International Journal of Control, Automation and Systems 16(1) (2018) 377-386
http://dx.doi.org/10.1007/s12555-017-0172-5
ISSN:1598-6446 eISSN:2005-4092
http://www.springer.com/12555

Reinforcement Q-learning based on Multirate Generalized Policy Iteration and Its Application to a 2-DOF Helicopter

Tae Yoon Chun, Jin Bae Park*, and Yoon Ho Choi

Abstract: In this paper, we propose a novel Q-learning method based on multirate generalized policy iteration (MGPI) for unknown discrete-time (DT) linear quadratic regulation (LQR) problems. Q-learning is an effective scheme for unknown dynamical systems because it does not require any knowledge of the system dynamics to solve optimal control problems. By applying the MGPI concept, which is an extension of basic GPI with multirate time horizon steps, a new Q-learning algorithm is proposed for solving the LQR problem. Further, it is proven that the proposed algorithm converges to an optimal solution *i.e.*, it learns the optimal control policy iteratively using the states and the control-input information. Finally, we employ the two degree-of-freedom helicopter model to verify the effectiveness of the proposed method and investigate its convergence properties.

Keywords: Adaptive optimal control, linear quadratic regulation, multirate generalized policy iteration, Q-learning.

1. INTRODUCTION

In recent years, considerable attention has been paid to the modeling and control design of helicopters because of the increase in the number of their potential applications, such as military reconnaissance and emergent transportation, as well as their scientific significance [1, 2]. For this reason, many reported studies have applied several control methodologies such as proportional-integral-derivative (PID) control [3], robust attitude regulation [2], adaptive particle swarm optimization based optimal linear quadratic regulation (LQR) control [4], sliding mode control [5, 6], and backstepping methods [7].

Among them, optimal control is one of the important technique used in control engineering for various applications to achieve the desired performance. By solving the Hamilton-Jacobi-Bellman (HJB) equation or the algebraic Riccati equation (ARE) for linear systems, an optimal controller, that maximizes or minimizes the performance indices, is obtained. However, it is often hard to obtain or solve these equations because these techniques usually rely on an accurate knowledge of the system dynamics, and even when solved backward-in-time, they often do not satisfy the requirement of real applications.

On the other hand, associated with reinforcement learning (RL) and optimal control, adaptive dynamic programming (ADP) proposed by Werbos [8], is a brain-like self-learning control methodology that solves the optimal control problems forward-in-time [9,10]. These methods usually consist of a series of iterative methods that find the optimal solution with adaptive behaviour. Especially, ADP learns the optimal controller without the knowledge of systems dynamics. Because of this advantage, it is a suitable method for controlling many applications such as helicopters, unmanned aerial vehicle (UAV), and robots.

According to [11], iterative ADP algorithms can be classified into J-learning and Q-learning. This classification comes from the difference of the value function and Q-function properties. In J-learning algorithm, the iterative value function is a function of the system states, which is implemented to approximate the solution of the HJB equation [11, 12]. Therefore, in order to obtain the optimal control gain, the J-learning algorithm requires the system model and utility functions [13]. In contrast, the Q-function in Q-learning algorithms is a function of both states and control inputs which already includes the information about the system action and the utility functions [12]. Therefore, Q-learning is a data-based iterative ADP algorithm for obtaining the optimal control, especially for unknown and model-free systems [12].

These iterative algorithms are usually developed based on policy iteration (PI) or value iteration (VI). Basically, PI and VI are classified by the difference in the approximate policy evaluation step. That is, while PI evaluates the exact value function, which needs an infinite number of recursions, VI iterates only one step of the recursion to

Manuscript received March 28, 2017; accepted May 9, 2017. Recommended by Associate Editor Do Wan Kim under the direction of Editor Euntai Kim.

Tae Yoon Chun and Jin Bae Park is with the School of Electrical and Electronic Engineering, Yonsei University, 50 Yonsei-ro, Seodaemun-gu, Seoul, Korea (e-mails: {yueyoon, jbpark}@yonsei.ac.kr). Yoon Ho Choi is with the Department of Electronic Engineering, Kyonggi University, 94-6 Yiui-dong, Yeongtong-gu, Suwon, Kyonggi-Do, Korea (e-mail: yhchoi@kyonggi.ac.kr).
* Corresponding author.

reduce the computational complexity. In the literature, a family of VI algorithms was proposed in [8, 14, 15] for solving optimal control problems for discrete-time (DT) systems without the need for an initial stabilizing policy. The relation between the Q-learning method and the model-based adaptive control for the LQR case is discussed in [16]. For a continuous time (CT) dynamic system, Q-learning based VI for an infinite-horizon discounted cost LQR is demonstrated in [17]. A PI based algorithm was developed for the DT LQR problem using a Q-function in [18]. Recently, Q-learning for nonlinear DT deterministic systems was studied in [12].

On the other hand, generalized policy iteration (GPI) was demonstrated as a class of iterative algorithms for solving decision-making problems in [19]. GPI contains PI and VI as special cases and has both advantages of PI and VI; it has faster convergence speed than VI and, when applied to dynamic systems for optimal control, does not need the admissible stabilizing policy required by PI. Despite these advantages, in the fields of optimal control, there are only a few studies on the GPI algorithms [20–23] and, to the best authors' knowledge, those GPI methods are not extended to Q-learning for model-free adaptive optimal control.

In [23], Chun et al. extended the GPI methods and proposed multirate GPI (MGPI). Here, MGPI have the ability to regulate the time and iteration horizons for trading off the computational burden caused by its sub-iterations against learning speed in time–any two MGPI algorithms generate the same results in the iteration domain as long as the products of the time and iteration horizons of each algorithm are equal. This implies that the time scale for learning can be arbitrarily larger than that for control to reduce the computational burden (low iteration horizon), or in other words, by sacrificing the computational efficiency (high iteration horizon), the time scale for control can be made arbitrarily smaller than that for learning (by reducing the base sampling period).

In this paper, we propose a novel Q-learning method based on MGPI. It has the advantages of MGPI mentioned above so, for a given learning period, the performance can be made equal for an arbitrary small control period by increasing the iteration horizon by the inverse of the same amount. Such kind of fast time-scale control is necessary especially when the proposed method is applied to the systems such as helicopters that are initially unstable and thus has to be properly controlled with sufficiently fast time-rate. Moreover, since the exploration signal is applied only at the first time instant of each learning period, the proposed Q-learning method improves the stability of the system during the online learning by applying the current policy *without exploration noise*. The convergence property of the proposed MGPI-based Q-learning is also proven in a similar way to our previous work [23]; its performance is finally investigated by numerical simulations with a 2-DOF helicopter model.

The rest of this paper is organized as follows: In Section 2, the preliminaries of the discounted cost LQR and the Bellman equation are provided. Section 3 introduces the Q-function and proposes a Q-learning based on MGPI. Further, using several equivalent matrix formulas, we provide its convergence property to the optimal solution. Section 4 presents the online implementation of the proposed Q-learning scheme without any knowledge of the system dynamics. We then apply this algorithm to the optimal control of the 2-DOF helicopter in Section 5. Finally, Section 6 concludes this paper.

2. PRELIMINARIES

In this section, we briefly describe the LQR problem for DT systems. Throughout this paper, we consider the following DT linear time-invariant dynamical system:

$$\begin{aligned} x_{k+1} &= Ax_k + Bu_k, \\ y_k &= Cx_k, \end{aligned} \tag{1}$$

where $x_k \in \mathbb{R}^n$ is the state, $u_k \in \mathbb{R}^m$ is the control input, and $y_k \in \mathbb{R}^p$ is the system output; $A \in \mathbb{R}^{n\times n}$ is the system matrix and $B \in \mathbb{R}^{n\times m}$ is the input coupling matrix, respectively. In the DT LQR problem, we consider the following infinite horizon performance index:

$$\begin{aligned} J(x_k;\{u_{k+l}\}_{l=0}^{\infty}) &= \sum_{l=0}^{\infty} \gamma^l \left(\|y_{k+l}\|^2 + u_{k+l}^T R u_{k+l} \right) \\ &= \sum_{l=0}^{\infty} \gamma^l \left(x_{k+l}^T S x_{k+l} + u_{k+l}^T R u_{k+l} \right), \end{aligned} \tag{2}$$

where $0 \le \gamma \le 1$ is the discount factor and $S := C^T C \in \mathbb{R}^{n\times n}$ and $R = R^T \in \mathbb{R}^{m\times m}$ are positive semi-definite and positive definite matrices, respectively. Further, the state vector x_{k+l} for any $l \in \mathbb{N}$ is generated by the system (1). In this paper, we assume that (A,B,C) is stabilizable and detectable.

For a given policy K, we define its value function V_K as

$$\begin{aligned} V_K(x_k) &= J(y_k;\{u_{k+l}\}_{l=0}^{\infty})|_{u_{k+l}=-Kx_{k+l}}, \\ &= \sum_{l=0}^{\infty} \gamma^l \left(x_{k+l}^T S x_{k+l} + u_{k+l}^T R u_{k+1} \right)|_{u_{k+l}=-Kx_{k+l}}, \end{aligned}$$

and a policy K is defined to be admissible if $V_K(x)$ is finite for all $x \in \mathbb{R}^n$. For notational convenience, $S + K^T R K$ is denoted by S_K, i.e., $S_K := S + K^T R K$, and A_K as $A_K := A - BK$.

Then, the value function V_K can be rewritten as the following Bellman equation:

$$V_K(x_k) = x_k^T S x_k + u_k^T R u_k + \gamma V_K(x_{k+1}). \tag{3}$$

In the LQR case, any value can be expressed in a quadratic in the state, *i.e.*, $V_K(x_k) = x_k^T P_K x_k$ for a positive semi-definite matrix $P \in \mathbb{R}^{n\times n}$. Then, the discounted cost LQR

Bellman equation becomes

$$x_k^T P_K x_k = x_k^T S_K x_k + \gamma x_{k+1}^T P_K x_{k+1},$$

which can also be expressed as the following Lyapunov matrix equation:

$$Ric_K(P_K) = 0, \tag{4}$$

where the Lyapunov operator $Ric_K(P)$ is defined as

$$Ric_K(P) := \gamma A_K^T P A_K - P + S_K. \tag{5}$$

Solving the LQR problem is equivalent to finding the optimal policy K^*, which minimizes the performance index J, and the corresponding optimal value function $V^* := V_{K^*}$. Here, the optimal solution (V^*, K^*) of the DT LQR problem can be characterized by the Bellman's optimality principle [9, 10], which states that the optimal value function $V^*(x)$ satisfies

$$V^*(x_k) = \min_{u_k}\{x_k^T S x_k + u_k^T R u_k + \gamma V^*(x_{k+1})\}, \quad \forall x_k \in \mathbb{R}^n. \tag{6}$$

Minimizing the right-hand side of (6) using the quadratic representation, the optimal policy u^* can be represented as

$$u^*(x) = -K^* x = -(R + \gamma B^T P^* B)^{-1} \gamma B^T P^* A x. \tag{7}$$

Moreover, substituting (7) into (6) yields the Lyapunov operator $\mathrm{Ric}_{K^*}(P^*) = 0$, which is expanded as follows:

$$\begin{aligned} &S - P^* + \gamma A^T P^* A \\ &- \gamma^2 A^T P^* B (R + \gamma B^T P^* B)^{-1} B^T P^* A = 0. \end{aligned} \tag{8}$$

3. Q-LEARNING BASED ON GENERALIZED POLICY ITERATION

In this section, we introduce a Q-function for the LQR problem and then using the definition and time-series extension of the Q-function, present our new MGPI-based Q-learning with its convergence analysis.

3.1. Derivation of Q-function

For a given admissible policy K, the Q-function $Q_K(x_k, u_k)$ is an action value function, meaning the value of the performance metric obtained when an arbitrary control action u_k is applied at the current state x_k and then follows control policy $u_{k+l} = -K x_{k+l}$ thereafter [19, 24]. From the Bellman equation (3), $Q_K(x_k, u_k)$ is defined as

$$\begin{aligned} &Q_K(x_k, u_k) \\ &:= x_k^T S x_k + u_k^T R u_k \\ &\quad + \gamma \sum_{l=1}^{\infty} \gamma^{l-1} \left(x_{k+l}^T S x_{k+l} + u_{k+l}^T R u_{k+1} \right) |_{u_{k+l} = -K x_{k+l}} \\ &= x_k^T S x_k + u_k^T R u_k + \gamma V_K(x_{k+1}). \end{aligned} \tag{9}$$

In a similar manner, for the optimal control policy u^*, the optimal Q-function $Q^*(x,u)$ can be defined as

$$Q^*(x_k, u_k) = x_k^T S x_k + u_k^T R u_k + \gamma V^*(x_{k+1}). \tag{10}$$

Hence, the optimal control policy u^* can be expressed in terms of the optimal Q-function Q^* as

$$u^*(x) = \arg\min\{Q^*(x,u) : u \in \mathbb{R}^m\} \tag{11}$$

To demonstrate the MGPI scheme, we rewrite (9) as

$$\begin{aligned} Q_K(x_k, u_k) =& x_k^T S x_k + u_k^T R u_k \\ &+ \sum_{l=1}^{h-1} \gamma^l x_{k+l}^T S_K x_{k+l} + \gamma^h \cdot V_K(x_{k+h}), \end{aligned} \tag{12}$$

with an arbitrary step $h \in \mathbb{N}$. In fact, (12) is of the exactly same form as (9), except that it extends $V_K(x_{k+1})$ up to h-steps using (3). This restatement of the Q-function is the key to derive our Q-learning scheme. Noting that both Q-functions and value functions for the LQR problem can also be always presented in a quadratic form [25], (12) can be rewritten in a matrix form by substituting $Q_K(x_k, u_k) = \begin{bmatrix} x_k^T & u_k^T \end{bmatrix} H \begin{bmatrix} x_k^T & u_k^T \end{bmatrix}^T := z_k^T H z_k$ and $V_K(x_k) = x_k^T P x_k$ as

$$\begin{aligned} &Q_K(x_k, u_k) \\ &= z_k^T H z_k := \begin{bmatrix} x_k \\ u_k \end{bmatrix}^T \begin{bmatrix} H^{xx} & H^{xu} \\ H^{ux} & H^{uu} \end{bmatrix} \begin{bmatrix} x_k \\ u_k \end{bmatrix} \\ &= x_k^T S x_k + u_k^T R u_k \\ &\quad + \gamma \begin{bmatrix} x_k \\ u_k \end{bmatrix}^T \underbrace{\begin{bmatrix} A^T \\ B^T \end{bmatrix} S_K \begin{bmatrix} A^T \\ B^T \end{bmatrix}^T}_{= \begin{bmatrix} A^T S_K A & A^T S_K B \\ B^T S_K A & B^T S_K B \end{bmatrix}} \begin{bmatrix} x_k \\ u_k \end{bmatrix} \\ &\quad \vdots \\ &\quad + \gamma^{h-1} \begin{bmatrix} x_k \\ u_k \end{bmatrix}^T \begin{bmatrix} A^T \\ B^T \end{bmatrix} (A_K^{h-2})^T S_K A_K^{h-2} \begin{bmatrix} A^T \\ B^T \end{bmatrix}^T \begin{bmatrix} x_k \\ u_k \end{bmatrix} \\ &\quad + \gamma^h \cdot \begin{bmatrix} x_k \\ u_k \end{bmatrix}^T \begin{bmatrix} A^T \\ B^T \end{bmatrix} (A_K^{h-1})^T P \; A_K^{h-1} \begin{bmatrix} A^T \\ B^T \end{bmatrix}^T \begin{bmatrix} x_k \\ u_k \end{bmatrix} \\ &= \begin{bmatrix} x_k \\ u_k \end{bmatrix}^T \left[\begin{bmatrix} S + \sum_{l=0}^{h-2} \gamma^{l+1} A^T (A_K^l)^T S_K A_K^l A & \sum_{l=0}^{h-2} \gamma^{l+1} A^T (A_K^l)^T S_K A_K^l B \\ \sum_{l=0}^{h-2} \gamma^{l+1} B^T (A_K^l)^T S_K A_K^l A & R + \sum_{l=0}^{h-2} \gamma^{l+1} B^T (A_K^l)^T S_K A_K^l B \end{bmatrix} \right. \\ &\qquad \left. + \gamma^h \cdot \begin{bmatrix} A^T (A_K^{h-1})^T P A_K^{h-1} A & A^T (A_K^{h-1})^T P A_K^{h-1} B \\ B^T (A_K^{h-1})^T P A_K^{h-1} A & B^T (A_K^{h-1})^T P A_K^{h-1} B \end{bmatrix} \right] \begin{bmatrix} x_k \\ u_k \end{bmatrix}, \end{aligned} \tag{13}$$

where the matrix H have the following block-matrix components associated with P:

$$\begin{aligned} H^{xx} &= S + \gamma A^T \Pi_h A, & H^{xu} &= \gamma A^T \Pi_h B, \\ H^{ux} &= (H^{xu})^T = \gamma B^T \Pi_h A, & H^{uu} &= R + \gamma B^T \Pi_h B \end{aligned}$$

with Π_h is defined as

$$\Pi_h = \sum_{l=0}^{h-2} \gamma^l (A_K^l)^T S_K A_K^l + \gamma^{h-1} (A_K^{h-1})^T P A_K^{h-1}. \tag{14}$$

If $h = 1$, (12) is simplified to (10) and Π_h to P. Solving $\frac{\partial Q(x_k,u_k)}{\partial u_k} = 0$ to (13), we obtain the improved control policy

$$u_k = -(H^{uu})^{-1}H^{ux}x_k = -(R+\gamma B^T\Pi_h B)^{-1}\gamma B^T\Pi_h A x_k. \tag{15}$$

3.2. Q-learning algorithm

Based on the results of the previous subsection, we propose a Q-learning based on MGPI. The proposed Q-learning algorithm consists of a series of iterations between two successive steps of the policy evaluation and the policy improvement. Basically, the Q-learning algorithms in [17, 24–26] have been developed based on PI and VI schemes. The following discussion summarizes Q-learning schemes based on VI and PI, where the difference comes from the policy evaluation processes.

Algorithm: Q-learning based on VI

1 Approximate Policy Evaluation:

$$Q_{i+1}(x_k,u_k) = x_k^T S x_k + u_k^T R u_k + \gamma\, Q_i(x_{k+1},u_{k+1})$$

2 Policy Improvement:

$$u^{i+1}(x) = -K_{i+1}(x) = \arg\min_u Q_{i+1}(x_k,u)$$

Algorithm: Q-learning based on PI

1 Approximate Policy Evaluation:

$$Q_{i+1}(x_k,u_k) = x_k^T S x_k + u_k^T R u_k + \gamma\, Q_{i+1}(x_{k+1},u_{k+1})$$

2 Policy Improvement:

$$u^{i+1}(x) = -K_{i+1}(x) = \arg\min_u Q_{i+1}(x_k,u)$$

Meanwhile, the GPI scheme [19, 21, 27] has a sub-iteration in the policy evaluation step, and contains the above PI and VI scheme as special cases. Specifically, in this paper, the multirate concept is employed at the policy evaluation; we demonstrate the MPGI-based Q-learning by considering the j-th sub-iteration of the policy evaluation step. For $i = 1,2,3,...$, and $j = 0,1,...$, the MGPI algorithm can be described by the following two iterations i and j. The approximate policy evaluation of the j-th sub-iteration can then be described as follows:

$$\begin{aligned} Q_{i|j+1}(x_k,u_k) =& x_k^T S x_k + u_k^T R u_k + \sum_{l=1}^{M-1}\gamma^l x_{k+l}^T S_{K_i} x_{k+l} \\ &+ \gamma^M Q_{i|j}(x_{k+M}, -K_i x_{k+M}), \end{aligned}$$

where the index j increases from 0 to $N-1$. Here, M is the multirate index and j is the number of sub-iterations in the policy evaluation. For the j-th sub-iteration, the iterative Q-function is updated, while the control policy unchanged. After obtaining Q_{i+1} from this equation, the policy is update by policy improvement using Q_{i+1}. This process continues until convergence. Algorithm 1 shows the overall procedure of the proposed method.

Algorithm 1: Q-learning based on MGPI

1 **Initialize:**

- $K_0 \in \mathbb{R}^{m\times n}$, the initial policy;
- $H_0 \in \mathbb{R}^{(n+m)\times(n+m)}$, the initial Q function index;
- $N, M \in \mathbb{N}$, the iteration and time horizons;

2 $i \leftarrow 0$ (and $Q_0(z) := z^T H_0 z$);

3 **repeat**

Approximate Policy Evaluation:

4 $H_{i|0} \leftarrow H_i$ (and $Q_{i|0}(x,u) := Q_i(x,u)$);

5 **for** $j = 0,1,\cdots,N-1$ **do**

6 find $Q_{i|j+1}(z_k) = z_k^T H_{i|j+1} z_k$ such that $\forall z_k \in \mathbb{R}^{n+m}$:

$$\begin{aligned} Q_{i|j+1}(x_k,u_k) &= x_k^T S x_k + u_k^T R u_k \\ &+ \sum_{l=1}^{M-1}\gamma^l x_{k+l}^T S_{K_i} x_{k+l} + \gamma^M Q_{i|j}(x_{k+M}, -K_i x_{k+M}) \end{aligned} \tag{16}$$

where $x_{k+l+1} = A_{K_i} x_{k+l}$ $(l = 0,1,2,\cdots,M-1)$;

7 $Q_{i+1} \leftarrow Q_{i|N}$ (and $Q_{i+1}(z_k) = z_k^T H_{i+1} z_k$);

8 **Policy Improvement:** update the next policy K_{i+1} by

$$u^{i+1}(x) = -K_{i+1}(x) = \arg\min_u Q_{i+1}(x_k,u)$$

9 $i \leftarrow i+1$;

until *convergence is met.*

Because of the sub-iteration with index j and the multirate index M in the policy evaluation step, MGPI has the advantages of both PI and VI. That is, MGPI has a faster convergence speed than VI, and does not need the initial stabilizing condition that is usually required in the PI scheme. Note that (16) can be restated with new update horizon $L := M \times N$ as

$$\begin{aligned} Q_{i+1}(x_k,u_k) =& x_k^T S x_k + u_k^T R u_k + \sum_{l=1}^{L-1}\gamma^l x_{k+l}^T S_{K_i} x_{k+l} \\ &+ \gamma^L Q_i(x_{k+L}, -K_i x_{k+L}) \end{aligned}$$

for the time horizon M and the iteration horizon N, which is similar to the heuristic dynamic programming (HDP) and dual HDP (DHP) cases mentioned in our previous results on MGPI [23].

Remark 1: The proposed Q-learning (Algorithm 1) has special cases, PI ($N \to \infty$ and/or $M \to \infty$), VI ($N = M = 1$), and GPI ($M = 1$ with arbitrary N).

3.3. Convergence analysis

We now analyze the convergence of the proposed Q-learning algorithm. To do this, we first derive the equivalent matrix formulas for the approximate policy evaluation step and prove that these are equivalent to those of MGPI-based HDP and DHP. Then, we present the convergence of Q-learning based on the MGPI algorithm using these matrix formulas, as similarly proven in [23]. In this subsection, we assume that P_0 is positive semi-definite.

In the approximate policy evaluation step of Algorithm 1, $Q_{i|j+h}$, obtained by an arbitrary h-th iteration of (16), can be derived as

$$\begin{aligned} Q_{i|j+h}(x_k,u_k) =& x_k^T S x_k + u_k^T R u_k \\ &+ \sum_{l=1}^{hM-1} \gamma^l x_{k+l}^T S_{K_i} x_{k+l} \\ &+ \gamma^{hM} Q_{i|j}(x_{k+hM}, u_{k+hM}^i). \end{aligned} \tag{17}$$

Since z_{k+l} is defined as $z_{k+l} = \begin{bmatrix} x_{k+l}^T & u_{k+l}^T \end{bmatrix}^T$ for any $l \in \mathbb{N}$, x_{k+l} and u_{k+l} can be presented as

$$\begin{aligned} x_{k+l} &= A_{K_i}^l x_k = A_{K_i}^{l-1} x_{k+1}, \\ u_{k+l} &= -K_i x_{k+l} = -K_i A_{K_i}^{l-1} x_{k+1}. \end{aligned}$$

Then, (17) can be restated in a matrix form in terms of H, as shown in Lemma 1.

Lemma 1: Matrices $H_{i|j}$ and $H_{i|j+h}$ ($0 \le j \le j+h \le N$) obtained by the approximate policy evaluation in (16) satisfy the following matrix formula:

$$\begin{aligned} &H_{i|j+h} \\ &= \begin{bmatrix} S & 0 \\ 0 & R \end{bmatrix} + \begin{bmatrix} A^T \\ B^T \end{bmatrix} \sum_{l=1}^{hM-1} \gamma^l A_{K_i}^{l-1} S_{K_i} A_{K_i}^{l-1} \begin{bmatrix} A^T \\ B^T \end{bmatrix}^T \\ &\quad + \gamma^{hM} \cdot \begin{bmatrix} A_{K_i}^{hM-1}A & A_{K_i}^{hM-1}B \\ -K_i A_{K_i}^{hM-1}A & -K_i A_{K_i}^{hM-1}B \end{bmatrix}^T H_{i|j} \begin{bmatrix} A_{K_i}^{hM-1}A & A_{K_i}^{hM-1}B \\ -K_i A_{K_i}^{hM-1}A & -K_i A_{K_i}^{hM-1}B \end{bmatrix}. \end{aligned} \tag{18}$$

Proof: Note that $Q_{i|j+h}$ and $Q_{i|j}$ can be represented in quadratic forms as $Q_{i|j+h}(x_k,u_k) = \begin{bmatrix} x_k^T & u_k^T \end{bmatrix} H_{i|j+h} \begin{bmatrix} x_k^T & u_k^T \end{bmatrix}^T$ and $Q_{i|j}(x_{k+hM}, -K_i x_{k+hM}) = [x_{k+hM}^T \ -x_{k+hM}^T K_i^T] H_{i|j} [x_{k+hM}^T \ -x_{k+hM}^T K_i^T]^T$, respectively. Substituting these quadratic forms into (17) and using the same expansions used to obtain (13), and omitting x_k and u_k, we obtain the matrix equation (18), which completes the proof. □

Equation (18) shows the relation between matrices H_i and $H_{i|j+h}$ obtained by the h-number of sub-iteration of the policy evaluation. Now, let the matrix $P_{i|j}$ be defined as $P_{i|j} = \begin{bmatrix} I & -K_i^T \end{bmatrix} H_{i|j} \begin{bmatrix} I & -K_i^T \end{bmatrix}^T$. In what follows, we show a key matrix equality between the matrices $P_{i|j}$ and $P_{i|j+h}$ ($0 \le j < j+h \le N$).

Lemma 2: $P_{i|j+h}$ and $P_{i|j}$ ($0 \le j < j+h \le N$) satisfy

$$P_{i|j+h} = \sum_{l=0}^{hM-1} \gamma^l (A_{K_i}^l)^T S_{K_i} A_{K_i}^l + \gamma^{hM} (A_{K_i}^{hM})^T P_{i|j} A_{K_i}^{hM}. \tag{19}$$

Proof: Multiplying $\begin{bmatrix} I & -K_i^T \end{bmatrix}$ and $\begin{bmatrix} I & -K_i^T \end{bmatrix}^T$ in both sides of (18), we obtain by the definition of $P_{i|j+h}$

$$\begin{aligned} P_{i|j+h} &= \begin{bmatrix} I & -K_i^T \end{bmatrix} H_{i|j+h} \begin{bmatrix} I & -K_i^T \end{bmatrix}^T \\ &= S_{K_i} + \sum_{l=1}^{hM-1} \gamma^l A_{K_i}^l S_{K_i} A_{K_i}^l + \gamma^{hM} \\ &\quad \times (A_{K_i}^{hM})^T \begin{bmatrix} I \\ -K_i \end{bmatrix}^T H_{i|j} \begin{bmatrix} I \\ -K_i \end{bmatrix} A_{K_i}^{hM} \\ &= \sum_{l=0}^{hM-1} \gamma^l (A_{K_i}^l)^T S_{K_i} A_{K_i}^l + \gamma^{hM} (A_{K_i}^{hM})^T P_{i|j} A_{K_i}^{hM}, \end{aligned}$$

which completes the proof. □

Lemma 3: $P_{i|j+h}$ and $P_{i|j}$ ($0 \le j < j+h \le N$) satisfy the following matrix formulas:

$$1)\ \mathrm{Ric}_{K_i}(P_{i|j+h}) = \gamma^{hM} (A_{K_i}^{hM})^T \mathrm{Ric}_{K_i}(P_{i|j}) A_{K_i}^{hM}, \tag{20}$$

$$2)\ P_{i|j+h} = P_{i|j} + \sum_{l=0}^{hM-1} \gamma^l (A_{K_i}^l)^T \mathrm{Ric}_{K_i}(P_{i|j}) A_{K_i}^l. \tag{21}$$

Proof: See Appendix A. □

Remark 2: From Lemmas 1-3, one can notice that the approximate policy evaluation step of the Q-learning algorithm consists of the same matrix formulas as those of HDP and DHP based on MGPI.

The next result is our main theorem, which shows the convergence of the Q-learning based on MGPI.

Theorem 1: Assume that there exists $\alpha \in (0,1)$ such that $\sqrt{\gamma^L} \|A_{K_i}^L\| \le \alpha$ for all $i \in \mathbb{Z}_+$. Then, the sequences $\{H_i\}_{i=0}^{\infty}$ and $\{K_i\}_{i=0}^{\infty}$ generated by Algorithm 1 converge to the optimal ones, *i.e.*, $\lim_{i\to\infty} H_i = H^*$ and $\lim_{i\to\infty} K_i = K^*$.

Proof: After the N-th iteration of the approximate policy evaluation and policy improvement step, *i.e.*, $j = 0$, $h = N$, and $P_{i|N} = P_{i+1}$, we obtain by Lemma 3 the following matrix formulas:

$$\begin{aligned} &1)\ \mathrm{Ric}_{K_{i+1}}(P_{i+1}) \\ &\quad = \gamma^L (A_{K_i}^L)^T \mathrm{Ric}_{K_i}(P_i) A_{K_i}^L - \Delta K_i^T (R + \gamma B^T P_{i+1} B) \Delta K_i, \end{aligned} \tag{22}$$

$$2)\ P_{i+1} = P_i + \sum_{l=0}^{L-1} \gamma^l (A_{K_i}^l)^T \mathrm{Ric}_{K_i}(P_i) A_{K_i}^l, \tag{23}$$

where $\Delta K_i := K_{i+1} - K_i$.

Now, let define $\bar{A}$ and $\bar{B}$ as $\bar{A} := \sqrt{\gamma} A$ and $\bar{B} := \sqrt{\gamma} B$. Then, discounted cost LQR problem can be considered as an undiscounted LQR one, and the Lyapunov operator (5) can be restated as $\mathrm{Ric}_K(P) = \gamma A_K^T P A_K - P + S_K = \bar{A}_K^T P \bar{A}_K - P + S_K$. Furthermore, the matrix formulas (22) and (23) are restated as

$$\begin{aligned} 1)\ &\mathrm{Ric}_{K_{i+1}}(P_{i+1}) \\ &= \bar{A}_{K_i}^L \mathrm{Ric}_{K_i}(P_i) \bar{A}_{K_i}^L - \Delta K_i^T (R + \bar{B}^T P_{i+1} \bar{B}) \Delta K_i, \\ 2)\ &P_{i+1} = P_i + \sum_{l=0}^{L-1} (\bar{A}_{K_i}^l)^T \mathrm{Ric}_{K_i}(P_i) \bar{A}_{K_i}^l, \end{aligned}$$

which are the same equations of undiscounted cost case [23]. Therefore, by following the same proof procedure of Theorem 6 in [23], we can show that $\{P_i\}$ obtained by Algorithm 1 converge to the optimal matrix index P^*.

On the other hand, Lemma 1 and 2 state that iterating on H_i matrix is equivalent to iterating on P_i. Since $\{P_i\}$ is a convergent sequence to P^* and

$$H_i^{xx} = S + \gamma A^T P_i A,\ \ H_i^{xu} = \gamma A^T P_i B,\ \ H_i^{uu} = R + \gamma B^T P_i B,$$

H_i^{xx}, H_i^{xu}, and H_i^{uu} also converge to $S + \gamma A^T P^* A$, $\gamma A^T P^* B$, and $R + \gamma B^T P^* B$, respectively. Hence, $\{H_i\}$ is also a convergent sequence to H^*. This implies K_i converges to K^*, and the proof is complete. □

4. ONLINE IMPLEMENTATION

In this section, we present an online implementation of Q-learning based on MGPI. In Algorithm 1, H_{i+1} is calculated and updated at the end of the approximate policy evaluation step, and a new control policy K_{i+1} is updated at the policy improvement step based on matrix H_{i+1}. Generally, (16) is solved online by batch least squares (LS) or recursive LS (RLS). A batch LS based implementation is demonstrated in this paper.

Note that for any vector $a \in \mathbb{R}^{n_a}$ and matrix $G \in \mathbb{R}^{n_a \times n_b}$, the following relation is satisfied:

$$a^T G a = \bar{a}^T vec(G), \tag{24}$$

where $\bar{a} = [a_1^2, a_1 a_2, \cdots, a_1 a_{n_a}, a_2^2, a_2 a_3, \cdots a_2 a_{n_a}, \cdots a_{n_a} a_{n_a}]^T$ is the quadratic polynomial basis vector $\bar{a} \in \mathbb{R}^{n_a(n_a+1)/2}$. Further, $vec(\cdot)$ is the invertible map that converts an $n \times n$ symmetric matrix into a column vector in $\mathbb{R}^{n(n+1)/2}$ by stacking the upper triangular part of the matrix with the doubled off-diagonals [28].

To solve the LS problem for (16), data tuples z_k to z_{k+M} must be obtained and $Q_{i+1}(z_k)$ can be evaluated by the N-steps recursion of (16) in the approximate policy evaluation step. Using (24), $Q_{i|j}(z_k)$ is restated as $z_k^T H_{i|j} z_k = \bar{z}_k^T vec(H_{i|j})$. We can then represent the one-step recursion in (16) as

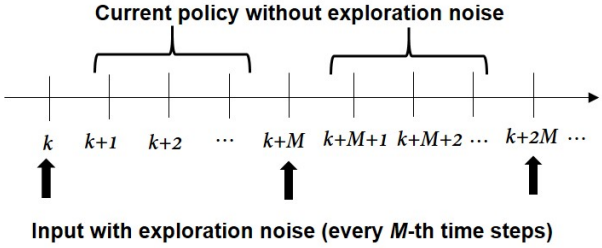

Fig. 1. Inputs with and without exploration noise.

$$\begin{aligned} \bar{z}_k^T vec(H_{i|j+1}) =& x_k^T S x_k + u_k^T R u_k \\ &+ \sum_{l=1}^{M-1} \gamma^l x_{k+l}^T S_{K_i} x_{k+l} \\ &+ \gamma^M \bar{z}_{k+M}^T vec(H_{i|j}), \end{aligned} \tag{25}$$

where $\bar{z}$ is the quadratic polynomial basis vector $\bar{z} \in \mathbb{R}^{(n+m)(n+m+1)/2}$. For the right hand side of (25), the term $x_k^T S x_k + u_k^T R u_k$ is the state performance metric obtained when u_k is applied and $\xi(x_{k+1:k+M}, H_{i|j})$ is obtained by the following control policy K_i thereafter, where

$$\begin{aligned} \xi(x_{k+1:k+M}, H_{i|j}) = & \sum_{l=0}^{M-1} \gamma^l x_{k+l}^T S_{K_i} x_{k+l} \\ &+ \gamma^M \bar{z}_{k+M}^T\ vec(H_{i|j}). \end{aligned} \tag{26}$$

In fact, (16) is a scalar equation and at least $f = (n+m)(n+m+1)/2$ data sets are required for solving the LS solution. If we then collect $s \geq f$ number of samples, matrices are obtained as follows:

$$Z_k = \begin{bmatrix} \bar{z}_k \\ \bar{z}_{k+M} \\ \vdots \\ \bar{z}_{k+sM} \end{bmatrix},\ \ \Xi(x_k, H_{i|j}) = \begin{bmatrix} \xi(x_{k+1:k+M}, H_{i|j}) \\ \xi(x_{k+M+1:k+2M}, H_{i|j}) \\ \vdots \\ \xi(x_{k+(s-1)M-1:k+sM}, H_{i|j}) \end{bmatrix}.$$

Then, the LS solution for $H_{i|j+1}$ in (16) becomes

$$vec(H_{i|j+1}) = (Z_k^T Z_k)^{-1} Z_k^T \Xi(x_k, H_{i|j}). \tag{27}$$

Finally, recursively iterating (27) until $j = N-1$, H_{i+1} is updated by the same rule (27) for $j = N-1$. Based on the LS solution for H_{i+1}, a new control policy is updated in the policy improvement, $u_{i+1}(x) = -(H_{i+1}^{uu})^{-1} H_{i+1}^{ux} x$.

Remark 3: To solve the LS problem (16) for H_{i+1}, the quadratic basis set $\bar{z}_k$ must satisfy the persistently excitation (PE) condition. A small exploratory signal consisting of sinusoids with different frequencies is usually added to the control input to ensure the PE condition is satisfied [25].

Remark 4: Since the exploration signal is applied only at the first time instant of learning period, the proposed method improves the stability of the system during the online learning by applying the input *without exploration noise* (See Fig. 1.)

5. APPLICATION TO A 2-DOF HELICOPTER MODEL

In this section, we apply the proposed algorithm to a 2-DOF helicopter model. As can be seen in Fig. 2, the 2-DOF helicopter (Quanser Consulting Inc., Canada) has two degrees of freedom: a motion around the yaw axis (Z-axis), and rotation around the pitch axis (Y-axis) represented by angles ψ and θ, respectively [4]. The nonlinear equation of motion of the helicopter system is obtained using the Euler Lagrangian energy based approach. That is, substituting the kinetic and potential energies, the differential equation of the 2-DOF helicopter system can be presented as [4]

$$\begin{aligned}&(J_{eq,p}+m_{heli}l_{cm}^2)\ddot{\theta}\\&=k_{pp}V_{mp}+k_{py}V_{my}-B_p\dot{\theta}\\&\quad-m_{heli}l_{cm}^2\sin(\theta)\cos(\theta)\dot{\psi}^2-m_{heli}gl_{cm}\cos(\theta),\end{aligned}\tag{28}$$

$$\begin{aligned}&(J_{eq,y}+m_{heli}\cos(\theta)^2l_{cm}^2)\ddot{\psi}\\&=k_{yy}V_{my}+k_{yp}V_{mp}-B_y\dot{\psi}+2m_{heli}l_{cm}^2\sin(\theta)\cos(\theta)\dot{\psi}.\end{aligned}\tag{29}$$

To design an LQR controller of the helicopter, the dynamics of the system (28)-(29) need to be represented in the form of the linear model by linearizing around the origin $\theta=0$, $\psi=0$, $\dot{\theta}=0$, and $\dot{\psi}=0$. Then, a state-space representation of the helicopter is given by

$$\begin{aligned}\dot{x}(t)&=A_cx(t)+B_cu(t),\\y(t)&=Cx(t),\end{aligned}$$

where the system states $x(t)$, control input $u(t)$, and matrices A_c, B_c, and C are defined as follows [29]:

$$x=\begin{bmatrix}\theta\\ \psi\\ \dot{\theta}\\ \dot{\psi}\end{bmatrix},\ u=\begin{bmatrix}V_{mp}\\ V_{my}\end{bmatrix},\ y=\begin{bmatrix}\theta\\ \psi\end{bmatrix},$$

$$A_c=\begin{bmatrix}0&0&1&0\\0&0&0&1\\0&0&-\frac{B_p}{J_{eqp}+m_{heli}l_{cm}^2}&0\\0&0&0&-\frac{B_y}{J_{eqy}+m_{heli}l_{cm}^2}\end{bmatrix},$$

$$B_c=\begin{bmatrix}0&0\\0&0\\\frac{K_{pp}}{J_{eqp}+m_{heli}l_{cm}^2}&\frac{K_{py}}{J_{eqp}+m_{heli}l_{cm}^2}\\\frac{K_{yp}}{J_{eqy}+m_{heli}l_{cm}^2}&\frac{K_{yy}}{J_{eqy}+m_{heli}l_{cm}^2}\end{bmatrix},C^T=\begin{bmatrix}1&0\\0&1\\0&0\\0&0\end{bmatrix}.$$

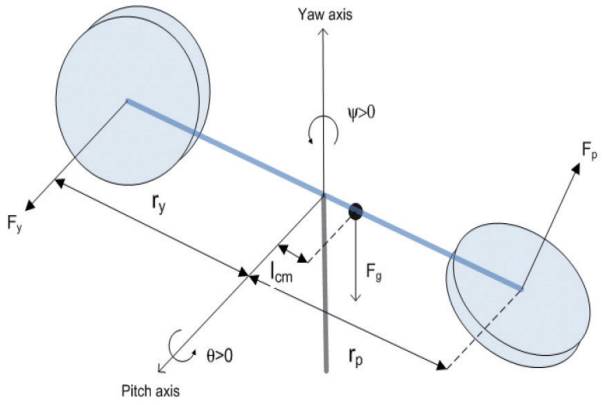

Fig. 2. Diagram of a 2-DOF helicopter [29].

Table 1. System parameters.

Symbol	Description	Value [Unit]
J_p	Total moment of inertia about the pitch axis	0.0384[kg m^2]
J_y	Total moment of inertia about the yaw axis	0.0432[kg m^2]
B_p	Equivalent viscous damping about the pitch axis	0.8 [N/V]
B_y	Equivalent viscous damping about the yaw axis	0.318 [N/V]
m_{heli}	Total moving mass of the helicopter	1.3872 [kg]
K_{pp}	Thrust torque constant of the yaw motor/propeller	0.204 [Nm/V]
K_{yy}	Thrust torque constant acting on the yaw axis from the yaw motor/propeller	0.072 [Nm/V]
K_{py}	Thrust torque constant acting on the pitch axis from the yaw motor/propeller	0.0068 [Nm/V]
K_{yp}	Thrust torque constant acting on the yaw axis from the pitch motor/propeller	0.0219 [Nm/V]
l_{cm}	Center of mass length along the helicopter body from the pitch axis	0.186 [m]

Table 1 shows the symbols, descriptions, and values of system parameters. Using zero-order hold discretization with 0.1 [s], the difference equation is obtained in the form of (1). For the purpose of simulation, we choose matrices S, R and discount factor γ as $S=\mathrm{diag}\{[20,15,10,20]\}$, $R=I_2$, and $\gamma=0.9$, respectively. Furthermore, we set the time and iteration horizon indices M and N as $M=2$ and $N=2$.

The optimal values of H^* and K^* are then obtained by

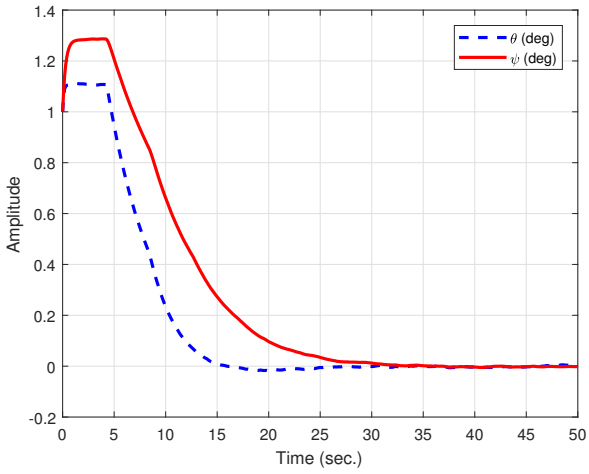

Fig. 3. Trajectories of outputs θ and ψ.

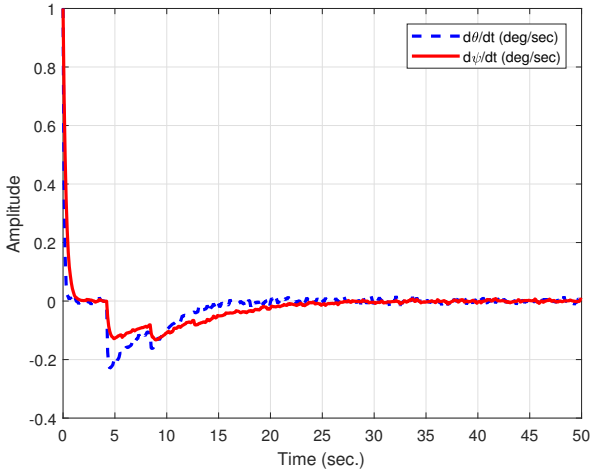

Fig. 4. Trajectories of outputs $d\theta/dt$ and $d\psi/dt$.

solving the DT ARE (8) and relation between P^* and H^*, *i.e.*,

$$H^* = \begin{bmatrix} 152.7263 & -4.0399 & 12.4843 & -2.0146 & 2.6134 & -0.0835 \\ -4.0399 & 129.7929 & -0.5165 & 22.2790 & 0.3506 & 1.5895 \\ 12.4843 & -0.5165 & 12.7879 & -0.2895 & 0.8704 & 0.0042 \\ -2.0146 & 22.2790 & -0.2895 & 39.3844 & 0.4475 & 1.7295 \\ 2.6134 & 0.3506 & 0.8704 & 0.4475 & 1.3077 & 0.0506 \\ -0.0835 & 1.5895 & 0.0042 & 1.7295 & 0.0506 & 1.1564 \end{bmatrix},$$

and

$$K^* = \begin{bmatrix} 1.9019 & 0.2043 & 0.6324 & 0.2703 \\ -0.1516 & 1.2950 & -0.0242 & 1.4070 \end{bmatrix}.$$

Figs. 3 and 4 present the states trajectories of the helicopter, *i.e.*, θ, ψ, $d\theta/dt$, and $d\psi/dt$. From the results shown in these figures, we identify that all states converge to zero as desired. Due to the advantages of the Q-learning scheme, knowledge about the system matrices is not required even the optimal solution is obtained.

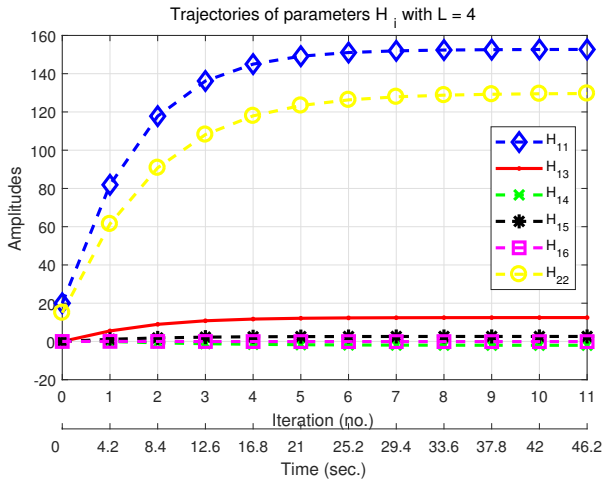

Fig. 5. Trajectories of H_{11}, H_{13}, H_{14}, H_{15}, H_{16}, and H_{22} with $L = 4$.

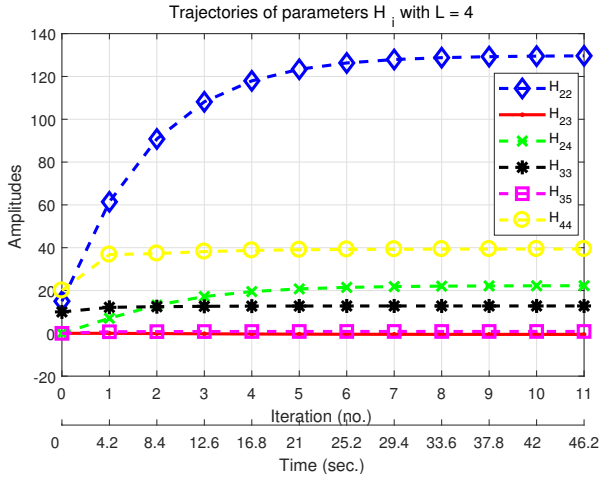

Fig. 6. Trajectories of H_{22}, H_{23}, H_{24}, H_{33}, H_{35}, and H_{44} with $L = 4$.

Figs. 5 and 6 present the trajectories of the parameters of H_i over time and iterations, respectively. These parameters converge to the optimal ones, which means that the optimal control gain has been obtained. In detail, after 11-th iterations, the approximate optimal values are obtained and converge to the corresponding optimal values.

To satisfy the PE condition of the LS problem, a exploration noise w is applied at the first time instant of each learning period, *i.e.*, $u = -Kx + w$. After all the parameters of H_i converge to the optimal ones ($\|H_i - H^*\| < \varepsilon$), a exploration noise is terminated.

6. CONCLUSION

In this paper, a novel Q-learning based on MGPI was developed for solving the discounted cost infinite hori-

zon optimal control for DT linear systems. The proposed method combines two advantages: 1) the ability to solve the optimal control problem without knowing any system dynamics and 2) a two-step iteration for the approximate policy evaluation that is free to choose the update horizon and the convergence speed to the optimal solution.

In the analysis of the equations of the approximate policy evaluation and improvement steps, the convergence property was also proved under certain conditions. Finally, a 2-DOF helicopter was simulated to demonstrate the effectiveness of the proposed approach.

APPENDIX A: PROOF OF LEMMA 3

Proof: From Lemma 2, we have

$$P_{i|j+h} = \sum_{l=0}^{hM-1} \gamma^l (A_{K_i}^l)^T S_{K_i} A_{K_i}^l + \gamma^{hM} (A_{K_i}^{hM})^T P_{i|j} A_{K_i}^{hM}.$$

Next, by the definition of the Lyapunov operator (5), we have $\mathrm{Ric}_{K_i}(P_{i|j}) = \gamma A_{K_i}^T P_{i|j} A_{K_i} - P_{i|j} + S_{K_i}$ and hence,

$$\begin{aligned} P_{i|j+h} &= \sum_{l=0}^{hM-1} \gamma^l (A_{K_i}^l)^T S_{K_i} A_{K_i}^l + \gamma^{hM} (A_{K_i}^{hM})^T P_{i|j} A_{K_i}^{hM} \\ &= \sum_{l=0}^{hM-2} \gamma^l (A_{K_i}^l)^T S_{K_i} A_{K_i}^l + \gamma^{hM-1} (A_{K_i}^{hM-1})^T \\ &\quad \times \left[\mathrm{Ric}_{K_i}(P_{i|j}) + P_{i|j}\right] A_{K_i}^{hM-1}. \end{aligned}$$

Repeating this procedure until the first summation term vanishes, we obtain (21). To prove (20), note that

$$\mathrm{Ric}_{K_i}(P_{i|j+h}) = \gamma A_{K_i}^T P_{i|j+h} A_{K_i} - P_{i|j+h} + S_{K_i} \qquad \text{(A.1)}$$

also holds by the definition in (5). Substituting (21) into (A.1), we have

$$\begin{aligned} &\mathrm{Ric}_{K_i}(P_{i|j+h}) \\ &= \gamma A_{K_i}^T \Big(\sum_{l=0}^{hM-1} \gamma^l (A_{K_i}^l)^T \mathrm{Ric}_{K_i}(P_{i|j}) A_{K_i}^l \Big) A_{K_i} \\ &\quad - \sum_{l=0}^{hM-1} \gamma^l (A_{K_i}^l)^T \mathrm{Ric}_{K_i}(P_{i|j}) A_{K_i}^l + \underbrace{\gamma A_{K_i}^T P_{i|j} A_{K_i} - P_{i|j} + S_{K_i}}_{=Ric_{K_i}(P_{i|j})} \\ &= \gamma^{hM} (A_{K_i}^{hM})^T \mathrm{Ric}_{K_i}(P_{i|j}) A_{K_i}^{hM} - Ric_{K_i}(P_{i|j}) + Ric_{K_i}(P_{i|j}) \\ &= \gamma^{hM} (A_{K_i}^{hM})^T \mathrm{Ric}_{K_i}(P_{i|j}) A_{K_i}^{hM}, \end{aligned}$$

which completes the proof. □

REFERENCES

[1] W. Gao, M. Huang, Z.-P. Jiang, and T. Chai, "Sampled-data-based adaptive optimal output-feedback control of a 2-degree-of-freedom helicopter," *IET Control Theory & Applications*, vol. 10, no. 12, pp. 1440-1447, 2016. [click]

[2] B. Zheng and Y. Zhong, "Robust attitude regulation of a 3-DOF helicopter benchmark: theory and experiments," *IEEE Trans. Ind. Electron.*, vol. 58, no. 2, pp. 660-670, 2011. [click]

[3] T. Bresciani, "Modelling, identification and control of a quadrotor helicopter," *MSc Theses*, 2008.

[4] E. V. Kumar, G. S. Raaja, and J. Jerome, "Adaptive PSO for optimal LQR tracking control of 2 DoF laboratory helicopter," *Applied Soft Computing*, vol. 41, pp. 77-90, 2016.

[5] G.-R. Yu and H.-T. Liu, "Sliding mode control of a two-degree-of-freedom helicopter via linear quadratic regulator," in *Proc. of IEEE International Conference on Systems, Man and Cybernetics*, vol. 4, pp. 3299-3304, IEEE, 2005.

[6] H. Ríos, A. Rosales, A. Ferreira, and A. Dávilay, "Robust regulation for a 3-dof helicopter via sliding-modes control and observation techniques," *Proceedings of the 2010 American Control Conference*, pp. 4427-4432, IEEE, 2010.

[7] Y. Yu, G. Lu, C. Sun, and H. Liu, "Robust backstepping decentralized tracking control for a 3-DOF helicopter," *Nonlinear Dynamics*, vol. 82, no. 1-2, pp. 947-960, 2015.

[8] P. J. Werbos, *A Menu of Designs for Reinforcement Learning Over Time*. MIT Press, Cambridge, 1990.

[9] 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, 2009. [click]

[10] J. Si, A. G. Barto, W. B. Powell, and D. Wunsch, *Handbook of learning and approximate dynamic programming*. Wiley-IEEE Press, 2004.

[11] J. M. Lee and J. H. Lee, "Approximate dynamic programming-based approaches for input-output data-driven control of nonlinear processes," *Automatica*, vol. 41, no. 7, pp. 1281-1288, 2005. [click]

[12] Q. Wei, F. L. Lewis, Q. Sun, P. Yan, and R. Song, "Discrete-time deterministic Q-learning: a novel convergence analysis," *IEEE Trans. Cybern.*, to be published, doi: 10.1109/TCYB.2016.2542923.

[13] L. Busoniu, R. Babuska, B. De Schutter, and D. Ernst, *Reinforcement learning and dynamic programming using function approximators*, vol. 39. CRC press, 2010.

[14] P. J. Werbos, "Approximate dynamic programming for real-time control and neural modeling," *Handbook of intelligent control: Neural, fuzzy, and adaptive approaches*, vol. 15, pp. 493-525, 1992.

[15] T. Y. Chun, J. Y. Lee, J. B. Park, and Y. H. Choi, "Integral temporal difference learning for continuous-time linear quadratic regulations," *International Journal of Control, Automation and Systems*, vol. 15, no. 1, pp. 226-238, 2017. [click]

[16] S. T. Hagen and B. Kröse, "Linear quadratic regulation using reinforcement learning," *Proc. Belgian Dutch Conf. Mech. Learn.*, pp. 39-46, 1998.

[17] M. Palanisamy, H. Modares, F. L. Lewis, and M. Aurangzeb, "Continuous-time Q-learning for infinite-horizon discounted cost linear quadratic regulator problems," *IEEE Transactions on Cybernetics*, vol. 45, no. 2, pp. 165-176, 2015. [click]

[18] S. J. Bradtke, B. E. Ydstie, and A. G. Barto, "Adaptive linear quadratic control using policy iteration," *Proc. of American Control Conference (ACC)*, vol. 3, pp. 3475-3479, 1994.

[19] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, Cambridge Univ Press, 1998.

[20] Q. Wei, D. Liu, and X. Yang, "Infinite horizon self-learning optimal control of nonaffine discrete-time nonlinear systems," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 4, pp. 866-879, 2015. [click]

[21] D. Liu, Q. Wei, and P. Yan, "Generalized policy iteration adaptive dynamic programming for discrete-time nonlinear systems," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 45, no. 12, pp. 1577-1591, 2015. [click]

[22] T. Y. Chun, J. Y. Lee, J. B. Park, and Y. H. Choi, "Stability and monotone convergence of generalized policy iteration for discrete-time linear quadratic regulations," *International Journal of Control*, vol. 89, no. 3, pp. 437-450, 2016. [click]

[23] T. Y. Chun, J. Y. Lee, J. B. Park, and Y. H. Choi, "Adaptive dynamic programming for discrete-time linear quadratic regulation based on multirate generalized policy iteration," *to appear in International Journal of Control(10.1080/00207179.2017.1312669.)*, 2016.

[24] B. Luo, D. Liu, T. Huang, and D. Wang, "Model-free optimal tracking control via critic-only Q-learning," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 27, no. 10, pp. 2134-2144, 2016. [click]

[25] A. Al-Tamimi, M. Abu-Khalaf, and F. L. Lewis, "Adaptive critic designs for discrete-time zero-sum games with application to H_∞ control," *IEEE Trans. Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 37, no. 1, pp. 240-247, 2007. [click]

[26] B. Kiumarsi, F. L. Lewis, H. Modares, A. Karimpour, and M.-B. Naghibi-Sistani, "Reinforcement Q-learning for optimal tracking control of linear discrete-time systems with unknown dynamics," *Automatica*, vol. 50, no. 4, pp. 1167-1175, 2014. [click]

[27] J. Y. Lee, J. B. Park, and Y. H. Choi, "On integral generalized policy iteration for continuous-time linear quadratic regulations," *Automatica*, vol. 50, no. 2, pp. 475-489, 2014. [click]

[28] J. W. Brewer, "Kronecker products and matrix calculus in system theory," *IEEE Trans. Circuits Syst.*, vol. 25, no. 9, pp. 772-781, 1978. [click]

[29] Q. Quanser, "2-DOF helicopter user and control manual," *Markham, Ontario*, 2006.

Tae Yoon Chun received his B.S., M.S., and Ph.D. degrees in Electrical and Electronic Engineering from Yonsei University, Seoul, Korea, in 2010, 2012, and 2017, respectively. His major research interests include approximate dynamic programming/reinforcement learning, optimal/adaptive control, synchrophasor, and power systems.

Jin Bae Park received the B.S. degree in electrical engineering from Yonsei University, Seoul, Korea, in 1977, and the M.S. and Ph.D. degrees in electrical engineering from Kansas State University, Manhattan, KS, USA, in 1985 and 1990, respectively. He has been with the Department of Electrical and Electronic Engineering, Yonsei University, since 1992, where he is currently a Professor. His current research interests include robust control and filtering, nonlinear control, drone, intelligent mobile robot, fuzzy logic control, neural networks, adaptive dynamic programming, and genetic algorithms. Dr. Park served as the Editor-in-Chief of the International Journal of Control, Automation, and Systems from 2006 to 2010, and the President of the Institute of Control, Robot, and Systems Engineers in 2013.

Yoon Ho Choi received his B.S., M.S., and Ph.D. degrees in electrical engineering from Yonsei University, Seoul, Korea, in 1980, 1982, and 1991, respectively. He was with the Department of Electrical Engineering, Ohio State University, Columbus, OH, USA, as a Visiting Scholar from 2000 to 2002 and from 2009 to 2010. He has been with the Department of Electronic Engineering, Kyonggi University, Suwon, Korea, since 1993, where he is currently a Professor. His current research interests include nonlinear control, intelligent control, multilegged and mobile robots, networked control systems, and ADP-based control. He was the Director of the Institute of Control, Robotics and Systems from 2003 to 2004 and from 2007 to 2008, where he also served as the Vice President from 2012 to 2015.