Automatica 78 (2017) 144–152

Contents lists available at ScienceDirect

Automatica

journal homepage: www.elsevier.com/locate/automatica

Brief paper

H_∞ control of linear discrete-time systems: Off-policy reinforcement learning☆

Bahare Kiumarsi [a], Frank L. Lewis [a,b], Zhong-Ping Jiang [c]

[a] UTA Research Institute UTARI, The University of Texas at Arlington, Ft. Worth, TX 76118, USA

[b] Consulting Professor, State Key Laboratory of Synthetical Automation for Process Industries, Northeastern University, Shenyang, China

[c] Control and Networks Lab, Department of Electrical and Computer Engineering, Polytechnic School of Engineering, New York University, Brooklyn, NY 11201, USA

ARTICLE INFO

Article history:
Received 22 October 2014
Received in revised form
13 September 2016
Accepted 22 November 2016

Keywords:
H_∞ control
Off-policy reinforcement learning
Optimal control

ABSTRACT

In this paper, a model-free solution to the H_∞ control of linear discrete-time systems is presented. The proposed approach employs off-policy reinforcement learning (RL) to solve the game algebraic Riccati equation online using measured data along the system trajectories. Like existing model-free RL algorithms, no knowledge of the system dynamics is required. However, the proposed method has two main advantages. First, the disturbance input does not need to be adjusted in a specific manner. This makes it more practical as the disturbance cannot be specified in most real-world applications. Second, there is no bias as a result of adding a probing noise to the control input to maintain persistence of excitation (PE) condition. Consequently, the convergence of the proposed algorithm is not affected by probing noise. An example of the H_∞ control for an F-16 aircraft is given. It is seen that the convergence of the new off-policy RL algorithm is insensitive to probing noise.

1. Introduction

The H_∞ control is a well-known robust control approach which is used to attenuate the effects of disturbances on the performance of dynamical systems (Doyle, Glover, Khargonekar, & Francis, 1989; Van der Schaft, 1992; Zames, 1981). It has a strong connection to the zero-sum game problem (Basar & Bernard, 1995), where the controller and the disturbance are considered as minimizing and maximizing players, respectively. Finding the solution to the zero-sum game problem leads to solving the game algebraic Riccati equation (GARE) for the linear systems. Numerical and iterative methods have been widely used to solve the GARE. However, they mostly require complete knowledge of the system dynamics.

Reinforcement learning (RL) has been applied for solving optimal control problems in an uncertain environment (Bertsekas & Tsitsiklis, 1996; Lewis, Vrabie, & Syrmos, 2012; Sutton & Barto, 1998; Werbos, 1989, 1990; Wu & Luo, 2012). For discrete-time (DT) systems, Q-learning algorithm was proposed to find the optimal control input without requiring any knowledge of the system dynamics (Bradtke, Ydstie, & Barto, 1994; Kiumarsi, Lewis, Modares, Karimpour, & Naghibi, 2014; Watkins, 1989). Q-learning algorithm has also been used to find the solution to the optimal control problem for systems with disturbances by solving the GARE (Al-Tamimi, Lewis, & Abu-Khalaf, 2007). Although elegant, there are two main problems with this algorithm. First, Q-learning requires the disturbance input to be updated in a prescribed manner. However, the disturbance input cannot be updated in a prescribed manner in more real-world applications. Second, Q-learning algorithm does not cancel out the effects of probing noise (which is used to excite the system) in the Bellman equation while evaluating the value function. This may result in bias and can affect the convergence of the algorithm.

To avoid these mentioned problems, in this paper, an off-policy RL algorithm (Sutton & Barto, 1998) is developed. In off-policy methods, two separate policies are used. The policy used to generate data, called the behavior policy, may in fact be unrelated to the policy that is evaluated and improved, called the estimation

☆ This work is supported by NSF grant ECCS-1405173, NSF grant IIS-1208623, ONR grant N00014-13-1-0562, ONR grant N000141410718. The work of Z.P. Jiang has been partially supported by NSF grants ECCS-1101401 and ECCS-1230040. The material in this paper was partially presented at the 7th International Conference on Cybernetics and Intelligent Systems (CIS) and IEEE Conference on Robotics, Automation and Mechatronics (RAM), July 15–17, 2015, Angkor Wat, Cambodia. This paper was recommended for publication in revised form by Associate Editor Raul Ordóñez under the direction of Editor Miroslav Krstic.

E-mail addresses: b_kiomarsi@yahoo.com (B. Kiumarsi), lewis@uta.edu (F.L. Lewis), zjiang@nyu.edu, zhongping_jiang@yahoo.com (Z.-P. Jiang).

http://dx.doi.org/10.1016/j.automatica.2016.12.009
0005-1098/

policy or target policy. Off-policy RL is presented for solving the optimal control problem of continuous-time (CT) systems with partially-unknown or completely-unknown dynamics (Jiang & Jiang, 2012; Li, Liu, & Wang, 2014; Luo, Huang, Wu, & Yang, 2015; Luo, Wu, & Huang, 2015; Luo, Wu, Huang, & Liu, 2014, 2015; Modares, Lewis, & Jiang, 2015).

To our knowledge, off-policy RL for DT systems has not been developed yet. Although Q-learning is originally off-policy, what is called Q-learning in control society is actually SARSA (Sutton & Barto, 1998), which is on-policy. Developing off-policy RL algorithms for DT systems is not straightforward because of the appearance of both system matrix A and control matrix B in the policy update equation. In this paper, an off-policy RL algorithm is presented to solve the H_∞ control of linear DT systems. The proposed method does not require any knowledge of the system dynamics, and, as such, is similar to other model-free RL algorithms, but it has at least two added advantages. First, the disturbance input is not required to be updated in a prescribed manner and this makes the proposed algorithm more practical in actual applications where the disturbance cannot be controlled. Second, adding probing noise does not cause incorrect solutions in the off-policy algorithm, as are liable to occur using on-policy RL algorithms.

2. Background

In this section, first, definitions of on-policy reinforcement learning and off-policy reinforcement learning are given. Then, the H_∞ control problem and its standard solution are presented. Finally, a policy iteration (PI) reinforcement learning algorithm is provided to solve the H_∞ control problem.

2.1. On-policy and off-policy reinforcement learning

Reinforcement learning (RL) methods are categorized into two classes: on-policy and off-policy. On-policy methods evaluate or improve the same policy as the one that is used to make decisions. Off-policy methods, on the other hand, evaluate one policy while following another policy. In other words, in off-policy methods, the policy which is used to generate data, called the behavior policy, may in fact be unrelated to the policy that is evaluated and improved, called the estimation policy or target policy.

Q-learning (Watkins, 1989) and SARSA (Sutton & Barto, 1998) are two methods of RL algorithms that use Q-function to evaluate a given policy. Q-learning is off-policy because it updates its Q-values using the Q-value of the next state and the greedy action. SARSA is on-policy because it updates its Q-values using the Q-value of the next state and the current policy action. However, what is called Q-learning in the control society is actually SARSA which is an on-policy method (Bradtke et al., 1994; Sutton & Barto, 1998).

2.2. Discrete-time H_∞ control problem

Consider the following linear discrete-time system

$$x_{k+1} = A\,x_k + B\,u_k + D\,w_k \tag{1}$$

where $x_k \in \mathbb{R}^n$ is the system state, $u_k \in \mathrm{R}^{m_1}$ is the control input, and $w_k \in \mathbb{R}^{m_2}$ is the external disturbance input.

Definition 1. The system (1) has L_2-gain less than or equal to γ if

$$\sum_{k=0}^{\infty}\left[x_k^{\mathrm{T}}Q\,x_k + u_k^{\mathrm{T}}R\,u_k\right] \le \gamma^2 \sum_{k=0}^{\infty} w_k^{\mathrm{T}} w_k \tag{2}$$

for all $w_k \in L_2[0,\infty)$, where Q and R are symmetric positive definite matrices, and $\gamma \ge 0$ is a prescribed constant disturbance attenuation level.

Note that functions in $L_2[0,\infty)$ represent the signals having finite energy over infinite interval $[0,\infty)$. That is,

$$\sum_{k=0}^{\infty} w_k^{\mathrm{T}} w_k < \infty.$$

The H_∞ control is to develop a control input such that the system (1) with $w_k = 0$ is asymptotically stable and it satisfies the disturbance attenuation condition (2). Based on (2), define the infinite horizon performance function as

$$\begin{aligned} J(x_k, u_k, w_k) &= \sum_{i=k}^{\infty} U_i \\ &= \sum_{i=k}^{\infty}\left[x_i^{\mathrm{T}}Q\,x_i + u_i^{\mathrm{T}}R\,u_i - \gamma^2 w_i^{\mathrm{T}} w_i\right]. \end{aligned} \tag{3}$$

Moreover, using (3), for an admissible control policy u_k and a disturbance policy w_k, the value function is defined as

$$V(x_k) = \sum_{i=k}^{\infty}[x_i^{\mathrm{T}}Q\,x_i + u_i^{\mathrm{T}}R\,u_i - \gamma^2 w_i^{\mathrm{T}} w_i]. \tag{4}$$

Assumption 1. The pair (A, B) is stabilizable and the pair $(A, \sqrt{Q})$ is observable.

2.3. Formulation of the H_∞ control as a zero-sum game problem

The H_∞ control problem can be expressed as a two-player zero-sum differential game in which the control policy player u_k seeks to minimize the value function, while the disturbance policy player w_k desires to maximize it. The goal is to find the feedback saddle point (u_k^*, w_k^*) such that

$$\begin{aligned} V^*(x_k) &= \min_{u_k}\max_{w_k} J(x_k, u_k, w_k) \\ &= \min_{u_k}\max_{w_k}\sum_{i=k}^{\infty}\left[x_i^{\mathrm{T}}Q\,x_i + u_i^{\mathrm{T}}R\,u_i - \gamma^2 w_i^{\mathrm{T}} w_i\right]. \end{aligned} \tag{5}$$

By using the value function (4), one has

$$\begin{aligned} V(x_k) &= x_k^{\mathrm{T}}Q\,x_k + u_k^{\mathrm{T}}R\,u_k - \gamma^2 w_k^{\mathrm{T}} w_k \\ &\quad + \sum_{i=k+1}^{\infty}\left[x_i^{\mathrm{T}}Q\,x_i + u_i^{\mathrm{T}}R\,u_i - \gamma^2 w_i^{\mathrm{T}} w_i\right] \end{aligned} \tag{6}$$

which yields the Bellman equation

$$V(x_k) = x_k^{\mathrm{T}}Q\,x_k + u_k^{\mathrm{T}}R\,u_k - \gamma^2 w_k^{\mathrm{T}} w_k + V(x_{k+1}). \tag{7}$$

It is known that for the system (1), the value function is quadratic as

$$V(x_k) = x_k^{\mathrm{T}}P\,x_k. \tag{8}$$

By using (8) in (7), the Bellman equation (7) becomes

$$x_k^{\mathrm{T}}P\,x_k = x_k^{\mathrm{T}}Q\,x_k + u_k^{\mathrm{T}}R\,u_k - \gamma^2 w_k^{\mathrm{T}} w_k + x_{k+1}^{\mathrm{T}}P\,x_{k+1}. \tag{9}$$

The Hamiltonian function is defined as

$$\begin{aligned} H(x_k, u_k, w_k) &= x_k^{\mathrm{T}}Q\,x_k \\ &\quad + u_k^{\mathrm{T}}R\,u_k - \gamma^2 w_k^{\mathrm{T}} w_k + V(x_{k+1}) - V(x_k). \end{aligned} \tag{10}$$

The optimal control policy u_k^* and the worst-case disturbance w_k^* should satisfy $\partial H(x_k, u_k, w_k)/\partial u_k = 0$ and $\partial H(x_k, u_k, w_k)/\partial w_k = 0$, respectively. Therefore, one has

$$u_k^* = -K_1^* x_k \tag{11}$$

$$w_k^* = -K_2^* x_k \tag{12}$$

where

$$K_1^* = (R + B^TPB + B^TPD(\gamma^2 I - D^TPD)^{-1}D^TPB)^{-1} \times (B^TPA + B^TPD(\gamma^2 I - D^TPD)^{-1}D^TPA) \quad (13)$$

$$K_2^* = (D^TPD - \gamma^2 I - D^TPB(R + B^TPB)^{-1}B^TPD)^{-1} \times (D^TPA - D^TPB(R + B^TPB)^{-1}B^TPA) \quad (14)$$

and P satisfies the game algebraic Riccati equation (GARE)

$$P = A^TPA + Q - \begin{bmatrix} A^TPB & A^TPD \end{bmatrix} \times \begin{bmatrix} R + B^TPB & B^TPD \\ D^TPB & D^TPD - \gamma^2 I \end{bmatrix}^{-1} \begin{bmatrix} B^TPA \\ D^TPA \end{bmatrix}. \quad (15)$$

It is shown in Basar and Bernard (1995), on one hand, that (13)–(15) solve the zero-sum game problem defined in (5), and, on the other hand, solving the zero-sum game problem defined in (5) is equivalent to finding a control policy that satisfies the disturbance attenuation condition (2). Therefore, the solution of (13)–(15) guarantees that the disturbance attenuation condition (2) is satisfied.

Remark 1. It is shown in Van der Schaft (1992) that there exists a γ^* such that for $\gamma < \gamma^*$, the H_∞ control problem has no solution. In Chen (2000), an explicit expression for the infimum of γ, i.e., γ^*, is found. It is shown that if $\gamma \geq \gamma^* \geq 0$, the GARE (15) has a unique positive semi-definite solution, the closed-loop system is asymptotically stable, and the disturbance attenuation condition is satisfied.

2.4. Online H_∞ PI algorithm

Various algorithms have been developed to solve the GARE (15). Policy iteration (PI) algorithm is one of the most used algorithms for solving the GARE (15) online which is as follows.

Algorithm 1. Online PI algorithm.

Initialization: Set the iteration number $j = 0$ and start with a stabilizing control policy u_k^0 and disturbance w_k^0.

1. Solve for P^{j+1} using the Bellman equation

$$x_k^TP^{j+1}x_k = x_k^TQ\,x_k + (u_k^j)^TR\,u_k^j - \gamma^2(w_k^j)^Tw_k^j + x_{k+1}^TP^{j+1}x_{k+1}. \quad (16)$$

2. Update the control and disturbance as

$$u_k^{j+1} = -K_1^{j+1}x_k = -(R + B^TP^{j+1}B + B^TP^{j+1}D(\gamma^2 I - D^TP^{j+1}D)^{-1}D^TP^{j+1}B)^{-1} \times (B^TP^{j+1}A + B^TP^{j+1}D(\gamma^2 I - D^TP^{j+1}D)^{-1}D^TP^{j+1}A)x_k \quad (17)$$

$$w_k^{j+1} = -K_2^{j+1}x_k = -(D^TP^{j+1}D - \gamma^2 I - D^TP^{j+1}B(R + B^TP^{j+1}B)^{-1}B^TP^{j+1}D)^{-1} \times (D^TP^{j+1}A - D^TP^{j+1}B(R + B^TP^{j+1}B)^{-1}B^TP^{j+1}A)x_k. \quad (18)$$

3. Stop if

$$\left|K_1^{j+1} - K_1^j\right| \leq \varepsilon \quad \text{and} \quad \left|K_2^{j+1} - K_2^j\right| \leq \varepsilon \quad (19)$$

for a small positive value of ε, otherwise set $j = j + 1$ and go to 1. ■

To implement Algorithm 1, the Bellman equation (16) can be written as

$$(x_k^T \otimes x_k^T - x_{k+1}^T \otimes x_{k+1}^T)vec(P^{j+1}) = x_k^TQ\,x_k + (u_k^j)^TR\,u_k^j - \gamma^2(w_k^j)^Tw_k^j. \quad (20)$$

Here, (20) is a scalar equation and $P^{j+1} \in \mathbb{R}^{n\times n}$ is a symmetric matrix with $n \times (n+1)/2$ independent elements. Therefore, at least $n \times (n+1)/2$ data sets are required to solve (20) using least squares (LS). To solve (20), one has $\zeta\, vec(P^{j+1}) = \upsilon$ where $\zeta = [\zeta_1^T \quad \cdots \quad \zeta_{q_1}^T]^T$with $\zeta_i = x_{k+i-1}^T \otimes x_{k+i-1}^T - x_{k+i}^T \otimes x_{k+i}^T$, $\upsilon = [\upsilon_1^T \quad \cdots \quad \upsilon_{q_1}^T]^T$with $\upsilon_i = x_{k+i-1}^TQ\,x_{k+i-1} + (u_{k+i-1}^j)^TR\,u_{k+i-1}^j - \gamma^2(w_{k+i-1}^j)^Tw_{k+i-1}^j$ and q_1 is the number of unknown elements of P^{j+1}. Matrix ζ must have independent rows.

The requirement of independent rows of these matrices is equivalent to the following condition.

Definition 2. A q-vector sequence $h = \begin{bmatrix} h_1 & \cdots & h_q \end{bmatrix}^T$is said to be persistently exciting over an interval $[k+1, k+l]$ if for some constant $\beta > 0$

$$\sum_{i=k+1}^{k+l} h_i\,h_i^T \geq \beta I. \quad (21)$$

Note that if $l < q$, (21) cannot be satisfied. ■

To satisfy the persistence of excitation (PE) condition (21) in Algorithm 1, probing noise is added to the system dynamics (1). To this end, the actual control input which is applied to the system to collect data is considered as

$$\hat{u}_k^j = u_k^j + e_k \quad (22)$$

with e_k being a probing noise or dither and u_k^j given by (17). The following lemma shows that probing noise may lead to incorrect solutions when solving the Bellman equation.

Lemma 1. *Effect of adding probing noise on the PI Algorithm.*

Let P^{j+1} be the solution to (16) with $e_k = 0$ in (22) and $\hat{P}^{j+1}$ be the solution to (16) with $e_k \neq 0$ in (22). Then, $P^{j+1} \neq \hat{P}^{j+1}$.

Proof. Let (16) be the undithered Bellman equation with $e_k = 0$ in (22), i.e. $\hat{u}_k^j = u_k^j$. On the other hand, using (22) with $e_k \neq 0$ in (16), the dithered Bellman equation yields

$$\begin{aligned} x_k^T\hat{P}^{j+1}x_k &= x_k^TQ\,x_k + (\hat{u}_k^j)^TR\,\hat{u}_k^j - \gamma^2(w_k^j)^Tw_k^j + \hat{x}_{k+1}^T\hat{P}^{j+1}\hat{x}_{k+1} \\ &= x_k^TQ\,x_k + (u_k^j + e_k)^TR\,(u_k^j + e_k) - \gamma^2(w_k^j)^Tw_k^j \\ &\quad + (Ax_k + B\,u_k^j + B\,e_k + D\,w_k^j)^T\hat{P}^{j+1} \\ &\quad \times (Ax_k + B\,u_k^j + B\,e_k + D\,w_k^j). \end{aligned} \quad (23)$$

By considering (1) in (23), one has

$$\begin{aligned} x_k^T\hat{P}^{j+1}x_k &= x_k^TQ\,x_k + (u_k^j)^TR\,u_k^j - \gamma^2(w_k^j)^Tw_k^j + x_{k+1}^T\hat{P}^{j+1}x_{k+1} \\ &\quad + e_k^T(R + B^T\hat{P}^{j+1}B)e_k + 2e_k^TR\,u_k^j + 2e_k^TB^T\hat{P}^{j+1}x_{k+1} \end{aligned} \quad (24)$$

which is the undithered Bellman equation (16) plus three terms depending on probing noise. Then, P^{j+1} is not the same as $\hat{P}^{j+1}$. ■

Remark 2. It is seen that the on-policy PI Algorithm 1 actually solves (24) online and hence obtains an incorrect solution $\hat{P}^{j+1}$ that is not the desired solution P^{j+1} to the Bellman equation (16). Since $\hat{P}^{j+1} \neq P^{j+1}$, this result shows that the control update (17) may not be correct if probing noise is added to Algorithm 1.

Remark 3. The online PI Algorithm 1 needs complete knowledge of the system dynamics to obtain the optimal control input and the worst-case disturbance input. In Al-Tamimi et al. (2007), a Q-learning algorithm was presented to solve the H_∞ control problem online without any knowledge of the system dynamics.

Remark 4. In Algorithm 1, the disturbance input must be updated in the prescribed optimal fashion (18) and applied to system dynamics to collect data. However, in practical applications, the disturbance is independent and cannot be specified. In Section 3, it is shown that the proposed method fixes this issue.

Remark 5. Algorithm 1 is the standard method for solving the discrete-time H_∞ optimal control problem online using RL. Note that if $w_k = 0$, for example, Algorithm 1 is the basis for the Heuristic Dynamic Programming (HDP) algorithm in Approximate Dynamic Programming (ADP). Lemma 1 shows that all standard approaches based on Algorithm 1 are vulnerable to bias caused by probing noise.

3. Off-policy RL algorithm for solving zero-sum game problem

In this section, an off-policy RL algorithm is presented to solve the zero-sum game problem arising in the H_∞ control problem. It is shown further that this off-policy algorithm does not suffer bias if probing noise is used.

3.1. Off-policy RL algorithm

To derive off-policy RL algorithm, the original system (1) is rewritten as

$$x_{k+1} = A_k x_k + B(K_1^j x_k + u_k) + D(K_2^j x_k + w_k) \tag{25}$$

where $A_k = A - BK_1^j - DK_2^j$.

In (25), the target policies are $u_k^j = -K_1^j x_k$ and $w_k^j = -K_2^j x_k$. They are the policies that are being learned and updated by the PI algorithm. By contrast, u_k and w_k are the behavior policies that are actually applied to the system dynamics (1) to generate data for learning.

For fixed policies u_k^j and w_k^j, the Bellman equation (7) yields

$$\begin{aligned} &V^{j+1}(x_k, u_k, w_k) - V^{j+1}(x_{k+1}, u_k, w_k) \\ &\quad = x_k^{\mathrm{T}} Q\, x_k + (u_k^j)^{\mathrm{T}} R\, u_k^j - \gamma^2 (w_k^j)^{\mathrm{T}} w_k^j. \end{aligned} \tag{26}$$

The Taylor expansion of the value function $V(x_k)$ at point x_{k+1} is

$$\begin{aligned} V(x_k) &= V(x_{k+1}) + (\nabla V)^{\mathrm{T}}(x_{k+1})(x_k - x_{k+1}) \\ &\quad + \frac{1}{2}(x_k - x_{k+1})^{\mathrm{T}} \nabla^2 V(x_{k+1})(x_k - x_{k+1}). \end{aligned} \tag{27}$$

By using (8) and (27), the left-hand side of (26) becomes

$$\begin{aligned} &V^{j+1}(x_k) - V^{j+1}(x_{k+1}) \\ &\quad = 2x_{k+1}^{\mathrm{T}} P^{j+1}(x_k - x_{k+1}) + (x_k - x_{k+1})^{\mathrm{T}} P^{j+1}(x_k - x_{k+1}). \end{aligned} \tag{28}$$

By substituting (25) in (28) and performing some manipulations, one has

$$\begin{aligned} &V^{j+1}(x_k) - V^{j+1}(x_{k+1}) = -x_k^{\mathrm{T}} A_k^{\mathrm{T}} P^{j+1} A_k x_k + x_k^{\mathrm{T}} P^{j+1} x_k \\ &\quad - (u_k + K_1^j x_k)^{\mathrm{T}} B^{\mathrm{T}} P^{j+1} x_{k+1} - (u_k + K_1^j x_k)^{\mathrm{T}} B^{\mathrm{T}} P^{j+1} A_k x_k \\ &\quad - (K_2^j x_k + w_k)^{\mathrm{T}} D^{\mathrm{T}} P^{j+1} x_{k+1} \\ &\quad - (K_2^j x_k + w_k)^{\mathrm{T}} D^{\mathrm{T}} P^{j+1} A_k x_k. \end{aligned} \tag{29}$$

On the other hand, the Bellman equation (9) can be written as the Lyapunov equation

$$Q - P^{j+1} + (K_1^j)^{\mathrm{T}} R\, K_1^j - \gamma^2 (K_2^j)^{\mathrm{T}} K_2^j + A_k^{\mathrm{T}} P^{j+1} A_k = 0. \tag{30}$$

Using (8) and (30) in (29) yields the following off-policy H_∞ Bellman equation (31). We now show that this equation can be iteratively solved to find the solution to the GARE (15) which gives the following off-policy RL algorithm.

Algorithm 2. Model-based off-policy RL.

Initialization: Set the iteration number $j = 0$ and start with a stabilizing control policy u_k.

1. Solve the following off-policy Bellman equation for $(P^{j+1}, K_1^{j+1}, K_2^{j+1})$ simultaneously

$$\begin{aligned} &x_k^{\mathrm{T}} P^{j+1} x_k - x_{k+1}^{\mathrm{T}} P^{j+1} x_{k+1} \\ &\quad = x_k^{\mathrm{T}} Q x_k + x_k^{\mathrm{T}} (K_1^j)^{\mathrm{T}} R K_1^j x_k - \gamma^2 x_k^{\mathrm{T}} (K_2^j)^{\mathrm{T}} K_2^j x_k \\ &\qquad - (u_k + K_1^j x_k)^{\mathrm{T}} B^{\mathrm{T}} P^{j+1} x_{k+1} \\ &\qquad - (u_k + K_1^j x_k)^{\mathrm{T}} B^{\mathrm{T}} P^{j+1} A_k x_k \\ &\qquad - (K_2^j x_k + w_k)^{\mathrm{T}} D^{\mathrm{T}} P^{j+1} x_{k+1} \\ &\qquad - (K_2^j x_k + w_k)^{\mathrm{T}} D^{\mathrm{T}} P^{j+1} A_k x_k. \end{aligned} \tag{31}$$

2. Stop if

$$\left| K_1^{j+1} - K_1^j \right| \le \varepsilon \quad \text{and} \quad \left| K_2^{j+1} - K_2^j \right| \le \varepsilon \tag{32}$$

for a small positive value of ε, otherwise set $j = j + 1$ and go to 1. ■

Note that (31) does not explicitly depend on K_1^{j+1} and K_2^{j+1}. Also, complete knowledge of the system dynamics is required for solving (31). In Section 3.2, it is shown in Algorithm 3 how to find $(P^{j+1}, K_1^{j+1}, K_2^{j+1})$ simultaneously by (31) without requiring any knowledge of the system dynamics.

The function of the on-policy Algorithm 1 is to solve the GARE (15) online in real-time. The next results show that Algorithm 2 also solves the GARE (15).

Theorem 1. *On-policy Algorithm 1 and off-policy Algorithm 2 are equivalent in the sense that (16) and (31) are equivalent.*

Proof. Substituting $A_k = A - BK_1^j - DK_2^j$ and system dynamics (1) in the off-policy Bellman equation (31), yields

$$\begin{aligned} &x_k^{\mathrm{T}} P^{j+1} x_k - (Ax_k + Bu_k + Dw_k)^{\mathrm{T}} \\ &\qquad \times P^{j+1}(Ax_k + Bu_k + Dw_k) \\ &\quad = x_k^{\mathrm{T}} Q\, x_k + x_k^{\mathrm{T}} (K_1^j)^{\mathrm{T}} R\, K_1^j x_k - \gamma^2 x_k^{\mathrm{T}} (K_2^j)^{\mathrm{T}} K_2^j x_k \\ &\qquad - (u_k + K_1^j x_k)^{\mathrm{T}} B^{\mathrm{T}} P^{j+1}(Ax_k + Bu_k + Dw_k) \\ &\qquad - (u_k + K_1^j x_k)^{\mathrm{T}} B^{\mathrm{T}} P^{j+1}(A - BK_1^j - DK_2^j) x_k \\ &\qquad - (w_k + K_2^j x_k)^{\mathrm{T}} D^{\mathrm{T}} P^{j+1}(Ax_k + Bu_k + Dw_k) \\ &\qquad - (w_k + K_2^j x_k)^{\mathrm{T}} D^{\mathrm{T}} P^{j+1}(A - BK_1^j - DK_2^j) x_k. \end{aligned} \tag{33}$$

By eliminating the common terms on the left-hand and right-hand sides of (33), one has

$$\begin{aligned} &x_k^{\mathrm{T}} P^{j+1} x_k - x_k^{\mathrm{T}} A^{\mathrm{T}} P^{j+1} A^{\mathrm{T}} x_k \\ &\quad = x_k^{\mathrm{T}} Q\, x_k + x_k^{\mathrm{T}} (K_1^j)^{\mathrm{T}} R\, K_1^j x_k - \gamma^2 x_k^{\mathrm{T}} (K_2^j)^{\mathrm{T}} K_2^j x_k \\ &\qquad + x_k^{\mathrm{T}} (K_1^j)^{\mathrm{T}} B^{\mathrm{T}} P^{j+1} B\, K_1^j x_k + x_k^{\mathrm{T}} (K_1^j)^{\mathrm{T}} B^{\mathrm{T}} P^{j+1} D\, K_2^j x_k \\ &\qquad - 2x_k^{\mathrm{T}} (K_2^j)^{\mathrm{T}} D^{\mathrm{T}} P^{j+1} A\, x_k + x_k^{\mathrm{T}} (K_2^j)^{\mathrm{T}} D^{\mathrm{T}} P^{j+1} B\, K_1^j x_k \\ &\qquad - 2x_k^{\mathrm{T}} (K_1^j)^{\mathrm{T}} B^{\mathrm{T}} P^{j+1} A\, x_k + x_k^{\mathrm{T}} (K_2^j)^{\mathrm{T}} D^{\mathrm{T}} P^{j+1} D\, K_2^j x_k. \end{aligned} \tag{34}$$

Eq. (34) can be rewritten as

$$\begin{aligned} &x_k^{\mathrm{T}} Q\, x_k + x_k^{\mathrm{T}} (K_1^j)^{\mathrm{T}} R\, K_1^j x_k - \gamma^2 x_k^{\mathrm{T}} (K_2^j)^{\mathrm{T}} K_2^j x_k - x_k^{\mathrm{T}} P^{j+1} x_k \\ &\quad + x_k^{\mathrm{T}} (A - B\, K_1^j - D\, K_2^j)^{\mathrm{T}} P^{j+1} (A - B\, K_1^j - D\, K_2^j) x_k = 0 \end{aligned} \tag{35}$$

which is equal to (16). Therefore, the on-policy Bellman equation (16) in Algorithm 1 and the off-policy Bellman equation (31) in Algorithm 2 are equivalent and the proof is completed. ■

Remark 6. In off-policy RL Algorithm 2, behavior policies applied to the system do not need to be the same as target policies which are improved and updated. In fact, in the proposed Algorithm 2, the policies u_k and w_k are behavior policies applied to the system dynamics (1) to collect data, while the policies $u_k^j = -K_1^j x_k$ and $w_k^j = -K_2^j x_k$ are the target policies and updated using measured data generated from the policies u_k and w_k. In Algorithm 2, u_k is assumed to be a stabilizing exploratory control policy and w_k is the external disturbance which is applied to the system. Therefore, the actual disturbance w_k applied to the system is not required to be updated in a prescribed manner according to $w_k^j = -K_2^j x_k$. This makes the proposed algorithm more practical than standard methods based on Algorithm 1.

Theorem 2. *Convergence of the off-policy RL Algorithm 2.*

The off-policy RL Algorithm 2 converges to the optimal control solution given by (13) and (14) where the matrix P satisfies the GARE (15).

Proof. The convergence is shown in two steps:

Step 1. In Theorem 1, it is shown that the off-policy Algorithm 2 is equivalent to Algorithm 1 at every iteration j.

Step 2. Substituting updated policies (17) and (18) into (35) yields

$$P^{j+1} = A^{\mathrm{T}}P^{j+1}A + Q - \begin{bmatrix} A^{\mathrm{T}}P^jB & A^{\mathrm{T}}P^jD \end{bmatrix} \times \begin{bmatrix} R + B^{\mathrm{T}}P^jB & B^{\mathrm{T}}P^jD \\ D^{\mathrm{T}}P^jB & D^{\mathrm{T}}P^jD - \gamma^2 I \end{bmatrix}^{-1} \begin{bmatrix} B^{\mathrm{T}}P^jA \\ D^{\mathrm{T}}P^jA \end{bmatrix}. \tag{36}$$

By using the result of Theorem 1, it can be concluded that iterating on (31) is equivalent to iterating on (36). In Stoorvogel and Weeren (1994), it is shown that iterating on (36) converges to the solution of GARE (15). Therefore, Algorithm 2 converges to the optimal solution. ■

To satisfy the PE condition, probing noise must be added to solve (16) in Algorithm 1 and (31) in Algorithm 2. Lemma 1 shows that this may result in incorrect solutions in Algorithm 1. The next result shows that adding probing noise does not lead to incorrect solutions while solving the Bellman equation (31) in the proposed Algorithm 2.

Theorem 3. *Effect of adding probing noise on off-policy RL Algorithm 2.*

Let $\hat{P}^{j+1}$ be the solution to (31) with $\hat{u}_k = u_k + e_k$ where $e_k \neq 0$ is the probing noise and $\bar{P}^{j+1}$ be the solution to (31) with $\bar{u}_k = u_k$. Then $\hat{P}^{j+1} = \bar{P}^{j+1}$.

Proof. 1. The off-policy Bellman equation (31) for control input $\hat{u}_k$ *is*

$$\begin{aligned} &x_k^{\mathrm{T}}\hat{P}^{j+1}x_k - (A_kx_k + B(K_1^j x_k + \hat{u}_k) + D(K_2^j x_k + w_k))^{\mathrm{T}}\hat{P}^{j+1} \\ &\quad \times (A_kx_k + B(K_1^j x_k + \hat{u}_k) + D(K_2^j x_k + w_k)) \\ &= x_k^{\mathrm{T}}Q\,x_k + x_k^{\mathrm{T}}(K_1^j)^{\mathrm{T}}RK_1^j x_k \\ &\quad - \gamma^2 x_k^{\mathrm{T}}(K_2^j)^{\mathrm{T}}K_2^j x_k - (\hat{u}_k + K_1^j x_k)^{\mathrm{T}} \\ &\quad \times B^{\mathrm{T}}\hat{P}^{j+1}(A_kx_k + B(K_1^j x_k + \hat{u}_k) + D(K_2^j x_k + w_k)) \\ &\quad - (\hat{u}_k + K_1^j x_k)^{\mathrm{T}}B^{\mathrm{T}}\hat{P}^{j+1}A_kx_k \\ &\quad - (K_2^j x_k + w_k)^{\mathrm{T}}D^{\mathrm{T}}\hat{P}^{j+1}A_kx_k - (K_2^j x_k + w_k)^{\mathrm{T}}D^{\mathrm{T}}\hat{P}^{j+1} \\ &\quad \times (A_kx_k + B(K_1^j x_k + \hat{u}_k) + D(K_2^j x_k + w_k)). \end{aligned} \tag{37}$$

Substituting $\hat{u}_k = u_k + e_k$ into (39) yields

$$\begin{aligned} &x_k^{\mathrm{T}}\hat{P}^{j+1}x_k - (A_kx_k + B(K_1^j x_k + u_k + e_k) \\ &\quad + D(K_2^j x_k + w_k))^{\mathrm{T}}\hat{P}^{j+1} \\ &\quad \times (A_kx_k + B(K_1^j x_k + u_k + e_k) + D(K_2^j x_k + w_k)) \\ &= x_k^{\mathrm{T}}Q\,x_k + x_k^{\mathrm{T}}(K_1^j)^{\mathrm{T}}RK_1^j x_k \\ &\quad - \gamma^2 x_k^{\mathrm{T}}(K_2^j)^{\mathrm{T}}K_2^j x_k - (u_k + e_k + K_1^j x_k)^{\mathrm{T}} \\ &\quad \times B^{\mathrm{T}}\hat{P}^{j+1}(A_kx_k \\ &\quad + B(K_1^j x_k + u_k + e_k) + D(K_2^j x_k + w_k)) \\ &\quad - (u_k + e_k + K_1^j x_k)^{\mathrm{T}}B^{\mathrm{T}}\hat{P}^{j+1}A_kx_k \\ &\quad - (K_2^j x_k + w_k)^{\mathrm{T}}D^{\mathrm{T}}\hat{P}^{j+1}A_kx_k \\ &\quad - (K_2^j x_k + w_k)^{\mathrm{T}}D^{\mathrm{T}}\hat{P}^{j+1} \\ &\quad \times (A_kx_k + B(K_1^j x_k + u_k + e_k) + D(K_2^j x_k + w_k)). \end{aligned} \tag{38}$$

Substituting (25) into (38) yields

$$\begin{aligned} &x_k^{\mathrm{T}}\hat{P}^{j+1}x_k - (x_{k+1} + B\,e_k)^{\mathrm{T}}\hat{P}^{j+1}(x_{k+1} + B\,e_k) \\ &= x_k^{\mathrm{T}}Q\,x_k + x_k^{\mathrm{T}}(K_1^j)^{\mathrm{T}}RK_1^j x_k - \gamma^2 x_k^{\mathrm{T}}(K_2^j)^{\mathrm{T}}K_2^j x_k \\ &\quad - (u_k + e_k + K_1^j x_k)^{\mathrm{T}}B^{\mathrm{T}}\hat{P}^{j+1}(x_{k+1} + B\,e_k) \\ &\quad - (u_k + e_k + K_1^j x_k)^{\mathrm{T}}B^{\mathrm{T}}\hat{P}^{j+1}A_kx_k \\ &\quad - (K_2^j x_k + w_k)^{\mathrm{T}}D^{\mathrm{T}}\hat{P}^{j+1}A_kx_k \\ &\quad - (K_2^j x_k + w_k)^{\mathrm{T}}D^{\mathrm{T}}\hat{P}^{j+1}(x_{k+1} + B\,e_k). \end{aligned} \tag{39}$$

Expanding terms in both sides of (39) yields

$$\begin{aligned} &x_k^{\mathrm{T}}\hat{P}^{j+1}x_k - x_{k+1}^{\mathrm{T}}\hat{P}^{j+1}x_{k+1} \\ &\quad - 2x_{k+1}^{\mathrm{T}}\hat{P}^{j+1}B\,e_k - e_k^{\mathrm{T}}B^{\mathrm{T}}\hat{P}^{j+1}B\,e_k \\ &= x_k^{\mathrm{T}}Qx_k + x_k^{\mathrm{T}}(K_1^j)^{\mathrm{T}}RK_1^j x_k - \gamma^2 x_k^{\mathrm{T}}(K_2^j)^{\mathrm{T}}K_2^j x_k \\ &\quad - (u_k + K_1^j x_k)^{\mathrm{T}}B^{\mathrm{T}}\hat{P}^{j+1}x_{k+1} \\ &\quad - (u_k + K_1^j x_k)^{\mathrm{T}}B^{\mathrm{T}}\hat{P}^{j+1}B\,e_k \\ &\quad - x_{k+1}^{\mathrm{T}}\hat{P}^{j+1}B\,e_k - e_k^{\mathrm{T}}B^{\mathrm{T}}\hat{P}^{j+1}B\,e_k \\ &\quad - (u_k + K_1^j x_k)^{\mathrm{T}}B^{\mathrm{T}}\hat{P}^{j+1}A_kx_k \\ &\quad - e_k^{\mathrm{T}}B^{\mathrm{T}}\hat{P}^{j+1}A_kx_k - (K_2^j x_k + w_k)^{\mathrm{T}}D^{\mathrm{T}}\hat{P}^{j+1}A_kx_k \\ &\quad - (K_2^j x_k + w_k)^{\mathrm{T}}D^{\mathrm{T}}\hat{P}^{j+1}x_{k+1} \\ &\quad - (K_2^j x_k + w_k)^{\mathrm{T}}D^{\mathrm{T}}\hat{P}^{j+1}B\,e_k. \end{aligned} \tag{40}$$

Eliminating the common terms and considering

$$\begin{aligned} x_{k+1}^{\mathrm{T}}\hat{P}^{j+1}B\,e_k = {} & x_k^{\mathrm{T}}A_k^{\mathrm{T}}\hat{P}^{j+1}Be_k + (u_k + K_1^j x_k)^{\mathrm{T}}B^{\mathrm{T}}\hat{P}^{j+1}B\,e_k \\ & + (K_2^j x_k + w_k)^{\mathrm{T}}D^{\mathrm{T}}\hat{P}^{j+1}B\,e_k \end{aligned} \tag{41}$$

in (45) yield

$$\begin{aligned} &x_k^{\mathrm{T}}\hat{P}^{j+1}x_k - x_{k+1}^{\mathrm{T}}\hat{P}^{j+1}x_{k+1} \\ &= x_k^{\mathrm{T}}Qx_k + x_k^{\mathrm{T}}(K_1^j)^{\mathrm{T}}RK_1^j x_k - \gamma^2 x_k^{\mathrm{T}}(K_2^j)^{\mathrm{T}}K_2^j x_k \\ &\quad - (u_k + K_1^j x_k)^{\mathrm{T}}B^{\mathrm{T}}\hat{P}^{j+1}x_{k+1} \\ &\quad - (u_k + K_1^j x_k)^{\mathrm{T}}B^{\mathrm{T}}\hat{P}^{j+1}A_kx_k \\ &\quad - (K_2^j x_k + w_k)^{\mathrm{T}}D^{\mathrm{T}}\hat{P}^{j+1}x_{k+1} \\ &\quad - (K_2^j x_k + w_k)^{\mathrm{T}}D^{\mathrm{T}}\hat{P}^{j+1}A_kx_k \end{aligned} \tag{42}$$

$\hat{P}^{j+1}$ is obtained by solving (42).

2. Substituting $\bar{u}_k = u_k$ into the off-policy Bellman equation (31) yields

$$\begin{aligned} & x_k^{\mathrm{T}}\bar{P}^{j+1}x_k - x_{k+1}^{\mathrm{T}}\bar{P}^{j+1}x_{k+1} \\ &= x_k^{\mathrm{T}}Qx_k + x_k^{\mathrm{T}}(K_1^j)^{\mathrm{T}}RK_1^jx_k - \gamma^2 x_k^{\mathrm{T}}(K_2^j)^{\mathrm{T}}K_2^jx_k \\ &\quad - (u_k + K_1^jx_k)^{\mathrm{T}}B^{\mathrm{T}}\bar{P}^{j+1}x_{k+1} \\ &\quad - (u_k + K_1^jx_k)^{\mathrm{T}}B^{\mathrm{T}}\bar{P}^{j+1}A_kx_k \\ &\quad - (K_2^jx_k + w_k)^{\mathrm{T}}D^{\mathrm{T}}\bar{P}^{j+1}x_{k+1} \\ &\quad - (K_2^jx_k + w_k)^{\mathrm{T}}D^{\mathrm{T}}\bar{P}^{j+1}A_kx_k \end{aligned} \tag{43}$$

$\bar{P}^{j+1}$ is obtained by solving (43).

By comparing (42) and (43), it can be concluded that $\hat{P}^{j+1}$ is the same as $\bar{P}^{j+1}$. Therefore, the off-policy Bellman equation (31) is insensitive to probing noise. ■

Remark 7. It was discussed in Section 2 that Algorithm 1 may result in a bias. This is because the control policy $\hat{u}_k^j = -K_1^jx_k + e_k$ is applied to the system dynamics to generate data while the value function is estimated for the control policy $u_k^j = -K_1^jx_k$. Therefore, measured state and input data used for learning are generated by a slightly different policy than the one under evaluation (while they are supposed to be the same in on-policy), which may cause a bias. On the other hand, Algorithm 2 is an off-policy RL algorithm, which allows us to separate the behavior policies (u_k, w_k) and target policies ($u_k^j = -K_1^jx_k$, $w_k^j = -K_2^jx_k$). Since the behavior policy is an arbitrary policy and unrelated to the estimated policy, as shown in Theorem 3, probing noise does not affect the estimation of the value for the policy under evaluation. In fact, probing noise added to the behavior policy is explicitly incorporated when solving the Bellman equation which leads to eliminating the bias.

3.2. Obtaining the optimal control input and the disturbance without system dynamics

Algorithm 2 requires the system dynamics to solve (31). In this section, the solution of Bellman equation (31) for $(P^{j+1}, K_1^{j+1}, K_2^{j+1})$ is presented in Algorithm 3. This solution does not require any knowledge of the system dynamics.

Based on Kronecker product, one has

$$a^{\mathrm{T}}W\,b = (b^{\mathrm{T}} \otimes a^{\mathrm{T}})vec(W) \tag{44}$$

with vectors $a \in \mathbb{R}^{va}$, $b \in \mathbb{R}^{vb}$, and matrix $W \in \mathbb{R}^{va\times vb}$. Then, by using (1), (44), and $A_k = A - BK_1^j - DK_2^j$, the off-policy Bellman equation (31) can be rewritten as

$$\begin{aligned} & (x_k^{\mathrm{T}} \otimes x_k^{\mathrm{T}})vec(P^{j+1}) - (x_{k+1}^{\mathrm{T}} \otimes x_{k+1}^{\mathrm{T}})vec(P^{j+1}) \\ &\quad + 2(x_k^{\mathrm{T}} \otimes (u_k + K_1^jx_k)^{\mathrm{T}})vec(B^{\mathrm{T}}P^{j+1}A) \\ &\quad - ((K_1^jx_k - u_k)^{\mathrm{T}} \otimes (u_k + K_1^jx_k)^{\mathrm{T}})vec(B^{\mathrm{T}}P^{j+1}B) \\ &\quad + 2(x_k^{\mathrm{T}} \otimes (w_k + K_2^jx_k)^{\mathrm{T}})vec(D^{\mathrm{T}}P^{j+1}A) \\ &\quad - ((K_1^jx_k - u_k)^{\mathrm{T}} \otimes (w_k + K_2^jx_k)^{\mathrm{T}})vec(D^{\mathrm{T}}P^{j+1}B) \\ &\quad + ((w_k - K_2^jx_k)^{\mathrm{T}} \otimes (u_k + K_1^jx_k)^{\mathrm{T}})vec(B^{\mathrm{T}}P^{j+1}D) \\ &\quad + ((w_k - K_2^jx_k)^{\mathrm{T}} \otimes (w_k + K_2^jx_k)^{\mathrm{T}})vec(D^{\mathrm{T}}P^{j+1}D) \\ &= x_k^{\mathrm{T}}Qx_k + x_k^{\mathrm{T}}(K_1^j)^{\mathrm{T}}RK_1^jx_k - \gamma^2 x_k^{\mathrm{T}}(K_2^j)^{\mathrm{T}}K_2^jx_k. \end{aligned} \tag{45}$$

Using LS method, the unique solution $(P^{j+1}, K_1^{j+1}, K_2^{j+1})$ can be obtained simultaneously and without any knowledge of the system dynamics. The Bellman equation (45) has $n^2 + m_1^2 + m_2^2 + 2m_1m_2 + n(m_1 + m_2)$ unknown parameters. Therefore, at least $n^2 + m_1^2 + m_2^2 + 2m_1m_2 + n(m_1 + m_2)$ data sets are required before (45) can be solved using LS at each iteration. For the positive integer $s \geq n^2 + m_1^2 + m_2^2 + 2m_1m_2 + n(m_1 + m_2)$, one defines

$$\phi^j = \begin{bmatrix} x_k^{\mathrm{T}}Q\,x_k + x_k^{\mathrm{T}}(K_1^j)^{\mathrm{T}}RK_1^jx_k - \gamma^2 x_k^{\mathrm{T}}(K_2^j)^{\mathrm{T}}K_2^jx_k \\ x_{k+1}^{\mathrm{T}}Q\,x_{k+1} + x_{k+1}^{\mathrm{T}}(K_1^j)^{\mathrm{T}}RK_1^jx_{k+1} - \gamma^2 x_{k+1}^{\mathrm{T}}(K_2^j)^{\mathrm{T}}K_2^jx_{k+1} \\ \vdots \\ x_{k+s-1}^{\mathrm{T}}Q\,x_{k+s-1} + x_{k+s-1}^{\mathrm{T}}(K_1^j)^{\mathrm{T}}RK_1^jx_{k+s-1} - \gamma^2 x_{k+s-1}^{\mathrm{T}}(K_2^j)^{\mathrm{T}}K_2^jx_{k+s-1} \end{bmatrix} \tag{46}$$

$$\psi^j = \begin{bmatrix} \mathrm{H}_{(xx)1} & \mathrm{H}_{(xu)1} & \mathrm{H}_{(uu)1} & \mathrm{H}_{(xw)1} & \mathrm{H}_{(uw)1} & \mathrm{H}_{(wu)1} & \mathrm{H}_{(ww)1} \\ \mathrm{H}_{(xx)2} & \mathrm{H}_{(xu)2} & \mathrm{H}_{(uu)2} & \mathrm{H}_{(xw)2} & \mathrm{H}_{(uw)2} & \mathrm{H}_{(wu)2} & \mathrm{H}_{(ww)2} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \mathrm{H}_{(xx)s} & \mathrm{H}_{(xu)s} & \mathrm{H}_{(uu)s} & \mathrm{H}_{(xw)s} & \mathrm{H}_{(uw)s} & \mathrm{H}_{(wu)s} & \mathrm{H}_{(ww)s} \end{bmatrix} \tag{47}$$

where

$$\begin{aligned} \mathrm{H}_{(xx)\mathrm{i}} &= x_{k+i-1}^{\mathrm{T}} \otimes x_{k+i-1}^{\mathrm{T}} - x_{k+i}^{\mathrm{T}} \otimes x_{k+i}^{\mathrm{T}} \\ \mathrm{H}_{(xu)\mathrm{i}} &= 2(x_{k+i-1}^{\mathrm{T}} \otimes (u_{k+i-1} + K_1^jx_{k+i-1})^{\mathrm{T}}) \\ \mathrm{H}_{(uu)\mathrm{i}} &= -(K_1^jx_{k+i-1} - u_{k+i-1})^{\mathrm{T}} \otimes (u_{k+i-1} + K_1^jx_{k+i-1})^{\mathrm{T}} \\ \mathrm{H}_{(xw)\mathrm{i}} &= 2(x_{k+i-1}^{\mathrm{T}} \otimes (w_{k+i-1} + K_2^jx_{k+i-1})^{\mathrm{T}}) \\ \mathrm{H}_{(uw)\mathrm{i}} &= -(K_1^jx_{k+i-1} - u_{k+i-1})^{\mathrm{T}} \otimes (w_{k+i-1} + K_2^jx_{k+i-1})^{\mathrm{T}} \\ \mathrm{H}_{(wu)\mathrm{i}} &= (w_{k+i-1} - K_2^jx_{k+i-1})^{\mathrm{T}} \otimes (u_{k+i-1} + K_1^jx_{k+i-1})^{\mathrm{T}} \\ \mathrm{H}_{(ww)\mathrm{i}} &= (w_{k+i-1} - K_2^jx_{k+i-1})^{\mathrm{T}} \otimes (w_{k+i-1} + K_2^jx_{k+i-1})^{\mathrm{T}}. \end{aligned}$$

Define the unknown variables in the Bellman equation (45), in which the control input and disturbance input gains depend on, as

$$\begin{aligned} & L_1^{j+1} = P^{j+1}, \quad L_2^{j+1} = B^TP^{j+1}A, \quad L_3^{j+1} = B^TP^{j+1}B \\ & L_4^{j+1} = D^TP^{j+1}A, \quad L_5^{j+1} = D^TP^{j+1}B, \\ & L_6^{j+1} = B^TP^{j+1}D \\ & L_7^{j+1} = D^TP^{j+1}D. \end{aligned}$$

Then, using (45)–(47), one has

$$\begin{aligned} & \psi^j \Big[vec(L_1^{j+1})^{\mathrm{T}}\, vec(L_2^{j+1})^{\mathrm{T}}\, vec(L_3^{j+1})^{\mathrm{T}}\, vec(L_4^{j+1})^{\mathrm{T}} \\ & \quad \times vec(L_5^{j+1})^{\mathrm{T}}\, vec(L_6^{j+1})^{\mathrm{T}}\, vec(L_7^{j+1})^{\mathrm{T}} \Big]^{\mathrm{T}} = \phi^j. \end{aligned} \tag{48}$$

Eq. (48) can be solved by LS method as

$$\begin{aligned} & \Big[vec(L_1^{j+1})^{\mathrm{T}}\, vec(L_2^{j+1})^{\mathrm{T}}\, vec(L_3^{j+1})^{\mathrm{T}}\, vec(L_4^{j+1})^{\mathrm{T}} vec(L_5^{j+1})^{\mathrm{T}} \\ & \quad \times\ vec(L_6^{j+1})^{\mathrm{T}}\, vec(L_7^{j+1})^{\mathrm{T}} \Big]^{\mathrm{T}} = ((\psi^j)^{\mathrm{T}}\psi^j)^{-1}(\psi^j)^{\mathrm{T}}\phi^j. \end{aligned} \tag{49}$$

Note that in (49), ψ^j and ϕ^j are known matrices and L_1^{j+1} through L_7^{j+1} are unknown values. This solution requires full rank of (47) which amounts to the PE condition and requires at least s time steps. s is a positive integer which is at least equal to the number of unknown parameters of the off-policy Bellman equation (45). That is, $s \geq n^2 + m_1^2 + m_2^2 + 2m_1m_2 + n(m_1 + m_2)$.

Using the solution of (49) for L_1^{j+1} through L_7^{j+1}, (17), and (18), the gains K_1^{j+1} and K_2^{j+1} can be obtained as

$$\begin{aligned} K_1^{j+1} &= (R + L_3^{j+1} + L_6^{j+1}(\gamma^2 I - L_7^{j+1})^{-1}L_5^{j+1})^{-1} \\ &\quad \times \left[L_2^{j+1} + L_6^{j+1}(\gamma^2 I - L_7^{j+1})^{-1}L_4^{j+1}\right] \end{aligned} \tag{50}$$

$$\begin{aligned} K_2^{j+1} &= (L_7^{j+1} - \gamma^2 I - L_5^{j+1}(R + L_3^{j+1})^{-1}L_6^{j+1})^{-1} \\ &\quad \times \left[L_4^{j+1} - L_5^{j+1}(R + L_3^{j+1})^{-1}L_2^{j+1}\right]. \end{aligned} \tag{51}$$

The following off-policy algorithm uses LS (49) and update laws (50) and (51) to find the solution to the H_∞ control problem of linear discrete-time systems.

Algorithm 3. Model-free off-policy RL.

Initialization: Set the iteration number $j = 0$ and start with a stabilizing control policy $u_k = -K_1 x + e_k$ where e_k is probing noise.

1. For $j = 0, 1, 2, \ldots$, solve (52) to obtain L_i^{j+1}, $i = 1, \ldots, 7$ using LS

$$\psi^j \left[vec(L_1^{j+1})^{\mathrm{T}}\, vec(L_2^{j+1})^{\mathrm{T}}\, vec(L_3^{j+1})^{\mathrm{T}}\, vec(L_4^{j+1})^{\mathrm{T}} \times vec(L_5^{j+1})^{\mathrm{T}}\, vec(L_6^{j+1})^{\mathrm{T}}\, vec(L_7^{j+1})^{\mathrm{T}} \right]^{\mathrm{T}} = \phi^j. \tag{52}$$

2. Update the control and disturbance gains using learned gains L_1^{j+1} through L_7^{j+1}

$$K_1^{j+1} = (R + L_3^{j+1} + L_6^{j+1}(\gamma^2 I - L_7^{j+1})^{-1} L_5^{j+1})^{-1} \times \left[L_2^{j+1} + L_6^{j+1}(\gamma^2 I - L_7^{j+1})^{-1} L_4^{j+1} \right] \tag{53}$$

$$K_2^{j+1} = (L_7^{j+1} - \gamma^2 I - L_5^{j+1}(R + L_3^{j+1})^{-1} L_6^{j+1})^{-1} \times \left[L_4^{j+1} - L_5^{j+1}(R + L_3^{j+1})^{-1} L_2^{j+1} \right]. \tag{54}$$

3. Stop if

$$\left| K_1^{j+1} - K_1^j \right| \le \varepsilon \quad \text{and} \quad \left| K_2^{j+1} - K_2^j \right| \le \varepsilon \tag{55}$$

for a small positive value of ε, otherwise set $j = j + 1$ and go to 1. ■

Remark 8. The proposed off-policy RL Algorithm 3 iteratively solves (49). This iterative algorithm does not require any knowledge of the system dynamics. The cost of control and disturbance policies are evaluated using measured data along the system trajectories. In fact, the algorithm has two steps. In the first step, in (52), the gains L_1^{j+1} through L_7^{j+1} are found using measured data ψ^j and ϕ^j (see (46) and (47)). In the second step, the control and disturbance policies are updated using the gains learned in the first step. Therefore, no knowledge of the system dynamics is required. Moreover, the disturbance policy which is specified and updated in (54) does not need to be applied to the system. This is in contrast to the existing RL and Q-learning methods that require this specified disturbance policy be applied to the system, which is not practical as the disturbance applied to the system cannot be specified.

4. Simulation results

In this section, the proposed scheme is used for control of an F-16 aircraft autopilot. Four cases are considered to show the effect of probing noise. In cases 1 through 3, different magnitudes of probing noise are considered with fix frequencies. In case 4, the frequencies of probing noise are changed. It is seen that the new off-policy H_∞ algorithm is insensitive to both magnitude and frequency of probing noise and always converges.

The F-16 short period dynamics has three states given as $x = \begin{bmatrix} \alpha & q & \delta_e \end{bmatrix}^{\mathrm{T}}$ where α is the angle of attack, q is the pitch rate, and δ_e is the elevator deflection angle. The discrete-time plant model of this aircraft dynamics is

$$x_{k+1} = Ax_k + Bu_k + Dw_k \tag{56}$$

where

$$A = \begin{bmatrix} 0.906488 & 0.0816012 & -0.0005 \\ 0.074349 & 0.90121 & -0.000708383 \\ 0 & 0 & 0.132655 \end{bmatrix}$$

$$B = \begin{bmatrix} -0.00150808 \\ -0.0096 \\ 0.867345 \end{bmatrix} \qquad D = \begin{bmatrix} 0.00951892 \\ 0.00038373 \\ 0 \end{bmatrix}.$$

The performance index is considered as (4) with $Q = \mathrm{diag}(1, 1, 1)$, $R = I$, and the disturbance attenuation $\gamma = 1$. Using (13) for the optimal control $u_k^* = -K_1^* x_k$ and (14) for the worst-case disturbance $w_k^* = -K_2^* x_k$, the gains K_1^* and K_2^* are given as

$$K_1^* = [-0.0842 \;\; -0.0961 \;\; 0.0661],$$

$$K_2^* = [-0.1477 \;\; -0.1244 \;\; 0].$$

Now the results of the proposed off-policy RL Algorithm 3 are given. The model-free off-policy RL Algorithm 3 is implemented as in (52)–(55). It is assumed that the dynamics A, B, and D are completely unknown. The initial state and the initial gains are chosen as $x_0 = \begin{bmatrix} 10 & -10 & -3 \end{bmatrix}^{\mathrm{T}}$, $K_1 = \begin{bmatrix} 3 & 2.5 & 1.1 \end{bmatrix}$, $K_2 = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}$.

In each iteration, 25 data samples are collected to perform the LS solution of the Bellman equations.

***Case* 1**: The probing noise is considered as

$$e_k = 0.2 \sin(1.009k) + \cos^2(0.538k) + \sin(0.9k) + \cos(100k).$$

After 5 iterations, the control and disturbance gains converge to

$$K_1 = [-0.0844 \;\; -0.096 \;\; 0.066], \quad K_2 = [-0.1477 \;\; -0.1244 \;\; 0].$$

Fig. 1 shows norm of the difference of the optimal control K_1^* gain and disturbance gain K_2^* and their computed values during the learning process. Fig. 2 shows the states of the system during and after learning with probing noise added up to time step 400. The probing noise is turned off after 400 time steps and the optimal control solution found by learning makes all states go to zero.

***Case* 2**: The probing noise is increased as

$$e_k = \sin(1.009k) + \cos^2(0.538k) + \sin(0.9k) + \cos(100k).$$

After 5 iterations, the control gain and the disturbance gain converge to

$$K_1 = [-0.0841 \;\; -0.096 \;\; 0.066], \quad K_2 = [-0.1477 \;\; -0.1244 \;\; 0].$$

Fig. 3 shows norm of the difference of the optimal control and disturbance gains K_1^* and K_2^* and their computed values during the learning process. Fig. 4 shows the states of the system during and after learning with probing noise added up to time step 400. The probing noise is turned off after 400 time steps and the optimal control solution found by learning makes all states go to zero.

***Case* 3**: The probing noise is increased as

$$e_k = 4 \sin(1.009k) + \cos^2(0.538k) + \sin(0.9k) + \cos(100k).$$

After 4 iterations, the control gain and the disturbance gain converge to

$$K_1 = [-0.0841 \;\; -0.096 \;\; 0.066], \quad K_2 = [-0.1476 \;\; -0.1245 \;\; 0].$$

Fig. 5 shows norm of the difference of the optimal control and disturbance gains K_1^* and K_2^* and their computed values during the learning process. In Fig. 6, the states of the system are shown during and after learning. The algorithm converges.

***Case* 4**: The frequencies of probing noise are changed as

$$e_k = 1 \sin(9.7k) + \cos^2(10.2k) + \sin(10k) + \cos(10k).$$

The control gain and the disturbance gain converge to

$$K_1 = [-0.0841 \;\; -0.096 \;\; 0.066], \quad K_2 = [-0.1477 \;\; -0.1244 \;\; 0].$$

Fig. 7 shows norm of the difference of the optimal control and disturbance gains K_1^* and K_2^* and their computed values during the learning process. In Fig. 8, the states of the system is shown during and after learning.

Fig. 9 shows the attenuation

$$\sum_{k=0}^{\infty} \left[x_k^{\mathrm{T}} Q\, x_k + u_k^{\mathrm{T}} R\, u_k \right] / \sum_{k=0}^{\infty} w_k^{\mathrm{T}} w_k$$

for the optimal control input and $w_k = \sin(k) e^{-0.01*k}$. It can be seen that the disturbance attenuation condition (2) is satisfied.

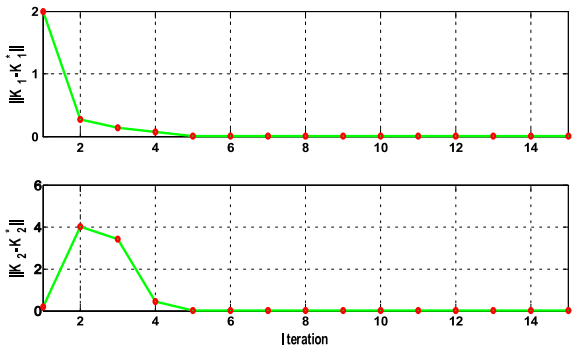

Fig. 1. Case 1: Convergence K_1 and K_2 in off-policy RL.

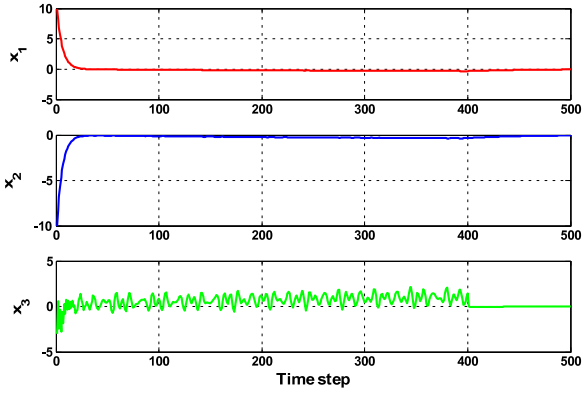

Fig. 2. Case 1: The system states in off-policy RL.

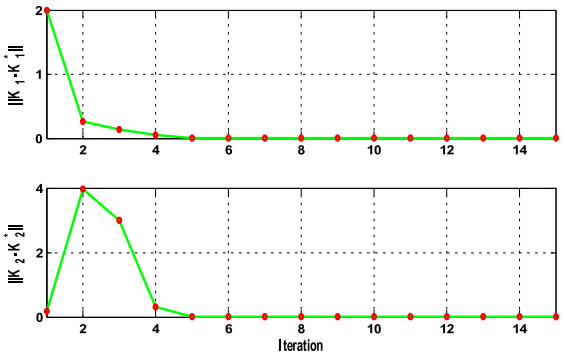

Fig. 3. Case 2: Convergence K_1 and K_2 in off-policy RL.

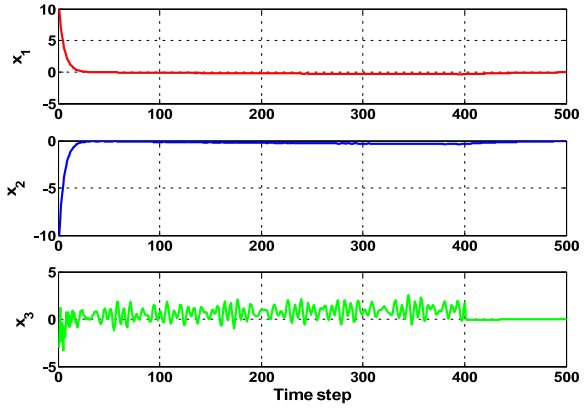

Fig. 4. Case 2: The system states in off-policy RL.

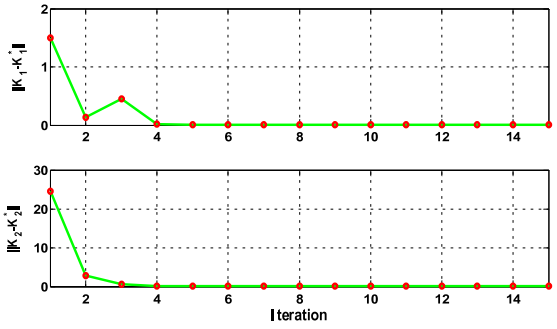

Fig. 5. Case 3: Convergence K_1 and K_2 in off-policy RL.

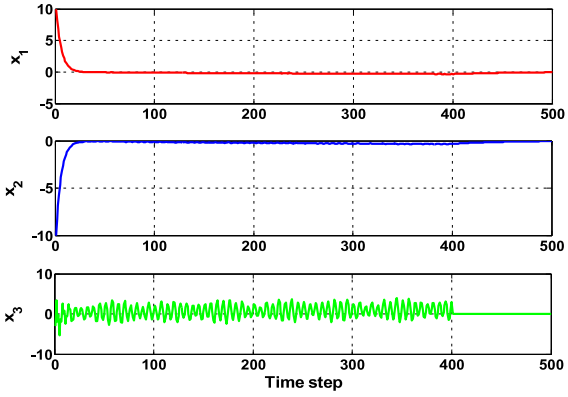

Fig. 6. Case 3: The system states in off-policy RL.

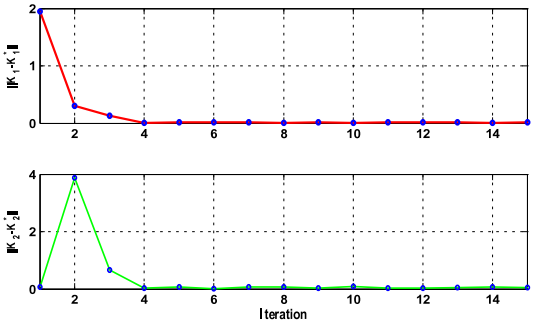

Fig. 7. Case 4: Convergence K_1 and K_2 in off-policy RL.

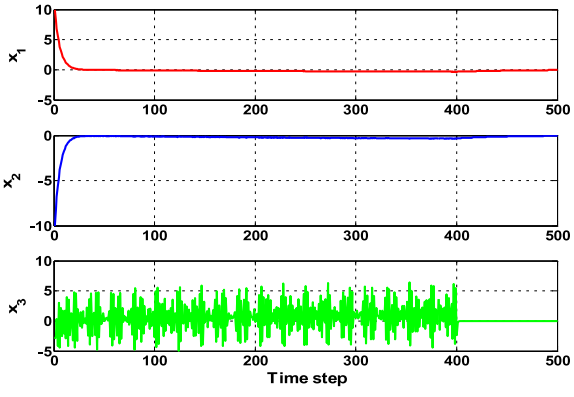

Fig. 8. Case 4: The system states in off-policy RL.

Remark 9. From the results of cases 1–4, it can be concluded the proposed off-policy algorithm converges to the optimal solution regardless of the level and frequency of the probing noise. This is in contrast to other model-free but on-policy RL approaches. This

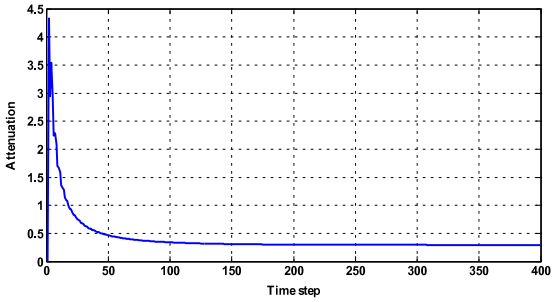

Fig. 9. Disturbance attenuation.

is because, if the magnitude of the probing noise is too small, the PE condition may not be satisfied and if the magnitude of the probing noise is too large, its covariance is increased and then based on Lemma 1, deleterious effect of the probing noise can be increased.

References

Al-Tamimi, A., Lewis, F. L., & Abu-Khalaf, M. (2007). Model-free Q-learning designs for linear discrete-time zero-sum games with application to H-infinity control. *Automatica*, *43*(3), 473–481.

Basar, T., & Bernard, P. (1995). *H_∞ optimal control and related minimax design problems*. Boston, MA: Birkhäuser.

Bertsekas, D. P., & Tsitsiklis, J. N. (1996). *Neuro-dynamic programming*. Belmont, Massachusetts: Athena Scientific.

Bradtke, S.J., Ydstie, B.E., & Barto, A.G. (1994). Adaptive linear quadratic control using policy iteration. In *Proceedings of IEEE American control conference, Baltimore, Maryland* (pp. 3475–3479).

Chen, B. M. (2000). *Robust and H_∞ control*. London: Springer -Veriag.

Doyle, J. C., Glover, K., Khargonekar, P. P., & Francis, B. A. (1989). State-space solutions to standard H2 and H_∞ control problems. *IEEE Transactions on Automatic Control*, *34*(8), 831–847.

Jiang, Y., & Jiang, Z. P. (2012). Computational adaptive optimal control for continuous-time linear systems with completely unknown dynamics. *Automatica*, *48*(10), 2699–2704.

Kiumarsi, B., Lewis, F. L., Modares, H., Karimpour, A., & Naghibi, M.-B. (2014). Reinforcement Q-learning for optimal tracking control of linear discrete-time systems with unknown dynamics. *Automatica*, *50*(4), 1167–1175.

Lewis, F. L., Vrabie, D., & Syrmos, V. (2012). *Optimal control* (3rd ed.). Hoboken, NJ: John Wiley & Sons, Inc..

Li, H., Liu, D., & Wang, D. (2014). Integral reinforcement learning for linear continuous-time zero-sum games with completely unknown dynamics. *IEEE Transactions on Automation Science and Engineering*, *11*(3), 706–714.

Luo, B., Huang, T., Wu, H., & Yang, X. (2015). Data-driven H_∞ control for nonlinear distributed parameter systems. *IEEE Transactions on Neural Networks and Learning Systems*, *26*(11), 2949–2961.

Luo, B., Wu, H., & Huang, T. (2015). Off-policy reinforcement learning for H_∞control design. *IEEE Transactions on Cybernetics*, *45*(1), 65–76.

Luo, B., Wu, H., Huang, T., & Liu, D. (2014). Data-based approximate policy iteration for affine nonlinear continuous-time optimal control design. *Automatica*, *50*(12), 3281–3290.

Luo, B., Wu, H., Huang, T., & Liu, D. (2015). Reinforcement learning solution for HJB equation arising in constrained optimal control problem. *Neural Networks*, *71*, 150–158.

Modares, H., Lewis, F. L., & Jiang, Z. P. (2015). Tracking control of completely unknown continuous-time systems via off-policy reinforcement learning. *IEEE Transactions on Neural Networks and Learning Systems*, *26*(10), 2550–2562.

Stoorvogel, A. A., & Weeren, A. J. T. M. (1994). The discrete-time Riccati equation related to the H_∞ control problem. *IEEE Transactions on Automatic Control*, *39*(3), 686–691.

Sutton, R. S., & Barto, A. G. (1998). *Reinforcement learning-an introduction*. Cambridge, MT, USA: MIT Press.

Van der Schaft, A. J. (1992). L_2-gain analysis of nonlinear systems and nonlinear state feedback H_∞control. *IEEE Transactions on Automatic Control*, *37*(6), 770–784.

Watkins, C. J. C. H. (1989). *Learning from delayed rewards.*. (Ph.D. thesis), England: University of Cambridge.

Werbos, P.J. (1989). Neural networks for control and system identification. In *Proceedings of IEEE conference on decision and control, Tampa, Florida* (pp. 260–265).

Werbos, P. J. (1990). A menu of designs for reinforcement learning over time. In *Neural networks for control* (pp. 67–95). Cambridge, MA, USA: MIT Press.

Wu, H., & Luo, B. (2012). Neural network based online simultaneous policy update algorithm for solving the HJI equation in nonlinear H_∞ control. *IEEE Transactions on Neural Networks and Learning Systems*, *23*(12), 1884–1895.

Zames, G. (1981). Feedback and optimal sensitivity: model reference transformations, multiplicative seminorms, and approximate inverses. *IEEE Transactions on Automatic Control*, *26*(2), 301–320.

Bahare Kiumarsi received the B.S. degree from Shahrood University of Technology, Iran, 2009 and the M.S. degree from Ferdowsi University of Mashhad, Iran, 2013. From August 2012 to December 2013, she was a Visiting Scholar with the University of Texas at Arlington, TX, USA, where she is currently working toward the Ph.D. degree. Her research interests include cooperative control systems, optimal control, reinforcement learning and neural networks.

Frank L. Lewis, is Member, National Academy of Inventors, Fellow IEEE, Fellow IFAC, Fellow UK Institute of Measurement & Control, PE Texas, UK Chartered Engineer. He is UTA Distinguished Scholar Professor, UTA Distinguished Teaching Professor, Moncrief-O'Donnell Chair at the University of Texas at Arlington Research Institute, and Qian Ren Thousand Talents Consulting Professor, Northeastern University, Shenyang, China. He obtained the Bachelor's Degree in Physics/EE and the MSEE at Rice University, the M.S. in Aeronautical Engineering from Univ. W. Florida, and the Ph.D. at Ga. Tech. He works in feedback control, intelligent systems, cooperative control systems, and nonlinear systems. He is author of 7 US patents, numerous journal special issues, journal papers, and 20 books, including *Optimal Control*, *Aircraft Control*, *Optimal Estimation*, and *Robot Manipulator Control* which are used as university textbooks worldwide. He received the Fulbright Research Award, NSF Research Initiation Grant, ASEE *Terman Award*, Int. Neural Network *Soc. Gabor Award*, UK Inst Measurement & Control *Honeywell Field Engineering Medal*, IEEE Computational Intelligence Society *Neural Networks Pioneer Award*, AIAA *Intelligent Systems Award*. He received Outstanding Service Award from Dallas IEEE section selected as Engineer of the year by Ft. Worth IEEE Section. He was listed in Ft. Worth Business Press Top 200 Leaders in Manufacturing. He received Texas Regents Outstanding Teaching Award 2013. He is Distinguished Visiting Professor at Nanjing University of Science & Technology and Project 111 Professor at Northeastern University in Shenyang, China. Founding Member of the Board of Governors of the Mediterranean Control Association.

Zhong-Ping Jiang received the B.Sc. degree in mathematics from the University of Wuhan, Wuhan, China, in 1988, the M.Sc. degree in statistics from the University of Paris XI, France, in 1989, and the Ph.D. degree in automatic control and mathematics from the Ecole des Mines de Paris (now, called ParisTech-Mines), France, in 1993, under the direction of Prof. Laurent Praly. He has been a Professor of Electrical and Computer Engineering at the Tandon School of Engineering, New York University. His main research interests include stability theory, robust/adaptive/distributed nonlinear control, adaptive dynamic programming and their applications to information, mechanical and biological systems. He is coauthor of the books: Stability and Stabilization of Nonlinear Systems (with Dr. I. Karafyllis, Springer, 2011), Nonlinear Control of Dynamic Networks (with Drs. T. Liu and D.J. Hill, Taylor & Francis, 2014), and Robust Adaptive Dynamic Programming (with Dr. Y. Jiang, IEEE-Wiley, 2017). He has written 14 book chapters, 182 published/accepted journal papers, and numerous conference papers. His work has received over 14,800 citations with an h-index of 62, by Google Scholar. Dr. Jiang is a Deputy co-Editor-in-Chief of the Journal of Control and Decision, an Editor for the International Journal of Robust and Nonlinear Control and he has served as an Associate Editor for several journals including Mathematics of Control, Signals and Systems (MCSS), Systems & Control Letters, IEEE Transactions on Automatic Control, European Journal of Control, and Science China: Information Sciences. Dr. Jiang is a recipient of the prestigious Queen Elizabeth II Fellowship Award from the Australian Research Council (1998), the CAREER Award from the US National Science Foundation (2001), JSPS Invitation Fellowship from the Japan Society for the Promotion of Science (2005), the Distinguished Overseas Chinese Scholar Award from the NSF of China (2007), and the Chair Professorship by the Ministry of Education of China (2009). His recent awards include the Best Theory Paper Award (with Y. Wang) at the 2008 WCICA, and the Guan Zhao Zhi Best Paper Award (with T. Liu and D. Hill) at the 2011 CCC, the Shimemura Young Author Prize (with his student Yu Jiang) at the 2013 Asian Control Conference in Istanbul, Turkey, and the Steve and Rosalind Hsia Best Biomedical Paper Award at the 2016 World Congress on Intelligent Control and Automation in Guilin, China. Prof. Jiang is a Fellow of the IEEE and a Fellow of the IFAC.