K_{i+1}}.
\end{aligned} \tag{37}$$

Therefore, based on (37), one has

$$\begin{aligned}
\sum_{\kappa=2}^{\infty}\bar{P}_i^{\kappa} &= \sum_{\kappa=2}^{\infty}\left(\tilde{P}_i^{\kappa} + \hat{P}_i^{\kappa}\right) = \sum_{\kappa=2}^{\infty}\left(\tilde{P}_i^{\kappa}\right) + \sum_{\kappa=2}^{\infty}\left(\hat{P}_i^{\kappa}\right)\\
&= \lambda\left(K_{i+1}^{\mathrm{T}}RK_{i+1} + Q + A_{K_{i+1}}^{\mathrm{T}}P_{i+1}A_{K_{i+1}}\right)\\
&= \lambda\mathcal{B}_{K_{i+1}}(P_{i+1}).
\end{aligned} \tag{38}$$

Based on (32), (34), and (38) and the composite Bellman operator in Definition 4, one can obtain that P_{i+1} satisfies the fixed-point equation (31). This completes the proof. ■

Remark 3: As shown in (30), the value function update in Algorithm 1 is a one-step recursion, which is similar to the VI algorithm. However, the λ-PI algorithm is different from VI due to the following facts.

1) From (30), the λ-PI algorithm depends on the weighted Bellman operator, which is defined based on a infinite sum of matrix recursion with geometric weights, which is difficult to obtain directly in each iteration of Algorithm 1.
2) From Lemma 6, we provide an equivalent formulation of the weighted Bellman operator in terms of an fixed-point equation with the composite Bellman operator. Then, the value function update (30) can be rewritten as

$$\begin{aligned}
P_{i+1} = {} & (1-\lambda)(A + BK_{i+1})^{\mathrm{T}}P_i(A + BK_{i+1})\\
& + \lambda(A + BK_{i+1})^{\mathrm{T}}P_{i+1}(A + BK_{i+1})\\
& + K_{i+1}^{\mathrm{T}}RK_{i+1} + Q
\end{aligned} \tag{39}$$

where the value function update P_{i+1} depends on both P_i and K_{i+1}. Therefore, the λ-PI algorithm can be viewed as a combination of VI (where the value function update P_{i+1} depends only on P_i) and PI (where the value function update P_{i+1} depends only on K_{i+1}). □

In the following, the fixed-point iteration with convergence proof is presented to solve the fixed-point matrix equation (31) with the composite Bellman operator.

Theorem 1 (Composite Bellman Operator Property): If the matrix P satisfies condition (12), then the composite Bellman operator in Definition 4 has the following properties.

1) *Contraction Property:* The composite Bellman operator is a contract mapping in the sense that
$$\|\Lambda_P(M_1) - \Lambda_P(M_2)\| < \|M_1 - M_2\|. \tag{40}$$
2) *Monotonic Property:* The composite Bellman operator is monotonic in the sense that
$$\Lambda_P(M_1) \succ (\succeq) \Lambda_P(M_2), \quad \text{if } M_1 \succ (\succeq) M_2 \tag{41}$$
where "$\succ$" holds if $M_1 \succ M_2$.
3) The fixed point of the composite Bellman operator is unique.
4) The fixed point M of the composite Bellman operator $\Lambda_P(\cdot)$ parameterized by P satisfies $M \leq P$.

Proof: The following conditions hold.

1) From Definition 4, one has
$$\Lambda_P(M_1) - \Lambda_P(M_2) = \lambda(A + BK)^{\mathrm{T}}(M_1 - M_2)(A + BK). \tag{42}$$
Since P satisfies condition (12), the matrix $A + BK$ is Schur stable with $K = \arg\min_K H(x_k, u_k, V(x_{k+1})))$, where $u_k = Kx_k$. Then, $\rho(A + BK) < 1$ and
$$\begin{aligned}\|\Lambda_P(M_1) - \Lambda_P(M_2)\| &\leq \lambda \cdot [\rho(A + BK)]^2 \cdot \|M_1 - M_2\| \\ &< \|M_1 - M_2\|. \end{aligned}\tag{43}$$
2) From (42), one can observe that $\Lambda_P(M_1) \succ (\succeq) \Lambda_P(M_2)$, if $M_1 \succ (\succeq) M_2$, where "$\succ$" holds if $M_1 \succ M_2$.
3) From Proposition 1), $\Lambda_P(\cdot)$ is a contraction mapping with modulus less than one. Then, according to [21], the fixed-point of the *composite Bellman operator* is unique.
4) The fixed-point equation $M = \Lambda_P(M)$ can be equivalently written as
$$\begin{aligned} M = {} & Q + K^{\mathrm{T}}RK + (A + BK)^{\mathrm{T}}P(A + BK) \\ & + \lambda(A + BK)^{\mathrm{T}}(M - P)(A + BK). \end{aligned}\tag{44}$$
Then,
$$\begin{aligned} M - P = {} & Q + K^{\mathrm{T}}RK + (A + BK)^{\mathrm{T}}P(A + BK) - P \\ & + \lambda(A + BK)^{\mathrm{T}}(M - P)(A + BK). \end{aligned}\tag{45}$$
Since P satisfies condition (12), according to Lemma 1, the matrix $A + BK$ is Schur stable with
$$K = \arg\min_K \mathcal{H}(x_k, u_k, V(x_{k+1})))$$
where $u_k = Kx_k$ and
$$Q + K^{\mathrm{T}}RK + (A + BK)^{\mathrm{T}}P(A + BK) - P \leq 0.$$
Then,
$$M - P \leq \lambda(A + BK)^{\mathrm{T}}(M - P)(A + BK). \tag{46}$$
Note that the matrix $A + BK$ is Schur stable, and then, $\rho(A + BK) < 1$ and
$$\left[1 - \lambda\rho^2(A + BK)\right](M - P) \leq 0 \tag{47}$$
which implies $M \leq P$. This completes the proof. ■

To this end, the convergence of the λ-PI algorithm is investigated as follows.

Theorem 2 (Convergence of λ-PI Algorithm): Consider the matrix sequence $\{P_i^j\}_{j=0}^{\infty}$ with $P_i^0 = P_i$ generated by the recursion
$$P_i^{j+1} = \Lambda_{P_i}\left(P_i^j\right). \tag{48}$$
Suppose that P_0 satisfies condition (12). Then, the following conditions hold.

1) The matrix sequence $\{P_i^j\}_{j=0}^{\infty}$ is nonincreasing, i.e., $P_i^{j+1} \leq P_i^j$.
2) The matrix sequence $\{P_i^j\}_{j=0}^{\infty}$ converges to the unique solution P_{i+1} to the fixed-point equation (31).
3) The sequence $\{P_i\}_{i=0}^{\infty}$ is nonincreasing and also satisfies condition (12), i.e., $\forall x_k \in \mathbb{R}^n$, one has
$$\begin{aligned} x_k^{\mathrm{T}}P_{i+1}x_k &\leq \min_{u_k = Kx_k} \mathcal{H}\left(x_k, u_k, x_{k+1}^{\mathrm{T}}P_i x_{k+1}\right) \\ &\leq x_k^{\mathrm{T}}P_i x_k. \end{aligned}\tag{49}$$
4) The sequence $\{P_i\}_{i=0}^{\infty}$ converges to the solution to the ARE (7), i.e., $\lim_{i\to\infty} P_i = P^*$.

Proof: First, we prove by induction. For $j = 0$, one has
$$\begin{aligned} P_i^1 = {} & (1 - \lambda)(A + BK_{i+1})^{\mathrm{T}}P_i(A + BK_{i+1}) \\ & + \lambda(A + BK_{i+1})^{\mathrm{T}}P_i(A + BK_{i+1}) \\ & + Q + K_{i+1}^{\mathrm{T}}RK_{i+1} \\ = {} & Q + K_{i+1}^{\mathrm{T}}RK_{i+1} + (A + BK_{i+1})^{\mathrm{T}}P_i(A + BK_{i+1}) \\ \leq {} & P_i = P_i^0 \end{aligned}\tag{50}$$

$$\bar{P}_i^{\kappa} = (1 - \lambda)\lambda^{\kappa-1}\mathcal{B}_{K_{i+1}}^{(\kappa)}(P_i) = (1 - \lambda)\lambda^{\kappa-1}\left[\sum_{j=0}^{\kappa-1}\left(A_K^{\mathrm{T}}\right)^j\left(K_{i+1}^{\mathrm{T}}RK_{i+1} + Q\right)(A_K)^j + \left(A_{K_{i+1}}^{\mathrm{T}}\right)^{\kappa}P_i\left(A_{K_{i+1}}\right)^{\kappa}\right], \tag{33}$$

$$\hat{P}_i^{\kappa} = (1 - \lambda)\lambda^{\kappa-1}A_{K_{i+1}}^{\mathrm{T}}\left\{\sum_{\kappa=0}^{\kappa-2}\left[\left(A_{K_{i+1}}^{\mathrm{T}}\right)^{\kappa}\left(K_{i+1}^{\mathrm{T}}RK_{i+1} + Q\right)\left(A_{K_{i+1}}\right)^{\kappa}\right] + \left(A_{K_{i+1}}^{\mathrm{T}}\right)^{\kappa-1}P_i\left(A_{K_{i+1}}\right)^{\kappa-1}\right\}A_{K_{i+1}} \tag{36}$$

where the inequality results from the facts that P_i satisfies condition (12).

Suppose that $P_i^j \le P_i^{j-1}$. Then, for P_i^{j+1} and P_i^j, one has

$$\begin{aligned} P_i^{j+1} - P_i^j &= \Lambda_{P_i}\left(P_i^j\right) - \Lambda_{P_i}\left(P_i^{j-1}\right) \\ &= \lambda(A+BK_{i+1})^{\mathrm{T}}\left(P_i^j - P_i^{j-1}\right)(A+BK_{i+1}). \end{aligned} \tag{51}$$

Therefore, $P_i^{j+1} \le P_i^j$ due to the fact that $P_i^j \le P_i^{j-1}$.

Second, from Propositions 1) and 3) in Theorem 1, one can conclude that the operator $\Lambda_{P_i}(\cdot)$ has modulus less than 1 and has a unique fixed point. According to [21], the fixed-point iteration (48) converges to the unique solution, P_{i+1}, to the fixed-point equation (31).

Third, we prove by induction. From Propositions 1) and 2), one has $P_1 = \lim_{j\to\infty} P_0^j \le P_0^1$. In addition, based on the fixed-point iteration (48), one has

$$\begin{aligned} P_0^1 &= (1-\lambda)(A+BK_1)^{\mathrm{T}}P_0(A+BK_1) \\ &\quad +\lambda(A+BK_1)^{\mathrm{T}}P_0^0(A+BK_1) \\ &\quad +Q+K_1^{\mathrm{T}}RK_1 \\ &= (A+BK_1)^{\mathrm{T}}P_0(A+BK_1)+Q+K_1^{\mathrm{T}}RK_1, \\ x_k^{\mathrm{T}}P_0^1x_k &= \min_{u_k=Kx_k}\mathcal{H}\left(x_k,u_k,x_{k+1}^{\mathrm{T}}P_0x_{k+1}\right). \end{aligned} \tag{52}$$

Therefore, for $i=0$, P_1 satisfies

$$x_k^{\mathrm{T}}P_ix_k \le \min_{u_k=Kx_k}\mathcal{H}\left(x_k,u_k,x_{k+1}^{\mathrm{T}}P_0x_{k+1}\right).$$

In addition, P_0 satisfies condition (12), i.e.,

$$\min_{u_k=Kx_k}\mathcal{H}\left(x_k,u_k,x_{k+1}^{\mathrm{T}}P_0x_{k+1}\right) \le x_k^{\mathrm{T}}P_0x_k.$$

Then, the pair (P_0, P_1) satisfies condition (49).

Suppose that (P_{i-1}, P_i) satisfies

$$\begin{aligned} x_k^{\mathrm{T}}P_ix_k &\le \min_{u_k=Kx_k}\mathcal{H}\left(x_k,u_k,x_{k+1}^{\mathrm{T}}P_{i-1}x_{k+1}\right) \\ &\le x_k^{\mathrm{T}}P_{i-1}x_k. \end{aligned} \tag{53}$$

For P_i, we have

$$\begin{aligned} x_k^{\mathrm{T}}P_ix_k &= (1-\lambda)x_k^{\mathrm{T}}(A+BK_i)^{\mathrm{T}}P_{i-1}(A+BK_i)x_k \\ &\quad -(1-\lambda)x_k^{\mathrm{T}}(A+BK_i)^{\mathrm{T}}P_i(A+BK_i)x_k \\ &\quad +x_k^{\mathrm{T}}\left[Q+K_i^{\mathrm{T}}RK_i+(A+BK_i)^{\mathrm{T}}P_i(A+BK_i)\right]x_k \\ &= (1-\lambda)(A+BK_i)^{\mathrm{T}}(P_{i-1}-P_i)(A+BK_i) \\ &\quad +\min_{u_k=Kx_k}\mathcal{H}\left(x_k,u_k,x_{k+1}^{\mathrm{T}}P_ix_{k+1}\right) \\ &\ge \min_{u_k=Kx_k}\mathcal{H}\left(x_k,u_k,x_{k+1}^{\mathrm{T}}P_ix_{k+1}\right) \end{aligned} \tag{54}$$

which implies that P_i satisfies

$$\min_{u_k=Kx_k}\mathcal{H}\left(x_k,u_k,x_{k+1}^{\mathrm{T}}P_ix_{k+1}\right) \le x_k^{\mathrm{T}}P_ix_k.$$

In addition, one has $P_{i+1} = \lim_{j\to\infty} P_i^j \le P_i^1$. Based on the fixed-point iteration (48), one has

$$\begin{aligned} x_k^{\mathrm{T}}P_i^1\,x_k &= (1-\lambda)x_k^{\mathrm{T}}(A+BK_{i+1})^{\mathrm{T}}P_i(A+BK_{i+1})x_k \\ &\quad +\lambda x_k^{\mathrm{T}}(A+BK_{i+1})^{\mathrm{T}}P_i^0(A+BK_{i+1})x_k \\ &\quad +x_k^{\mathrm{T}}\left[Q+K_{i+1}^{\mathrm{T}}RK_{i+1}\right]x_k \\ &= x_k^{\mathrm{T}}\left[Q+K_{i+1}^{\mathrm{T}}RK_{i+1}\right]x_k \\ &\quad +x_k^{\mathrm{T}}(A+BK_{i+1})^{\mathrm{T}}P_i(A+BK_{i+1})x_k \\ &= \min_{u_k=Kx_k}\mathcal{H}\left(x_k,u_k,x_{k+1}^{\mathrm{T}}P_ix_{k+1}\right) \end{aligned} \tag{55}$$

which implies that

$$x_k^{\mathrm{T}}P_{i+1}x_k \le \min_{u_k=Kx_k}\mathcal{H}\left(x_k,u_k,x_{k+1}^{\mathrm{T}}P_ix_{k+1}\right).$$

Therefore, the pair (P_{i+1}, P_i) also satisfies condition (49).

Fourth, from Proposition 3), the sequence $\{P_i\}_{i=0}^{\infty}$ is nonincreasing. In addition, the sequence $\{P_i\}_{i=0}^{\infty}$ is lower bounded by zero matrix, i.e., $P_i \ge 0$. Therefore, $P_\infty := \lim_{i\to\infty} P_i$ exists. As $i, j \to \infty$, both the fixed-point iteration (48) and the fixed-point equation (31) satisfy the ARE (7), i.e.,

$$P_\infty = A^{\mathrm{T}}P_\infty A + Q - A^{\mathrm{T}}P_\infty B\left(R+B^{\mathrm{T}}P_\infty B\right)^{-1}B^{\mathrm{T}}P_\infty A.$$

This completes the proof. ■

B. Performance and Parameter Analysis

It is well known that the PI has a faster convergence speed toward the optimum compared to the VI algorithm under the same iterative condition [21]. In this section, it is shown that the λ-PI algorithm is a tradeoff between PI and VI in terms of convergence rate to the optimum. In addition, the effect of the parameter λ on the λ-PI algorithm is discussed. It is shown that the λ-PI algorithm is equivalent to the VI algorithm as $\lambda \to 0$ and is equivalent to the PI algorithm as $\lambda \to 1$.

Theorem 3 (Performance Comparison): Consider a positive definite value function kernel matrix Π_i satisfying condition (12) and starting from Π_i in the ith iteration with the control policy update $K_{i+1} = -\left(R+B^{\mathrm{T}}\Pi_iB\right)B^{\mathrm{T}}\Pi_iA$ in the $(i+1)$th iteration. Denote the value function update in the PI algorithm, VI algorithm and λ-PI algorithm, respectively, as

$$\begin{aligned} \textbf{PI}: \Psi_{i+1} &= (A+BK_{i+1})^{\mathrm{T}}\Psi_{i+1}(A+BK_{i+1}) \\ &\quad +Q+K_{i+1}^{\mathrm{T}}RK_{i+1} \end{aligned} \tag{56}$$

$$\begin{aligned} \textbf{VI}: \Gamma_{i+1} &= (A+BK_{i+1})^{\mathrm{T}}\Pi_i(A+BK_{i+1}) \\ &\quad +Q+K_{i+1}^{\mathrm{T}}RK_{i+1} \end{aligned} \tag{57}$$

$$\begin{aligned} \lambda\textbf{-PI}: \Phi_{i+1} &= \lambda(A+BK_{i+1})^{\mathrm{T}}\Phi_{i+1}(A+BK_{i+1}) \\ &\quad +(1-\lambda)(A+BK_{i+1})^{\mathrm{T}}\Pi_i(A+BK_{i+1}) \\ &\quad +Q+K_{i+1}^{\mathrm{T}}RK_{i+1}. \end{aligned} \tag{58}$$

Then, $\Psi_{i+1} \le \Phi_{i+1} \le \Gamma_{i+1}$.

Proof: First, from (56) and (58), one can obtain

$$\begin{aligned} \Psi_{i+1} - \Phi_{i+1} &= (A+BK_{i+1})^{\mathrm{T}}(\Psi_{i+1}-\Phi_{i+1})(A+BK_{i+1}) \\ &\quad +(1-\lambda)(A+BK_{i+1})^{\mathrm{T}}(\Phi_{i+1}-\Pi_i) \\ &\quad \times(A+BK_{i+1}). \end{aligned} \tag{59}$$

Since the matrix Π_i satisfies condition (12), from Lemma 1, one has $\bar{\rho} \triangleq \rho(A + BK_{i+1}) \leq 1$. Therefore, (59) further implies that

$$\left(1-\bar{\rho}^2\right)(\Psi_{i+1}-\Phi_{i+1}) \leq (1-\lambda)\bar{\rho}^2(\Phi_{i+1}-\Pi_i). \quad (60)$$

On the other hand, for λ-PI algorithm (58), it is true from Theorem 2 that $\Phi_{i+1} \leq \Pi_i$, i.e.,

$$\Phi_{i+1} - \Pi_i \leq 0. \quad (61)$$

Combining (60) and (61) yields

$$\left(1-\bar{\rho}^2\right)(\Psi_{i+1}-\Phi_{i+1}) \leq 0 \quad (62)$$

which implies that $\Psi_{i+1} \leq \Phi_{i+1}$ due to the fact that $\bar{\rho} \leq 1$.

Second, for the λ-PI algorithm, the matrix Φ_{i+1} can be obtained using the fixed-point iteration

$$\Phi_i^{j+1} = \Lambda_{\Pi_i}\left(\Phi_i^j\right) \quad (63)$$

with initial condition $\Phi_i^0 = \Pi_i$. Then, from Theorem 2, the sequence $\{\Phi_i^j\}_{j=0}^{\infty}$ satisfies, $\forall j = 0, 1, 2, \ldots$

$$\Phi_{i+1} = \lim_{j\to\infty} \Phi_i^j \leq \Phi_i^{j+1} \leq \Phi_i^j \leq \Phi_i^0 = \Pi_i. \quad (64)$$

On the other hand, the fixed-point iteration (63) with $j = 0$ can be written as

$$\begin{aligned}\Phi_i^1 &= \lambda(A+BK_{i+1})^{\mathrm{T}}\Phi_i^0(A+BK_{i+1}) \\ &\quad +(1-\lambda)(A+BK_{i+1})^{\mathrm{T}}\Pi_i(A+BK_{i+1}) \\ &\quad +Q+K_{i+1}^{\mathrm{T}}RK_{i+1} \\ &= (A+BK_{i+1})^{\mathrm{T}}\Pi_i(A+BK_{i+1}) \\ &\quad +Q+K_{i+1}^{\mathrm{T}}RK_{i+1} \\ &= \Gamma_{i+1}. \end{aligned} \quad (65)$$

From (65), one can obtain $\Phi_{i+1} \leq \Gamma_{i+1}$. This completes the proof. ■

Theorem 4 (Parameter Comparison): Consider the value function update in the λ-PI algorithm

$$\begin{aligned}P_{i+1}^{(r)} &= \left(1-\lambda^{(r)}\right)(A+BK_{i+1})^{\mathrm{T}}P_i(A+BK_{i+1}) \\ &\quad +\lambda^{(r)}(A+BK_{i+1})^{\mathrm{T}}P_{i+1}^{(r)}(A+BK_{i+1}) \\ &\quad +Q+K_{i+1}^{\mathrm{T}}RK_{i+1}, r=1,2 \end{aligned} \quad (66)$$

with two different parameters $\lambda^{(1)} \geq \lambda^{(2)}$. Then, $P_{i+1}^{(1)} \leq P_{i+1}^{(2)}$.

Proof: From (66), one has

$$\begin{aligned}P_{i+1}^1 - P_{i+1}^2 &= \left(\lambda^{(2)}-\lambda^{(1)}\right)(A+BK_{i+1})^{\mathrm{T}}P_i(A+BK_{i+1}) \\ &\quad +(A+BK_{i+1})^{\mathrm{T}}\left[\lambda^{(1)}P_{i+1}^{(1)}-\lambda^{(2)}P_{i+1}^{(2)}\right] \\ &\quad \times(A+BK_{i+1}). \end{aligned} \quad (67)$$

Note that

$$\lambda^{(1)}P_{i+1}^{(1)}-\lambda^{(2)}P_{i+1}^{(2)} = \lambda^{(1)}P_{i+1}^{(1)}-\lambda^{(1)}P_{i+1}^{(2)}+\lambda^{(1)}P_{i+1}^{(2)}-\lambda^{(2)}P_{i+1}^{(2)}. \quad (68)$$

Then, inserting (68) back into (67) yields

$$\begin{aligned}P_{i+1}^1 - P_{i+1}^2 &= \left(\lambda^{(2)}-\lambda^{(1)}\right)(A+BK_{i+1})^{\mathrm{T}}P_i(A+BK_{i+1}) \\ &\quad +\lambda^{(1)}(A+BK_{i+1})^{\mathrm{T}}\left[P_{i+1}^{(1)}-P_{i+1}^{(2)}\right] \\ &\quad \times(A+BK_{i+1}) \\ &\quad +(A+BK_{i+1})^{\mathrm{T}}\left[\lambda^{(1)}P_{i+1}^{(2)}-\lambda^{(2)}P_{i+1}^{(2)}\right] \\ &\quad \times(A+BK_{i+1}) \\ &\leq \left(\lambda^{(2)}-\lambda^{(1)}\right)\bar{\rho}^2P_i+\lambda^{(1)}\bar{\rho}^2\left[P_{i+1}^{(1)}-P_{i+1}^{(2)}\right] \\ &\quad +\bar{\rho}^2\left[\lambda^{(1)}-\lambda^{(2)}\right]P_{i+1}^{(2)} \\ &= \bar{\rho}^2\left(\lambda^{(1)}-\lambda^{(2)}\right)\left[P_{i+1}^{(2)}-P_i\right] \\ &\quad +\lambda^{(1)}\bar{\rho}^2\left[P_{i+1}^{(1)}-P_{i+1}^{(2)}\right] \end{aligned}$$

which is equivalent to

$$\left(1-\lambda^{(1)}\bar{\rho}^2\right)\left[P_{i+1}^1-P_{i+1}^2\right] \leq \bar{\rho}^2\left(\lambda^{(1)}-\lambda^{(2)}\right)\left[P_{i+1}^{(2)}-P_i\right] \leq 0 \quad (69)$$

where the last inequality results from the fact that $P_{i+1}^{(2)} \leq P_i$ owing to Proposition 1) in Theorem 2. Note that $\lambda^{(r)}, \bar{\rho} \in (0, 1)$. Then, from (69), one has $P_{i+1}^{(1)} \leq P_{i+1}^{(2)}$. This completes the proof. ■

IV. λ-PI Algorithm: Implementation Discussion

A. On-Policy λ-PI Algorithm

With the iterative policy $\mu^i(x_k) = K^i x_k$, the one-step state transition can be denoted as

$$x_{k+1}^{\mathrm{on}} = Ax_k + B\mu^{i+1}(x_k). \quad (70)$$

Based on the linear quadratic theory [2], the iterative value function can be written as in quadratic form, i.e., $V^i(x_k) = x_k^{\mathrm{T}}P^i x_k$. Therefore, (39) in Remark 3 can be equivalently written as

$$\begin{aligned}V^{i+1}(x_k) &= r\left(x_k, \mu^{i+1}(x_k)\right) + \lambda V^{i+1}\left(x_{k+1}^{\mathrm{on}}\right) \\ &\quad +(1-\lambda)V^i\left(x_{k+1}^{\mathrm{on}}\right) \end{aligned} \quad (71)$$

which is equivalent to

$$V^{i+1}(x_k) - \lambda V^{i+1}\left(x_{k+1}^{\mathrm{on}}\right) = r\left(x_k, \mu^{i+1}(x_k)\right) + (1-\lambda)V^i\left(x_{k+1}^{\mathrm{on}}\right). \quad (72)$$

With the linear feedback control policy $\mu^{i+1}(x_k) = K^{i+1}x_k$, the state update follows:

$$x_{k+1}^{\mathrm{on}} = \left(A + BK^{i+1}\right)x_k = A^{i+1}x_k. \quad (73)$$

Similarly, from (72), (39) can be equivalently written as the on-policy composite Bellman equation as

$$\begin{aligned}&x_k^{\mathrm{T}}P^{i+1}x_k - \lambda\left(x_{k+1}^{\mathrm{on}}\right)^{\mathrm{T}}P^{i+1}x_{k+1}^{\mathrm{on}} \\ &\quad = x_k^{\mathrm{T}}Qx_k + x_k^{\mathrm{T}}\left(K^{i+1}\right)^{\mathrm{T}}RK^{i+1}x_k \\ &\qquad + (1-\lambda)x_k^{\mathrm{T}}\left(A+BK^{i+1}\right)P^i\left(A+BK^{i+1}\right)x_k. \end{aligned} \quad (74)$$

To this end, the on-policy λ-PI algorithm can be summarized in Algorithm 2.

Given the iterative kernel matrix P^i, the iterative feedback gain K^{i+1} can be updated using (29). Then, in the on-policy

Algorithm 2 On-Policy λ-PI Without Noise

1: Set $i = 0$ and begin with $P^i(\cdot)$ satisfying condition (12);
2: *Data Collection Phase*: Collect the online sequential data;
3: *Learning Phase*:
   i. (*Policy Improvement Step*) Update K^{i+1} as (29);
   ii. (*Policy Evaluation Step*) Solve (74) for P^{i+1};
   iii. Stop if $\left\| V^i(x_k) - V^{i+1}(x_k) \right\| \leq \varepsilon$, for $\forall x_k$; Otherwise, set $i = i + 1$ and go to Step 2.

Algorithm 3 Model-Based Off-Policy λ-PI Without Noise

1: Set $i = 0$ and begin with $P^i(\cdot)$ satisfying condition (12);
2: *Data Collection Phase*: Collect the online sequential data;
3: *Learning Phase*:
   i. (*Policy Improvement Step*) Update K^{i+1} as (29);
   ii. (*Policy Evaluation Step*) Solve (81) or (82) for P^{i+1};
   iii. Stop if $\left\| V^i(x_k) - V^{i+1}(x_k) \right\| \leq \varepsilon$, for $\forall x_k$; Otherwise, set $i = i + 1$ and go to Step 2.

composite Bellman equation (74), the variable to be determined is only P^{i+1}. Using notation of the Kronecker product, (74) can be equivalently written as

$$\varphi_k^{\text{on}} \cdot \upsilon ec\left(P^{i+1}\right) = \theta_k^{\text{on}} \tag{75}$$

with

$$\begin{aligned}\varphi_k^{\text{on}} &= x_k^{\mathrm{T}} \otimes x_k^{\mathrm{T}} - \lambda\left[\left(x_{k+1}^{\text{on}}\right)^{\mathrm{T}} \otimes \left(x_{k+1}^{\text{on}}\right)^{\mathrm{T}}\right] \\ \theta_k^{\text{on}} &= x_k^{\mathrm{T}} Q x_k + x_k^{\mathrm{T}}\left(K^{i+1}\right)^{\mathrm{T}} R K^{i+1} x_k \\ &\quad + (1-\lambda) x_k^{\mathrm{T}}\left(A + BK^{i+1}\right) P^i \left(A + BK^{i+1}\right) x_k \end{aligned} \tag{76}$$

which is a least squares (LS) equation of $(((n+1)n)/2)$ independent elements in P^{i+1}. To guarantee the existence and uniqueness of solution to (75) for online implementation, the concept of persistent excitation (PE) is required.

Definition 5 (Persistent Excitation) [44]: A bounded vector signal $\eta_i \in \mathbb{R}^q$, $q > 1$ is said to satisfy the persistent excitation (PE) condition if there exist $L > 0$ and $\alpha_0 > 0$ such that

$$\sum_{i=k}^{k+L} \eta_i \eta_i^{\mathrm{T}} \geq \alpha_0 \; I \;\; \forall k \geq i_0. \qquad \square$$

Remark 4: First, to solve the on-policy composite Bellman equation (74), the complete system dynamics $\{A, B\}$ is required. In addition, for linear dynamical system (1), the uniqueness of the solution to on-policy composite Bellman equation (74) cannot be guaranteed because the regressor φ_k^{on} in the LS equation (75) might not satisfy the PE condition. In the following, we present the off-policy extension of the λ-PI algorithm to guarantee the uniqueness of the on-policy composite Bellman equation (74) without the requirement of the complete system dynamics $\{A, B\}$. $\square$

B. Off-Policy λ-PI Algorithm

1) Model-Based Off-Policy λ-PI Without Probing Noise: With the behavior policy w_k, the state transition follows:

$$x_{k+1}^{\text{off}} = Ax_k + Bw_k. \tag{77}$$

Adding and subtracting the iterative policy $\mu^{i+1}(x_k) = K^{i+1}x_k$ into (77) yields

$$\begin{aligned} x_{k+1}^{\text{off}} &= Ax_k + BK^{i+1}x_k + Bw_k - BK^{i+1}x_k \\ &= \left(A + BK^{i+1}\right)x_k + Bw_k - BK^{i+1}x_k \\ &= A^{i+1}x_k + B\left(w_k - K^{i+1}x_k\right) \\ &= x_{k+1}^{\text{on}} + B\left(w_k - K^{i+1}x_k\right). \end{aligned} \tag{78}$$

Based on the Taylor expansion, we have

$$\begin{aligned} &x_k^{\mathrm{T}} P^{i+1} x_k - \lambda\left(x_{k+1}^{\text{off}}\right)^{\mathrm{T}} P^{i+1} x_{k+1}^{\text{off}} \\ &\quad = x_k^{\mathrm{T}} P^{i+1} x_k - \lambda\left(x_{k+1}^{\text{on}}\right) P^{i+1} x_{k+1}^{\text{on}} \\ &\qquad + \lambda\left(w_k - K^{i+1}x_k\right)^{\mathrm{T}} B^{\mathrm{T}} P^{i+1} B\left(w_k - K^{i+1}x_k\right) \\ &\qquad - 2\lambda\left(w_k - K^{i+1}x_k\right)^{\mathrm{T}} B^{\mathrm{T}} P^{i+1} x_{k+1}^{\text{off}}. \end{aligned} \tag{79}$$

From the on-policy Bellman equation (74), the terms $x_k^{\mathrm{T}} P^{i+1} x_k - \lambda\left(x_{k+1}^{\text{on}}\right)^{\mathrm{T}} P^{i+1} x_{k+1}^{\text{on}}$ satisfy

$$\begin{aligned} &x_k^{\mathrm{T}} P^{i+1} x_k - \lambda\left(x_{k+1}^{\text{on}}\right)^{\mathrm{T}} P^{i+1} x_{k+1}^{\text{on}} \\ &\quad = x_k^{\mathrm{T}} Q x_k + x_k^{\mathrm{T}}\left(K^{i+1}\right)^{\mathrm{T}} R K^{i+1} x_k \\ &\qquad + (1-\lambda) x_k^{\mathrm{T}}\left(A + BK^{i+1}\right) P^i \left(A + BK^{i+1}\right) x_k. \end{aligned} \tag{80}$$

Then, inserting (80) into (79) yields the off-policy composite Bellman equation

$$\begin{aligned} &x_k^{\mathrm{T}} P^{i+1} x_k - \lambda\left(x_{k+1}^{\text{off}}\right)^{\mathrm{T}} P^{i+1} x_{k+1}^{\text{off}} \\ &\quad = x_k^{\mathrm{T}} Q x_k + x_k^{\mathrm{T}}\left(K^{i+1}\right)^{\mathrm{T}} R K^{i+1} x_k \\ &\qquad + (1-\lambda) x_k^{\mathrm{T}}\left(A + BK^{i+1}\right) P^i \left(A + BK^{i+1}\right) x_k \\ &\qquad + \lambda\left(w_k - K^{i+1}x_k\right)^{\mathrm{T}} B^{\mathrm{T}} P^{i+1} B\left(w_k - K^{i+1}x_k\right) \\ &\qquad - 2\lambda\left(w_k - K^{i+1}x_k\right)^{\mathrm{T}} B^{\mathrm{T}} P^{i+1} x_{k+1}^{\text{off}} \end{aligned} \tag{81}$$

where the online data $\left\{x_k, w_k, x_{k+1}^{\text{off}}\right\}$ are involved. Using notation of the Kronecker product, (81) can be equivalently written as

$$\varphi_k^{\text{off}} \cdot \upsilon ec\left(P^{i+1}\right) = \theta_k^{\text{off}} \tag{82}$$

with

$$\begin{aligned} \varphi_k^{\text{off}} &= x_k^{\mathrm{T}} \otimes x_k^{\mathrm{T}} - \lambda\left[\left(x_{k+1}^{\text{off}}\right)^{\mathrm{T}} \otimes \left(x_{k+1}^{\text{off}}\right)^{\mathrm{T}}\right] \\ \theta_k^{\text{off}} &= x_k^{\mathrm{T}} Q x_k + x_k^{\mathrm{T}}\left(K^{i+1}\right)^{\mathrm{T}} R K^{i+1} x_k \\ &\quad + (1-\lambda) x_k^{\mathrm{T}}\left(A + BK^{i+1}\right) P^i \left(A + BK^{i+1}\right) x_k \\ &\quad + \lambda\left(w_k - K^{i+1}x_k\right)^{\mathrm{T}} B^{\mathrm{T}} P^{i+1} B\left(w_k - K^{i+1}x_k\right) \\ &\quad - 2\lambda\left(w_k - K^{i+1}x_k\right)^{\mathrm{T}} B^{\mathrm{T}} P^{i+1} x_{k+1}^{\text{off}}. \end{aligned} \tag{83}$$

To this end, the model-based off-policy RL for solving the off-policy composite Bellman equation (81) and (82) is shown in Algorithm 3.

The equivalence between the off-policy RL in Algorithm 3 and the on-policy RL in Algorithm 2 is discussed in the following lemma.

Corollary 1: Starting from the iterative kernel matrix P^i, denote the value function update as $\bar{P}_{\text{on}}^{i+1}$ and $\hat{P}_{\text{off}}^{i+1}$ for the on- and off-policy RL algorithms, respectively. Then,

the on-policy RL Algorithm 2 is equivalent to the off-policy RL Algorithm 3 in the sense that the on-policy composite Bellman equation (74) and the off-policy composite Bellman equation (81) are equivalent.

Proof: See the Appendix. ■

2) Model-Based Off-Policy λ-PI With Probing Noise: Applying the behavior policy w_k with the probing noise e_k, the state update follows:

$$\tilde{x}_{k+1}^{\text{off}} = Ax_k + Bv_k \tag{84}$$

where $v_k = w_k + e_k$ is the behavior policy with probing noise. Similar to (78), one has

$$\begin{aligned}\tilde{x}_{k+1}^{\text{off}} &= Ax_k + Bv_k \\ &= Ax_k + BK^{i+1}x_k + B\big(v_k - K^{i+1}x_k\big) \\ &= A^{i+1}x_k + B\big(v_k - K^{i+1}x_k\big) \\ &= x_{k+1}^{\text{on}} + B\big(v_k - K^{i+1}x_k\big).\end{aligned} \tag{85}$$

Applying off-policy composite Bellman equation (81) for system (84) with input v_k yields

$$\begin{aligned}&x_k^{\mathrm{T}}P^{i+1}x_k - \lambda\big(\tilde{x}_{k+1}^{\text{off}}\big)^{\mathrm{T}}P^{i+1}\tilde{x}_{k+1}^{\text{off}} \\ &\quad= x_k^{\mathrm{T}}Qx_k + x_k^{\mathrm{T}}\big(K^{i+1}\big)^{\mathrm{T}}RK^{i+1}x_k \\ &\qquad+ (1-\lambda)x_k^{\mathrm{T}}\big(A + BK^{i+1}\big)P^i\big(A + BK^{i+1}\big)x_k \\ &\qquad+ \lambda\big(v_k - K^{i+1}x_k\big)^{\mathrm{T}}B^{\mathrm{T}}P^{i+1}B\big(v_k - K^{i+1}x_k\big) \\ &\qquad- 2\lambda\big(v_k - K^{i+1}x_k\big)^{\mathrm{T}}B^{\mathrm{T}}P^{i+1}\tilde{x}_{k+1}^{\text{off}}\end{aligned} \tag{86}$$

where the online data $\big\{x_k, v_k, \tilde{x}_{k+1}^{\text{off}}\big\}$ are involved. Using notation of the Kronecker product, (81) can be equivalently written as

$$\tilde{\varphi}_k^{\text{off}} \cdot vec\big(P^{i+1}\big) = \tilde{\theta}_k^{\text{off}} \tag{87}$$

with

$$\begin{aligned}\tilde{\varphi}_k^{\text{off}} &= x_k^{\mathrm{T}} \otimes x_k^{\mathrm{T}} - \lambda\Big[\big(\tilde{x}_{k+1}^{\text{off}}\big)^{\mathrm{T}} \otimes \big(\tilde{x}_{k+1}^{\text{off}}\big)^{\mathrm{T}}\Big] \\ \tilde{\theta}_k^{\text{off}} &= x_k^{\mathrm{T}}Qx_k + x_k^{\mathrm{T}}\big(K^{i+1}\big)^{\mathrm{T}}RK^{i+1}x_k \\ &\quad+(1-\lambda)x_k^{\mathrm{T}}\big(A + BK^{i+1}\big)P^i\big(A + BK^{i+1}\big)x_k \\ &\quad+\lambda\big(v_k - K^{i+1}x_k\big)^{\mathrm{T}}B^{\mathrm{T}}P^{i+1}B\big(v_k - K^{i+1}x_k\big) \\ &\quad-2\lambda\big(v_k - K^{i+1}x_k\big)^{\mathrm{T}}B^{\mathrm{T}}P^{i+1}\tilde{x}_{k+1}^{\text{off}}.\end{aligned} \tag{88}$$

For the subsequent discussions, we refer (86) as the dithered off-policy composite Bellman equation due to the fact that $v_k = w_k + e_k$.

Based on the dithered off-policy composite Bellman equation (86) and (87), the off-policy RL algorithm with probing noise in the control input can be shown in Algorithm 4.

The effect of probing noise e_k in the control input v_k on solving the off-policy composite Bellman equation (86) is investigated in the following lemma.

Corollary 2: Starting from the iterative kernel matrix P^i, denote the value function update as $\hat{P}_{\text{off}}^{i+1}$ and $\tilde{P}_{\text{off}}^{i+1}$ for Algorithms 3 and 4, respectively. Then, Algorithm 3 is equivalent to Algorithm 4 in the sense that the off-policy composite Bellman equation (81) and the dithered off-policy composite Bellman equation (86) are equivalent, i.e., $\hat{P}_{\text{off}}^{i+1} = \tilde{P}_{\text{off}}^{i+1}$.

Proof: See the Appendix. ■

Algorithm 4 Model-Based Off-Policy λ-PI With Probing Noise

1: Set $i = 0$ and begin with $P^i(\cdot)$ satisfying condition (12);
2: *Data Collection Phase*: Collect the online sequential data;
3: *Learning Phase*:
   i. (*Policy Improvement Step*) Update K^{i+1} as (29);
   ii. (*Policy Evaluation Step*) Solve (86) or (87) for P^{i+1};
   iii. Stop if $\big\|V^i(x_k) - V^{i+1}(x_k)\big\| \le \varepsilon$, for $\forall x_k$; Otherwise, set $i = i + 1$ and go to Step 2.

3) Off-Policy λ-PI Algorithm in LS Compact Form: With the probing noise e_k in the behavior policy v_k, the off-policy composite Bellman equation (86) or (87) can be equivalently written as

$$\begin{aligned}&x_k^{\mathrm{T}}P^{i+1}x_k - \lambda\big(\tilde{x}_{k+1}^{\text{off}}\big)^{\mathrm{T}}P^{i+1}\tilde{x}_{k+1}^{\text{off}} \\ &\quad= x_k^{\mathrm{T}}Qx_k + x_k^{\mathrm{T}}\big(K^{i+1}\big)^{\mathrm{T}}RK^{i+1}x_k \\ &\qquad+ (1-\lambda)x_k^{\mathrm{T}}\big(A + BK^{i+1}\big)P^i\big(A + BK^{i+1}\big)x_k \\ &\qquad- \lambda\big(w_k - K^{i+1}x_k\big)^{\mathrm{T}}B^{\mathrm{T}}P^{i+1}B\big(w_k + K^{i+1}x_k\big) \\ &\qquad- 2\lambda\big(w_k - K^{i+1}x_k\big)^{\mathrm{T}}B^{\mathrm{T}}P^{i+1}Ax_k\end{aligned} \tag{89}$$

which is equivalent to

$$\begin{aligned}&\Big\{\big(x_k^{\mathrm{T}} \otimes x_k^{\mathrm{T}}\big) - \lambda\Big[\big(\tilde{x}_{k+1}^{\text{off}}\big)^{\mathrm{T}} \otimes \big(\tilde{x}_{k+1}^{\text{off}}\big)^{\mathrm{T}}\Big]\Big\}vec\big(P^{i+1}\big) \\ &\quad+ \lambda\Big[\big(w_k - K^{i+1}x_k\big)^{\mathrm{T}} \otimes \big(w_k + K^{i+1}x_k\big)\Big]vec\big(B^{\mathrm{T}}P^{i+1}B\big) \\ &\quad+ 2\lambda\Big[\big(w_k - K^{i+1}x_k\big)^{\mathrm{T}} \otimes x_k^{\mathrm{T}}\Big]vec\big(B^{\mathrm{T}}P^{i+1}A\big) \\ &\quad- (1-\lambda)\big(x_k^{\mathrm{T}} \otimes x_k^{\mathrm{T}}\big)vec\Big(\big(A + BK^{i+1}\big)P^i\big(A + BK^{i+1}\big)\Big) \\ &= x_k^{\mathrm{T}}Qx_k + x_k^{\mathrm{T}}\big(K^{i+1}\big)^{\mathrm{T}}RK^{i+1}x_k\end{aligned} \tag{90}$$

with a compact form as $\Gamma_k^i \cdot \Psi^{i+1} = \delta_k^i$, where

$$\begin{aligned}\Gamma_k^i &= \big[\,\Gamma_{xx1,k}^i\ \Gamma_{ww,k}^i\ \Gamma_{wx,k}^i\ \Gamma_{xx2,k}^i\,\big] \\ \Gamma_{xx1,k}^i &= \big(x_k^{\mathrm{T}} \otimes x_k^{\mathrm{T}}\big) - \lambda\Big[\big(\tilde{x}_{k+1}^{\text{off}}\big)^{\mathrm{T}} \otimes \big(\tilde{x}_{k+1}^{\text{off}}\big)^{\mathrm{T}}\Big] \\ \Gamma_{ww,k}^i &= \lambda\big(w_k - K^{i+1}x_k\big)^{\mathrm{T}} \otimes \big(w_k + K^{i+1}x_k\big) \\ \Gamma_{wx,k}^i &= 2\lambda\big(w_k - K^{i+1}x_k\big)^{\mathrm{T}} \otimes x_k^{\mathrm{T}} \\ \Gamma_{xx2,k}^i &= -(1-\lambda)\big(x_k^{\mathrm{T}} \otimes x_k^{\mathrm{T}}\big) \\ \Psi^{i+1} &= \begin{bmatrix} vec\big(\psi_{xx1}^{i+1}\big) \\ vec\big(\psi_{ww}^{i+1}\big) \\ vec\big(\psi_{wx}^{i+1}\big) \\ vec\big(\psi_{xx2}^{i+1}\big)\end{bmatrix},\ \begin{cases}\psi_{xx1}^{i+1} = P^{i+1} \\ \psi_{ww}^{i+1} = B^{\mathrm{T}}P^{i+1}B \\ \psi_{wx}^{i+1} = B^{\mathrm{T}}P^{i+1}A \\ \psi_{xx2}^{i+1} = \big(A^{i+1}\big)^{\mathrm{T}}P^iA^{i+1}\end{cases} \\ \delta_k^i &= x_k^{\mathrm{T}}Qx_k + x_k^{\mathrm{T}}\big(K^{i+1}\big)^{\mathrm{T}}RK^{i+1}x_k\end{aligned} \tag{91}$$

with $A^{i+1} = A + BK^{i+1}$. Collecting the data on the interval $[k_1, k_1 + N]$ yields the LS equation as

$$\Gamma_{k_1:k_1+N}^i \cdot \Psi^{i+1} = \delta_{k_1:k_1+N}^i \tag{92}$$

where $\Gamma_{k_1:k_1+N}^i$ and $\delta_{k_1:k_1+N}^i$ are defined in (93) (see at the bottom of the next page). In addition, the policy update $K^{i+1} = -\big(R + B^{\mathrm{T}}P^iB\big)^{-1}B^{\mathrm{T}}P^iA$ can be rewritten as

$$K^{i+1} = -\big(R + \psi_{ww}^i\big)^{-1}\psi_{wx}^i. \tag{94}$$

Algorithm 5 Model-Free Off-Policy λ-PI With Probing Noise

1: Set $i = 0$ and begin with $V_i(x_k) = x_k^{\mathrm{T}} P_i x_k$ satisfying condition (12);
2: ***Data Collection Phase***: Collect online data to establish matrix Γ_k^i and δ_k^i;
3: ***Learning Phase***:
4: **while** $\left\| V^i(x_k) - V^{i+1}(x_k) \right\| > \varepsilon$ **do**
5: Update feedback gain K^{i+1} using (94).
6: Solve the LS equation (92) for Ψ^{i+1}.
7: **end while**
8: ***Implementation Phase***:
9: Extract $\left\{ \psi_{xx1}^i, \psi_{ww}^i, \psi_{wx}^i, \psi_{xx2}^i \right\}$ from Ψ^i and determine the approximate optimal feedback gain K^{i+1} based on (94).
10: Apply the approximate optimal policy $u_k = K_k^{i+1}$ to system (1).

Corollary 3: Suppose that the collected online data guarantees that the regressor Γ_k^i satisfies the PE condition. Then, Algorithm 4 is equivalent to Algorithm 5 in the sense that the following conditions hold.

1) The LS equation (92) has the same solution as the LS equation (87).
2) Equations (29) and (94) yield the same policy improvement.

Proof: See the Appendix. ■

Remark 5: First, from Remark 4, the PE condition is necessary to guarantee the uniqueness and existence of the solution to the on-policy composite Bellman equation (74) in Algorithm 2. In addition, from Corollaries 1–3, one can conclude that the LS equations (75), (82), (87), and (92) have the same solution. Note that the LS equation (75) is equivalent to (74), which is essentially a matrix Lyapunov equation with a unique solution P^{i+1} due to Assumption 1. Therefore, the PE condition on the regressors φ_k^{on} of (75), φ_k^{off} of (82), $\tilde{\varphi}_k^{\mathrm{off}}$ of (87), and $\Gamma_{k_1:k_1+N}^i$ of (92) is necessary to guarantee the existence and uniqueness of the LS solution P^{i+1}. □

V. Simulation Study

This section presents a simulation example to illustrate the effectiveness of the proposed off-policy RL algorithm. The linear system (1) with the following system matrix and control input dynamic matrix is considered:

$$A = \begin{bmatrix} 0.9065 & 0.0816 & -0.0005 \\ 0.0743 & 0.9012 & -0.0007 \\ 0 & 0 & 0.1327 \end{bmatrix}$$
$$B = \begin{bmatrix} -0.0027 \\ -0.0068 \\ 1 \end{bmatrix}.$$

For the performance specification, the weight matrices are selected as

$$Q = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, R = 1.$$

From the optimal control theory [2], the solution to the ARE and the optimal feedback gain can be determined as

$$P^* = \begin{bmatrix} 15.1268 & 12.0315 & -0.0082 \\ 12.0315 & 15.2282 & -0.0080 \\ -0.0082 & -0.0080 & 1.0088 \end{bmatrix}$$
$$K^* = \begin{bmatrix} 0.0643 & 0.0699 & -0.0667 \end{bmatrix}. \quad (95)$$

For the data collection phase, in order to guarantee the full rank of the regressor matrix in LS equations (92), the behavior policy $u_k = Kx_k + e_k$ is applied to the system for 30 steps with the feedback gain K and probing noise e_k as

$$K = \begin{bmatrix} 0 & -0.12 & -1 \end{bmatrix}$$
$$e(k) = \sin^2(0.5k) + \sin(k) + \cos(k). \quad (96)$$

The state evolution during the data collection phase is shown in Fig. 1.

For the λ-PI algorithm, the initial iterative kernel matrix P^0 is selected as identity matrix, and the parameter λ is selected as three different values as 0.5, 0.7, and 0.9. In the learning phase, the difference between the iterative kernel matrix P^i and the optimal kernel matrix P^*, and the difference between the iterative feedback gain matrix K^i and the optimal feedback gain K^* are shown in Fig. 2. From Fig. 2, one can observe that the learning phase with the iterative matrix P^i and K^i converges to the optimum after 80 iterations. Then, the learning policies K^{80} can be regarded as the approximate optimal controller. After the data collection phase, the online running data collected from the first 30 steps are used in the learning phase for 80 iterations. After the learning phase, the converged learning policy is applied to the system in the implementation, where the state evolution during the interval [30, 50] is shown in Fig. 1. It can be shown in Fig. 1 that, at the convergence of the learning phase, the closed-loop stability can also be guaranteed in the implementation phase. In addition, in both cases, the iterative learning gain K^i converge to the optimal gain K^*, regardless of the probing noise in the behavior policy.

We also compare the λ-PI algorithm with VI [45] and PI [46] in terms of convergence rate and investigate the effect of parameter λ, as discussed in Section III-B. As shown in Fig. 2, one can observe that as the iteration continues, all the iterative algorithms converge toward the optimum. PI has the fastest convergence rate, VI has the slowest convergence rate, and λ-PI has the intermediate convergence rate. In addition, we investigate three cases with different λ values for the λ-PI

$$\Gamma_{k_1:k_1+N}^i = \begin{bmatrix} \Gamma_{xx1,k_1}^i & \Gamma_{ww,k_1}^i & \Gamma_{wx,k_1}^i & \Gamma_{xx2,k_1}^i \\ \vdots & \vdots & \vdots & \vdots \\ \Gamma_{xx1,k_1+N}^i & \Gamma_{ww,k_1+N}^i & \Gamma_{wx,k_1+N}^i & \Gamma_{xx2,k_1+N}^i \end{bmatrix}, \quad \delta_{k_1:k_1+N}^i = \begin{bmatrix} \delta_{k_1}^i \\ \vdots \\ \delta_{k_1+N}^i \end{bmatrix} \quad (93)$$

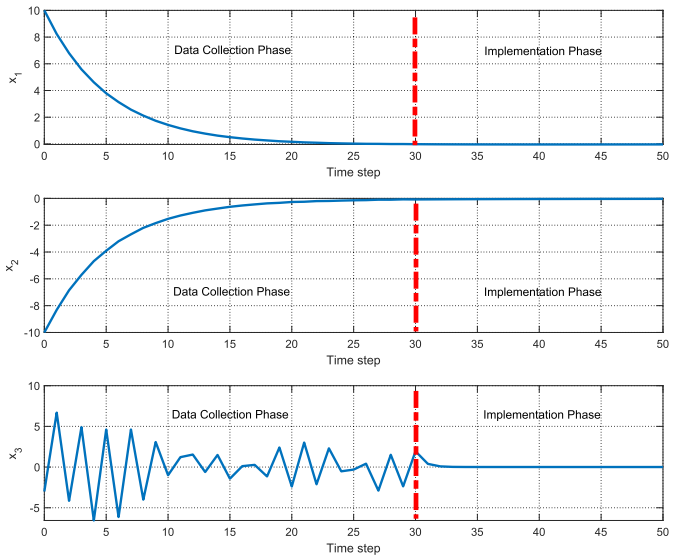

Fig. 1. Data collection phase and implementation phase.

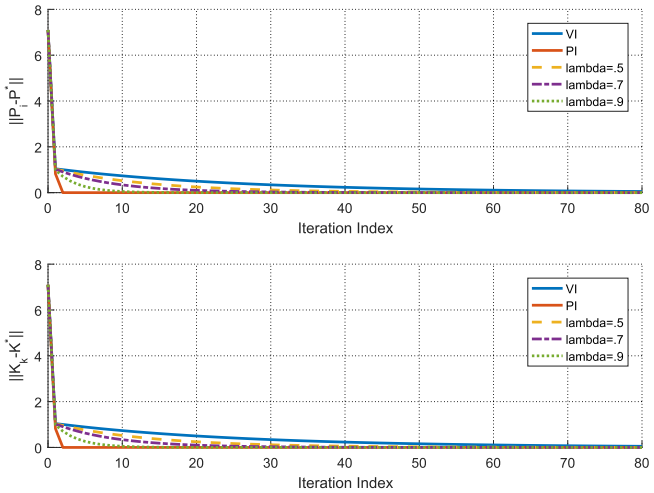

Fig. 2. Learning process of PI, VI, and λ-PI algorithm.

algorithm. As discussed in Section III-B, as $\lambda \to 0$, the λ-PI algorithm behaves like the VI algorithm, and as $\lambda \to 1$, the λ-PI algorithm behaves like the PI algorithm.

VI. Conclusion

In this article, we investigate the λ-PI algorithm for the discrete-time LQR problem. The λ-PI algorithm is first defined as the weighted sum of a sequence of Bellman operators. Then, an equivalent formulation of the λ-PI algorithm is derived in terms of the fixed-point equation. On this basis, the fixed-point iteration algorithm for the λ-PI algorithm is developed with convergence proof. The performance comparison between the λ-PI algorithm and classical VI and PI algorithm is conducted to show that λ-PI is an intermediate between VI and PI. In addition, on- and off-policy variants of the λ-PI algorithm are developed for online implementation. It is shown that the off-policy λ-PI algorithm does not require the system dynamics and is robust against the probing noise. Simulation examples are conducted to verify the efficacy of the λ-PI algorithm for LQR problem.

Future work aims to investigate the extension of λ-PI algorithm for dynamical systems in continuous time with challenging scenarios, such as nonlinear systems with unmodeled dynamics [47] and multiobjective consideration [48].

Appendix

Proof of Corollary 1: First, replacing P^{i+1} in (81) with $\hat{P}_{\text{off}}^{i+1}$ and inserting $x_{k+1}^{\text{off}} = (A + BK^{i+1})x_k + B(w_k - K^{i+1}x_k)$ into the left-hand side of off-policy composite Bellman equation (81) yields

$$\begin{aligned}
&-2\lambda\left[(A+BK^{i+1})x_k\right]^{\mathrm{T}}\hat{P}_{\text{off}}^{i+1}B(w_k - K^{i+1}x_k)\\
&\quad -\lambda(w_k - K^{i+1}x_k)^{\mathrm{T}}B^{\mathrm{T}}\hat{P}_{\text{off}}^{i+1}B(w_k - K^{i+1}x_k)\\
&\quad -\lambda\left[(A+BK^{i+1})x_k\right]^{\mathrm{T}}\hat{P}_{\text{off}}^{i+1}\left[(A+BK^{i+1})x_k\right]\\
&\quad + x_k^{\mathrm{T}}\hat{P}_{\text{off}}^{i+1}x_k\\
&= x_k^{\mathrm{T}}Qx_k + x_k^{\mathrm{T}}(K^{i+1})^{\mathrm{T}}RK^{i+1}x_k\\
&\quad + (1-\lambda)x_k^{\mathrm{T}}(A+BK^{i+1})P^i(A+BK^{i+1})x_k\\
&\quad - 2\lambda\left[(A+BK^{i+1})x_k\right]^{\mathrm{T}}\hat{P}_{\text{off}}^{i+1}B(w_k - K^{i+1}x_k)\\
&\quad -\lambda(w_k - K^{i+1}x_k)^{\mathrm{T}}B^{\mathrm{T}}\hat{P}_{\text{off}}^{i+1}B(w_k - K^{i+1}x_k). \qquad (97)
\end{aligned}$$

Canceling the common terms on both sides of (97) yields

$$\begin{aligned}
&x_k^{\mathrm{T}}\hat{P}_{\text{off}}^{i+1}x_k - \lambda x_k^{\mathrm{T}}(A+BK^{i+1})^{\mathrm{T}}\hat{P}_{\text{off}}^{i+1}(A+BK^{i+1})x_k\\
&\quad = (1-\lambda)x_k^{\mathrm{T}}(A+BK^{i+1})P^i(A+BK^{i+1})x_k\\
&\quad\quad + x_k^{\mathrm{T}}Qx_k + x_k^{\mathrm{T}}(K^{i+1})^{\mathrm{T}}RK^{i+1}x_k
\end{aligned}$$

from which one can observe that $\hat{P}_{\text{off}}^{i+1}$ also satisfies the on-policy composite Bellman equation (74). This completes the proof. ■

Proof of Corollary 2: First, replacing P^{i+1} in the off-policy composite Bellman equation (81) with $\hat{P}_{\text{off}}^{i+1}$ generated by Algorithm 3 yields

$$\begin{aligned}
&x_k^{\mathrm{T}}\hat{P}_{\text{off}}^{i+1}x_k - \lambda(x_{k+1}^{\text{off}})^{\mathrm{T}}\hat{P}_{\text{off}}^{i+1}x_{k+1}^{\text{off}}\\
&\quad = x_k^{\mathrm{T}}Qx_k + x_k^{\mathrm{T}}(K^{i+1})^{\mathrm{T}}RK^{i+1}x_k\\
&\quad\quad + (1-\lambda)x_k^{\mathrm{T}}(A+BK^{i+1})P^i(A+BK^{i+1})x_k\\
&\quad\quad + \lambda(w_k - K^{i+1}x_k)^{\mathrm{T}}B^{\mathrm{T}}\hat{P}_{\text{off}}^{i+1}B(w_k - K^{i+1}x_k)\\
&\quad\quad - 2\lambda(w_k - K^{i+1}x_k)^{\mathrm{T}}B^{\mathrm{T}}\hat{P}_{\text{off}}^{i+1}x_{k+1}^{\text{off}}. \qquad (98)
\end{aligned}$$

Second, replacing P^{i+1} in the off-policy composite Bellman equation (81) with $\bar{P}_{\text{off}}^{i+1}$ generated by Algorithm 4 and inserting $v_k = w_k + e_k$ into equation (86) yields

$$\begin{aligned}
&x_k^{\mathrm{T}}\tilde{P}_{\text{off}}^{i+1}x_k - \lambda(\tilde{x}_{k+1}^{\text{off}})^{\mathrm{T}}\tilde{P}_{\text{off}}^{i+1}\tilde{x}_{k+1}^{\text{off}}\\
&\quad = x_k^{\mathrm{T}}Qx_k + x_k^{\mathrm{T}}(K^{i+1})^{\mathrm{T}}RK^{i+1}x_k\\
&\quad\quad + (1-\lambda)x_k^{\mathrm{T}}(A+BK^{i+1})P^i(A+BK^{i+1})x_k\\
&\quad\quad + \lambda e_k^{\mathrm{T}}B^{\mathrm{T}}\tilde{P}_{\text{off}}^{i+1}Be_k\\
&\quad\quad + \lambda(w_k - K^{i+1}x_k)^{\mathrm{T}}B^{\mathrm{T}}\tilde{P}_{\text{off}}^{i+1}B(w_k - K^{i+1}x_k)\\
&\quad\quad + 2\lambda(w_k - K^{i+1}x_k)^{\mathrm{T}}B^{\mathrm{T}}\tilde{P}_{\text{off}}^{i+1}Be_k\\
&\quad\quad - 2\lambda(w_k - K^{i+1}x_k)^{\mathrm{T}}B^{\mathrm{T}}\tilde{P}_{\text{off}}^{i+1}\tilde{x}_{k+1}^{\text{off}}\\
&\quad\quad - 2\lambda e_k^{\mathrm{T}}B^{\mathrm{T}}\tilde{P}_{\text{off}}^{i+1}\tilde{x}_{k+1}^{\text{off}}. \qquad (99)
\end{aligned}$$

From (70) and (77), one has $\tilde{x}_{k+1}^{\text{off}} = x_{k+1}^{\text{off}} + Be_k$. Inserting this fact into the right-hand side of (99) yields

$$\begin{aligned} x_k^{\mathrm{T}} \tilde{P}_{\text{off}}^{i+1} x_k &- \lambda \big(\tilde{x}_{k+1}^{\text{off}}\big)^{\mathrm{T}} \tilde{P}_{\text{off}}^{i+1} \tilde{x}_{k+1}^{\text{off}} \\ &= x_k^{\mathrm{T}} Q x_k + x_k^{\mathrm{T}} \big(K^{i+1}\big)^{\mathrm{T}} R K^{i+1} x_k \\ &\quad + (1-\lambda) x_k^{\mathrm{T}} \big(A + BK^{i+1}\big) P^i \big(A + BK^{i+1}\big) x_k \\ &\quad + \lambda \big(w_k - K^{i+1} x_k\big)^{\mathrm{T}} B^{\mathrm{T}} \tilde{P}_{\text{off}}^{i+1} B \big(w_k - K^{i+1} x_k\big) \\ &\quad - 2\lambda \big(w_k - K^{i+1} x_k\big)^{\mathrm{T}} B^{\mathrm{T}} \tilde{P}_{\text{off}}^{i+1} x_{k+1}^{\text{off}} \\ &\quad - 2\lambda e_k^{\mathrm{T}} B^{\mathrm{T}} \tilde{P}_{\text{off}}^{i+1} x_{k+1}^{\text{off}} \\ &\quad - \lambda e_k^{\mathrm{T}} B^{\mathrm{T}} \tilde{P}_{\text{off}}^{i+1} B e_k. \end{aligned} \tag{100}$$

On the other hand, inserting $\tilde{x}_{k+1}^{\text{off}} = x_{k+1}^{\text{off}} + Be_k$ into the left-hand side of (100) yields

$$\begin{aligned} x_k^{\mathrm{T}} \tilde{P}_{\text{off}}^{i+1} x_k &- \lambda \big(\tilde{x}_{k+1}^{\text{off}}\big)^{\mathrm{T}} \tilde{P}_{\text{off}}^{i+1} \tilde{x}_{k+1}^{\text{off}} \\ &= x_k^{\mathrm{T}} \tilde{P}_{\text{off}}^{i+1} x_k - \lambda \big(x_{k+1}^{\text{off}}\big)^{\mathrm{T}} \tilde{P}_{\text{off}}^{i+1} x_{k+1}^{\text{off}} \\ &\quad - \lambda e_k^{\mathrm{T}} B^{\mathrm{T}} \tilde{P}_{\text{off}}^{i+1} B e_k - 2\lambda \big(x_{k+1}^{\text{off}}\big)^{\mathrm{T}} \tilde{P}_{\text{off}}^{i+1} B e_k. \end{aligned} \tag{101}$$

Combining the facts in (100) and (101), one has

$$\begin{aligned} x_k^{\mathrm{T}} \tilde{P}_{\text{off}}^{i+1} x_k &- \lambda \big(x_{k+1}^{\text{off}}\big)^{\mathrm{T}} \tilde{P}_{\text{off}}^{i+1} x_{k+1}^{\text{off}} \\ &\quad - \lambda e_k^{\mathrm{T}} B^{\mathrm{T}} \tilde{P}_{\text{off}}^{i+1} B e_k - 2\lambda \big(x_{k+1}^{\text{off}}\big)^{\mathrm{T}} \tilde{P}_{\text{off}}^{i+1} B e_k \\ &= x_k^{\mathrm{T}} Q x_k + x_k^{\mathrm{T}} \big(K^{i+1}\big)^{\mathrm{T}} R K^{i+1} x_k \\ &\quad + (1-\lambda) x_k^{\mathrm{T}} \big(A + BK^{i+1}\big) P^i \big(A + BK^{i+1}\big) x_k \\ &\quad + \lambda \big(w_k - K^{i+1} x_k\big)^{\mathrm{T}} B^{\mathrm{T}} \tilde{P}_{\text{off}}^{i+1} B \big(w_k - K^{i+1} x_k\big) \\ &\quad - 2\lambda \big(w_k - K^{i+1} x_k\big)^{\mathrm{T}} B^{\mathrm{T}} \tilde{P}_{\text{off}}^{i+1} x_{k+1}^{\text{off}} \\ &\quad - 2\lambda e_k^{\mathrm{T}} B^{\mathrm{T}} \tilde{P}_{\text{off}}^{i+1} x_{k+1}^{\text{off}} \\ &\quad - \lambda e_k^{\mathrm{T}} B^{\mathrm{T}} \tilde{P}_{\text{off}}^{i+1} B e_k. \end{aligned} \tag{102}$$

Canceling the common terms on both sides of equation (102) yields

$$\begin{aligned} x_k^{\mathrm{T}} \tilde{P}_{\text{off}}^{i+1} x_k &- \lambda \big(x_{k+1}^{\text{off}}\big)^{\mathrm{T}} \tilde{P}_{\text{off}}^{i+1} x_{k+1}^{\text{off}} \\ &= x_k^{\mathrm{T}} Q x_k + x_k^{\mathrm{T}} \big(K^{i+1}\big)^{\mathrm{T}} R K^{i+1} x_k \\ &\quad + (1-\lambda) x_k^{\mathrm{T}} \big(A + BK^{i+1}\big) P^i \big(A + BK^{i+1}\big) x_k \\ &\quad + \lambda \big(w_k - K^{i+1} x_k\big)^{\mathrm{T}} B^{\mathrm{T}} \tilde{P}_{\text{off}}^{i+1} B \big(w_k - K^{i+1} x_k\big) \\ &\quad - 2\lambda \big(w_k - K^{i+1} x_k\big)^{\mathrm{T}} B^{\mathrm{T}} \tilde{P}_{\text{off}}^{i+1} x_{k+1}^{\text{off}}. \end{aligned} \tag{103}$$

Therefore, collecting the results in (81), (86), (98), and (103), one can observe that the solution $\tilde{P}_{\text{off}}^{i+1}$ to the dithered off-policy composite Bellman equation (86) also satisfies the off-policy composite Bellman equation (81). This completes the proof. ■

Proof of Corollary 3: The following facts hold.

1) Based on (89) and (90), one can observe that the LS equation (92) is equivalent to (86) and (87) with the same solution. In addition, from Corollaries 1 and 2, (86) and (87) are equivalent to the on-policy composite Bellman equation (74) and (75), which is essentially a matrix Lyapunov equation with a unique solution P^{i+1} due to Assumption 1. Therefore, the LS equation (92) has the same unique solution as the LS equation (87).
2) Based on Proposition 1 and (29) and (94), Algorithms 4 and 5 yields the same policy update. This completes the proof. ■

REFERENCES

[1] R. Bellman, "Dynamic programming," *Science*, vol. 153, nos. 37–31, pp. 34–37, 1966.

[2] F. L. Lewis, D. Vrabie, and V. L. Syrmos, *Optimal Control*. Hoboken, NJ, USA: Wiley, 2012.

[3] J. L. Speyer and D. H. Jacobson, *Primer on Optimal Control Theory*. Philadelphia, PA, USA: SIAM, 2010.

[4] W. B. Powell, *Approximate Dynamic Programming: Solving the Curses of Dimensionality*, vol. 703. Hoboken, NJ, USA: Wiley, 2007.

[5] J. Y. Lee, J. B. Park, and Y. H. Choi, "On integral value iteration for continuous-time linear systems," in *Proc. Amer. Control Conf.*, Jun. 2013, pp. 4215–4220.

[6] Y. Li, Z. Hou, Y. Feng, and R. Chi, "Data-driven approximate value iteration with optimality error bound analysis," *Automatica*, vol. 78, pp. 79–87, Apr. 2017.

[7] H. G. Su, H. G. Zhang, K. Zhang, and W. Z. Gao, "Online reinforcement learning for a class of partially unknown continuous-time nonlinear systems via value iteration," *Optim. Control Appl. Meth.*, vol. 39, no. 2, pp. 1011–1028, Mar. 2018.

[8] K. G. Vamvoudakis and F. L. Lewis, "Online actor-critic algorithm to solve the continuous-time infinite horizon optimal control problem," *Automatica*, vol. 46, no. 5, pp. 878–888, May 2010.

[9] Y. Yang, D. Wunsch, and Y. Yin, "Hamiltonian-driven adaptive dynamic programming for continuous nonlinear dynamical systems," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 8, pp. 1929–1940, Aug. 2017.

[10] H. Modares, F. L. Lewis, and M.-B. Naghibi-Sistani, "Adaptive optimal control of unknown constrained-input systems using policy iteration and neural networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 24, no. 10, pp. 1513–1525, Oct. 2013.

[11] A. Heydari, "Stability analysis of optimal adaptive control using value iteration with approximation errors," *IEEE Trans. Autom. Control*, vol. 63, no. 9, pp. 3119–3126, Sep. 2018.

[12] T. Bian and Z.-P. Jiang, "Value iteration and adaptive dynamic programming for data-driven adaptive optimal control design," *Automatica*, vol. 71, pp. 348–360, Sep. 2016.

[13] T. Bian, Y. Jiang, and Z.-P. Jiang, "Adaptive dynamic programming and optimal control of nonlinear nonaffine systems," *Automatica*, vol. 50, no. 10, pp. 2624–2632, Oct. 2014.

[14] D. Liu, H. Li, and D. Wang, "Error bounds of adaptive dynamic programming algorithms for solving undiscounted optimal control problems," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 6, pp. 1323–1334, Jun. 2015.

[15] Y. Yang, K. G. Vamvoudakis, and H. Modares, "Safe reinforcement learning for dynamical games," *Int. J. Robust Nonlinear Control*, vol. 30, no. 9, pp. 3706–3726, Jun. 2020.

[16] Y. Yang, D.-W. Ding, H. Xiong, Y. Yin, and D. C. Wunsch, "Online barrier-actor-critic learning for H_∞ control with full-state constraints and input saturation," *J. Franklin Inst.*, vol. 357, no. 6, pp. 3316–3344, Apr. 2020.

[17] M. Liang, D. Wang, and D. Liu, "Improved value iteration for neural-network-based stochastic optimal control design," *Neural Netw.*, vol. 124, pp. 280–295, Apr. 2020.

[18] X. Li, L. Xue, and C. Sun, "Linear quadratic tracking control of unknown discrete-time systems using value iteration algorithm," *Neurocomputing*, vol. 314, pp. 86–93, Nov. 2018.

[19] F. L. Lewis and D. Vrabie, "Reinforcement learning and adaptive dynamic programming for feedback control," *IEEE Circuits Syst. Mag.*, vol. 9, no. 3, pp. 32–50, Aug. 2009.

[20] D. P. Bertsekas, *Dynamic Programming and Optimal Control*, 1st ed., vols. 1–2. Belmont, MA, USA: Athena Scientific, 1995.

[21] D. P. Bertsekas, *Abstract Dynamic Programming*. Belmont, MA, USA: Athena Scientific, 2018.

[22] D. P. Bertsekas, "Lambda-policy iteration: A review and a new implementation," in *Reinforcement Learning and Approximate Dynamic Programming for Feedback Control*. Hoboken, NJ, USA: Wiley, 2013.

[23] S. Bhasin, R. Kamalapurkar, M. Johnson, K. G. Vamvoudakis, F. L. Lewis, and W. E. Dixon, "A novel actor-critic-identifier architecture for approximate optimal control of uncertain nonlinear systems," *Automatica*, vol. 49, no. 1, pp. 82–92, Jan. 2013.

[24] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. Cambridge, MA, USA: MIT Press, 2018.

[25] Y. Jiang and Z.-P. Jiang, "Computational adaptive optimal control for continuous-time linear systems with completely unknown dynamics," *Automatica*, vol. 48, no. 10, pp. 2699–2704, 2012.

[26] Z.-P. Jiang, T. Bian, and W. Gao, "Learning-based control: A tutorial and some recent results," *Found. Trends Syst. Control*, vol. 8, no. 3, pp. 176–284, 2020.

[27] B. Kiumarsi, F. L. Lewis, and Z.-P. Jiang, "H_∞ control of linear discrete-time systems: Off-policy reinforcement learning," *Automatica*, vol. 78, pp. 144–152, Apr. 2017.

[28] Y. Yang, Z. Guo, H. Xiong, D. Ding, Y. Yin, and D. C. Wunsch, "Data-driven robust control of discrete-time uncertain linear systems via off-policy reinforcement learning," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 12, pp. 3735–3747, Dec. 2019.

[29] B. Kiumarsi, K. G. Vamvoudakis, H. Modares, and F. L. Lewis, "Optimal and autonomous control using reinforcement learning: A survey," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 6, pp. 2042–2062, Jun. 2018.

[30] B. Luo, H.-N. Wu, T. Huang, and D. Liu, "Data-based approximate policy iteration for affine nonlinear continuous-time optimal control design," *Automatica*, vol. 50, no. 12, pp. 3281–3290, 2014.

[31] Y. Jiang and Z.-P. Jiang, *Robust Adaptive Dynamic Programming*. Hoboken, NJ, USA: Wiley, 2017.

[32] T. Bian and Z.-P. Jiang, "Continuous-time robust dynamic programming," *SIAM J. Control Optim.*, vol. 57, no. 6, pp. 4150–4174, Jan. 2019.

[33] Y. Yang, K. G. Vamvoudakis, H. Modares, Y. Yin, and D. C. Wunsch, "Hamiltonian-driven hybrid adaptive dynamic programming," *IEEE Trans. Syst., Man, Cybern., Syst.*, early access, Jan. 13, 2020, doi: 10.1109/TSMC.2019.2962103.

[34] Z. Chen and S. Jagannathan, "Generalized Hamilton–Jacobi–Bellman formulation-based neural network control of affine nonlinear discrete-time systems," *IEEE Trans. Neural Netw.*, vol. 19, no. 1, pp. 90–106, Jan. 2008.

[35] J. Škach, B. Kiumarsi, F. L. Lewis, and O. Straka, "Actor-critic off-policy learning for optimal control of multiple-model discrete-time systems," *IEEE Trans. Cybern.*, vol. 48, no. 1, pp. 29–40, Jan. 2018.

[36] F. L. Lewis and K. G. Vamvoudakis, "Reinforcement learning for partially observable dynamic processes: Adaptive dynamic programming using measured output data," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 41, no. 1, pp. 14–25, Feb. 2011.

[37] R. A. Horn and C. R. Johnson, *Matrix Analysis*. Cambridge, U.K.: Cambridge Univ. Press, 2012.

[38] N. S. Tripathy, I. N. Kar, and K. Paul, "Stabilization of uncertain discrete-time linear system with limited communication," *IEEE Trans. Autom. Control*, vol. 62, no. 9, pp. 4727–4733, Sep. 2017.

[39] N. Komaroff, "Iterative matrix bounds and computational solutions to the discrete algebraic Riccati equation," *IEEE Trans. Autom. Control*, vol. 39, no. 8, pp. 1676–1678, Aug. 1994.

[40] F. L. Lewis, D. Vrabie, and K. G. Vamvoudakis, "Reinforcement learning and feedback control: Using natural decision methods to design optimal adaptive controllers," *IEEE Control Syst.*, vol. 32, no. 6, pp. 76–105, Dec. 2012.

[41] P. Lancaster and L. Rodman, *Algebraic Riccati Equations*. Oxford, U.K.: Clarendon Press, 1995.

[42] P. Lancaster and L. Rodman, *Algebraic Riccati Equations*. London, U.K.: Oxford Univ. Press, 1995.

[43] G.-R. Duan and H.-H. Yu, *LMIs in Control Systems: Analysis, Design and Applications*. Boca Raton, FL, USA: CRC Press, 2013.

[44] G. Tao, *Adaptive Control Design and Analysis*. Hoboken, NJ, USA: Wiley, 2003.

[45] A. Al-Tamimi, F. L. Lewis, and M. Abu-Khalaf, "Discrete-time nonlinear HJB solution using approximate dynamic programming: Convergence proof," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 38, no. 4, pp. 943–949, Aug. 2008.

[46] D. Liu and Q. Wei, "Policy iteration adaptive dynamic programming algorithm for discrete-time nonlinear systems," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 25, no. 3, pp. 621–634, Mar. 2014.

[47] Y. Yang, W. Gao, H. Modares, and C.-Z. Xu, "Robust actor-critic learning for continuous-time nonlinear systems with unmodeled dynamics," *IEEE Trans. Fuzzy Syst.*, early access, Apr. 26, 2021, doi: 10.1109/TFUZZ.2021.3075501.

[48] M. Mazouchi, Y. Yang, and H. Modares, "Data-driven dynamic multi-objective optimal control: An aspiration-satisfying reinforcement learning approach," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Apr. 22, 2021, doi: 10.1109/TNNLS.2021.3072571.

Yongliang Yang (Member, IEEE) received the B.S. degree in electrical engineering from Hebei University, Baoding, China, in 2011, and the Ph.D. degree in electrical engineering from the University of Science and Technology Beijing (USTB), Beijing, China, in 2018.

From 2015 to 2017, he was a Visiting Scholar with the Missouri University of Science and Technology, Rolla, MO, USA, sponsored by the China Scholarship Council. He was an Assistant Professor at USTB from 2018 to 2020. From 2020 to 2021, he was a Post-Doctoral Research Fellow with the State Key Laboratory of Internet of Things for Smart City, Faculty of Science and Technology, University of Macau, Taipa, Macau. He is currently an Associate Professor at USTB. His research interests include reinforcement learning theory, robotics, distributed optimization, and control for cyber-physical systems.

Dr. Yang was a recipient of the Best Ph.D. Dissertation of China Association of Artificial Intelligence, the Best Ph.D. Dissertation of USTB, the Chancellor's Scholarship in USTB, the Excellent Graduates Awards in Beijing, and the UM Macao Talent Programme in Macau. He also serves as a Reviewer for several international journals and conferences, including *Automatica*, IEEE TRANSACTIONS ON AUTOMATIC CONTROL, IEEE TRANSACTIONS ON CYBERNETICS, and IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS.

Bahare Kiumarsi (Member, IEEE) received the B.S. degree in electrical engineering from the Shahroud University of Technology, Shahroud, Iran, in 2009, the M.S. degree in electrical engineering from the Ferdowsi University of Mashhad, Mashhad, Iran, in 2013, and the Ph.D. degree in electrical engineering from The University of Texas at Arlington (UT Arlington), Arlington, TX, USA, in 2017.

In 2018, she was a Post-Doctoral Research Associate with the Coordinated Science Laboratory, University of Illinois at Urbana–Champaign, Champaign, IL, USA, for 11 months. She is currently an Assistant Professor with the Department of Electrical and Computer Engineering, Michigan State University, East Lansing, MI, USA. Her current research interests include machine learning in control, security of cyber-physical systems, game theory, and distributed control.

Dr. Kiumarsi was a recipient of the UT-Arlington N. M. Stelmakh Outstanding Student Research Award and the UT Arlington Graduate Dissertation Fellowship in 2017.

Hamidreza Modares (Senior Member, IEEE) received the B.Sc. degree from Tehran University, Tehran, Iran, in 2004, the M.Sc. degree from the Shahroud University of Technology, Shahroud, Iran, in 2006, and the Ph.D. degree from the University of Texas at Arlington (UTA), Arlington, TX, USA, in 2015.

From 2006 to 2009, he was with the Shahroud University of Technology as a Senior Lecturer. From 2015 to 2016, he was a Faculty Research Associate with UTA. From 2016 to 2018, he was an Assistant Professor with the Department of Electrical and Computer Engineering, Missouri University of Science and Technology, Rolla, MO, USA. He is currently an Assistant Professor with the Department of Mechanical Engineering, Michigan State University, East Lansing, MI, USA. He has authored several journal articles and conference papers on the design of optimal controllers using reinforcement learning. His current research interests include cyber-physical systems, machine learning, distributed control, robotics, and renewable energy microgrids.

Dr. Modares was a recipient of the Best Paper Award from the 2015 IEEE International Symposium on Resilient Control Systems, the Stelmakh Outstanding Student Research Award from the Department of Electrical Engineering, UTA, in 2015, and the Summer Dissertation Fellowship from UTA in 2015. He is also an Associate Editor of the IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS.

Chengzhong Xu (Fellow, IEEE) received the B.Sc. and M.Sc. degrees from Nanjing University, Nanjing, China, in 1986 and 1989, respectively, and the Ph.D. degree from The University of Hong Kong, Hong Kong, in 1993, all in computer science and engineering.

He was a Professor with Wayne State University, Detroit, MI, USA, and the Director of the Institute of Advanced Computing and the Shenzhen Institutes of Advanced Technologies, Chinese Academy of Sciences, Beijing, China, before he joined UM in 2019. He is currently the Dean of the Faculty of Science and Technology and the Interim Director of the Institute of Collaborative Innovation, University of Macau, Taipa, Macau, and a Chair Professor of computer and information science. He is also a Chief Scientist of the Key Project on Smart City of MOST, China, and a Principal Instigator of the Key Project on Autonomous Driving of FDCT, Macau. He has published two research monographs and more than 300 peer-reviewed articles in journals and conference proceedings; his papers received over 10k citations with an H-index of 54. He received more than 100 patents or PCT patents and spun off a business "Shenzhen Baidou Applied Technology" with dedication to location-based services and technologies. His main research interests lie in parallel and distributed computing and cloud computing, in particular, with an emphasis on resource management for system's performance, reliability, availability, power efficiency, and security, and in big data and data-driven intelligence applications in smart cities and self-driving vehicles. The systems of particular interest include distributed systems and the Internet, servers and cloud datacenters, scalable parallel computers, and wireless embedded devices and mobile edge systems.

Dr. Xu was the Best Paper Nominee or Awardee of the 2013 IEEE High Performance Computer Architecture (HPCA), the 2013 ACM High Performance Distributed Computing (HPDC), IEEE Cluster 2015, ICPP 2015, GPC 2018, UIC 2018, and AIMS 2019. He received the most prestigious "President's Awards for Excellence in Teaching" of Wayne State University in 2002. He serves or served on a number of journal editorial boards, including IEEE TRANSACTIONS ON COMPUTERS (TC), IEEE TRANSACTIONS ON CLOUD COMPUTING (TCC), IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS (TPDS), *Journal of Parallel and Distributed Computing* (JPDC), *Science China Information Science*, and *ZTE Communication*. He has been the Chair of the IEEE Technical Committee on Distributed Processing (TCDP) from 2015 to 2020.