MANCHESTER 1824
The University of Manchester

The University of Manchester Research

Off-policy interleaved Q-learning: optimal control for affine nonlinear discrete-time systems

DOI:
10.1109/TNNLS.2018.2861945

Document Version
Accepted author manuscript

Link to publication record in Manchester Research Explorer

Citation for published version (APA):
Li, J., Chai, T., Lewis, F. W., Ding, Z., & Jiang, Y. (2018). Off-policy interleaved Q-learning: optimal control for affine nonlinear discrete-time systems. *IEEE Transactions on NEural Networks and Learning Systems*. https://doi.org/10.1109/TNNLS.2018.2861945

Published in:
IEEE Transactions on NEural Networks and Learning Systems

Citing this paper
Please note that where the full-text provided on Manchester Research Explorer is the Author Accepted Manuscript or Proof version this may differ from the final Published version. If citing, it is advised that you check and use the publisher's definitive version.

General rights
Copyright and moral rights for the publications made accessible in the Research Explorer are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

Takedown policy
If you believe that this document breaches copyright please refer to the University of Manchester's Takedown Procedures [http://man.ac.uk/04Y6Bo] or contact uml.scholarlycommunications@manchester.ac.uk providing relevant details, so we can investigate your claim.

OPEN ACCESS

Download date:24. Apr. 2024

To appear in IEEE Transactions on Neural Networks and Learning Systems

Off-policy interleaved Q-learning: optimal control for affine nonlinear discrete-time systems

Jinna Li, *Member, IEEE*, Tianyou Chai*, *Fellow, IEEE*, Frank L. Lewis, *Fellow, IEEE,* Zhengtao Ding, *Senior member, IEEE,* and Yi Jiang, *Student member, IEEE*

***Abstract*—In this paper, a novel off-policy interleaved Q-learning algorithm is presented for solving optimal control problem of affine nonlinear discrete-time (DT) systems, using only the measured data along the system trajectories. Affine nonlinear feature of systems, unknown dynamics and off-policy learning approach pose tremendous challenges on approximating optimal controllers. To this end, on-policy Q-learning method for optimal control of affine nonlinear DT systems is reviewed first, and its convergence is rigorously proven. The bias of solution to Q-function based Bellman equation caused by adding probing noises to systems for satisfying persistent excitation is also analyzed when using on-policy Q-learning approach. Then, a behavior control policy is introduced followed by proposing an off-policy Q-learning algorithm. Meanwhile, the convergence of algorithm and no bias of solution to optimal control problem when adding probing noise to systems are investigated. Third, three neural networks run by interleaved Q-learning approach in the actor-critic framework. Thus, a novel off-policy interleaved Q-learning algorithm is derived and its convergence is proven. Simulation results are given to verify the effectiveness of the proposed method.**

***Index Terms*—Q-learning, off-policy learning, affine nonlinear systems, interleaved learning, optimal control.**

I. INTRODUCTION

Reinforcement learning (RL), one of machine learning tools, has become a powerful and practical tool for tackling optimal control problems [1]-[4]. Increasingly large scale, high complexity of systems as well as growing requirements of cost, efficiency, energy, quality of products, etc. for practical industries, such as process industry, smart grid, smart residential energy systems, make data-driven control very promising for achieving optimum of control processes [5-8]. Q-learning, also known as action-dependent heuristic dynamic programming (ADHDP), is one of RL schemes, which combines adaptive critics, RL technique with dynamic programming to solve optimal control problems [8-19]. One of the strengths of Q-learning is that it is able to evaluate utility and update control policy without requiring models of the environment to be known a priori [9, 10].

It is well known that Q-learning has been studied for several decades aiming at Markov decision processes (MDP) [2, 4, 11-13], and the basic problem for which is to find a policy to minimize the expected cumulated costs (denoted by Q-function value) given state transition depending on only the present state-action pairs of the system, but not on its future and full past history. For the case of deterministic policy and deterministic state transition, increasing results using Q-learning to design an approximate optimal controller for the purpose of achieving optimum of control performance have been reported. For linear DT systems, [9, 10, 14, 15] solved H∞ control problem, optimal tracking control problem and optimal regulation problem using Q-learning. For linear continuous-time systems, [16-18] focused on the linear quadratic regulation problem and linear graphical game problem. Notice that the model-free optimal control for affine nonlinear systems using the Q-learning method has rarely been studied. This fact thus motivates this work for a better insight into how to design Q-learning algorithm to learn optimal controllers only using data for affine nonlinear systems.

Moreover, one can find that the above mentioned methods [8-10, 14-18] are implemented by using on-policy Q-learning approach. What kind of evaluating policy is called on-policy or off-policy? The essential difference between on-policy learning and off-policy learning lies on how to get data used for evaluating policy. If a target policy is evaluated using trajectories drawn from a behaviour policy not the target policy, then this learning method is referred to as off-policy learning. Otherwise, it is known as on-policy learning [4, 14-26]. Off-policy learning offers some advantages over on-policy learning with desired properties: (a) it resolves the exploration-exploitation dilemma. In fact, the arbitrary behaviour policy is applied to the systems to guarantee full data exploration, whereas the optimal exploitation policy, or the target policy, is actually learned; (b) probing noises are generally needed to guarantee persistent excitation (PE) condition, so that the optimal policy can be precisely learned. However, in on-policy learning, adding probing noises results in biased solutions [19]. On the other hand, in off-policy learning, adding probing noises does not result in biased solutions; (c) using off-policy learning mechanism for real systems is safer and more practical than on-policy learning,

This work is partly supported by the NSFC Projects under Grants 61673280, 61525302, 71602124, 61590922, 61503257, the Open Project of State Key Laboratory of Synthetical Automation for Process Industries under Grant PAL-N201603 and the Project of Liaoning Province under Grant LR2017006.

J. Li is with the School of Information and Control Engineering, Liaoning Shihua University, Liaoning 113001, P.R. China and also with the International Joint Research Laboratory of Integrated Automation, Northeastern University, Shenyang 110819, P.R. China. (lijinna_721@126.com)

T. Chai and Y. Jiang are with the State Key Laboratory of Synthetical Automation for Process Industries and the International Joint Research Laboratory of Integrated Automation, Northeastern University, Shenyang, 110819, P.R. China. (Corresponding author: T. Chai; tychai@mail.neu.edu.cn; JY369 356904@163.com)

F. Lewis is with the UTA Research Institute, the University of Texas at Arlington, Texas 76118, USA. He is also a Qian Ren Consulting Professor, the State Key Laboratory of Synthetical Automation for Process Industries and with the International Joint Research Laboratory of Integrated Automation, Northeastern University, Shenyang 110819, P.R. China. (lewis@uta.edu)

Z. Ding is with the School of Electrical & Electronic Engineering, the University of Manchester, Manchester M13 9PL, UK. (zhengtao.ding@ manch ester.ac.uk)

since there is potential risk, such as instability, high overshoot, etc., when the learned policies calculated by biased solutions in on-policy learning have to act at the systems.

Off-policy RL with the goal of finding the control policy for achieving optimal control of unknown dynamics has been attracted increasing attention in recent years. Included is for continuous-time (CT) systems [22-25], for DT systems [19-21, 26]. Even though the property of nonlinearity poses the great challenge on off-policy based RL for finding the optimal control policy without knowing the dynamics of systems, it is promising and practical since practical physical systems generally are nonlinear [5-7, 27]. To the best of our knowledge, off-policy Q-learning for affine nonlinear DT systems has not been fully developed yet. In this paper, an off-policy interleaved Q-learning algorithm is presented to solve the optimal control of affine nonlinear DT systems.

The contributions of this paper are summarized below:

1. Propose an off-policy Q-learning algorithm to approximate the optimal control policy for affine nonlinear DT systems. As opposed to on-policy Q-learning [8-10, 14-18], the off-policy Q-learning is investigated in this paper to handle the optimal control of affine nonlinear DT systems.

2. Prove no bias of solution to the optimal control problem for the first time from the perspective of off-policy Q-learning for affine nonlinear DT systems, which is the extension of [19-21] where the off-policy RL for linear DT systems was concerned. There exist two differences from [26] where an off-policy critic-only Q-learning algorithm was presented and one neural network was employed for solving the model-free optimal tracking control of nonlinear DT systems. One is that we develop a novel off-policy Q-learning algorithm by utilizing the relationship between Q function and value function. The other is that we present a rigorously theoretical proof on the unbiasedness of solution to the Q-function based iterative Bellman equation even though probing noises are added into systems for satisfying PE condition.

3. Develop an interleaved Q-learning approach for achieving approximate optimal control policy by interleaving iteration of critic network and actor network, which is different from the traditional policy iteration and value iteration approaches.

The rest of paper is given as follows. Section II devotes to on-policy Q-learning algorithm review and proving its convergence for optimal control of affine nonlinear DT systems. Section III presents an off-policy Q-learning algorithm and analyze its convergence and no bias of solution to the optimal control problem. In Section IV, an off-policy interleaved Q-learning algorithm is proposed by constructing three neural networks for implementing interleaved critic-actor iteration. Moreover, the rigorous proof of its convergence is presented. Section V verifies the effectiveness and no bias of solutions for the proposed method. Conclusions are stated in Section VI.

Notations: R^n denotes the n dimensional Euclidean space. $\otimes$ stands for the Kronecker product. tr(A) means the trace of matrix A, and vec(L) is used to turn any matrix L into a single column vector

II. PROBLEM STATEMENT

In this section, the optimal control problem of affine nonlinear DT systems is formulated and its standard solution by solving HJB equation is presented.

Consider the following affine nonlinear DT system

$$x_{k+1} = f(x_k) + g(x_k)u_k \tag{1}$$

where $x_k \in R^n$ and $u_k \in R^m$ are the state and control input, respectively, $f(x_k) \in R^n$ and $g(x_k) \in R^{n\times m}$. Without loss of generality, suppose that (1) is drift free, i.e., $f(0) = 0$ and $g(0) = 0$; (1) can be stabilized on a prescribed compact set $\Omega \in R^n$.

It is well known that it is the basic target for optimal control problem to find the control policy $u_k = u(x_k)$ which minimizes the infinite-horizon performance index expressed as

$$J(x_0) = \sum_{k=0}^{\infty} l(x_k, u_k) \tag{2}$$

where $l(x_k, u_k)$ is the utility function with $l(x_k, u_k) \geq 0$ for any x_k and u_k. In general, the utility function is chosen as a quadratic form $l(x_k, u_k) = x_k^{\mathrm{T}} Q x_k + u_k^{\mathrm{T}} R u_k$, where $Q \geq 0$ and $R > 0$ are respectively positive semi-definite matrix and positive definite matrix.

According to dynamic programming theory [26], the optimal value function should satisfy the DT HJB equation

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

From (3), solving the optimal control policy by minimizing the right-hand side of (3) yields the optimal value function $V^*(x_k)$. Based on the necessary condition for optimality, u_k^* can be obtained by taking the derivative of the right-hand side of (3) with respect to u_k. Thus, one has

$$u_k^* = -\frac{1}{2} R^{-1} g(x_k)^{\mathrm{T}} \frac{\partial V^*(x_{k+1})}{\partial x_{k+1}} \tag{4}$$

Substituting (4) into (3) yields DT HJB equation as

$$V^*(x_k) = x_k^{\mathrm{T}} Q x_k + \frac{1}{4} \frac{\partial V^{*\mathrm{T}}(x_{k+1})}{\partial x_{k+1}} g(x_k) R^{-1} \cdot g^{\mathrm{T}}(x_k) \frac{\partial V^*(x_{k+1})}{\partial x_{k+1}} + V^*(x_{k+1}) \tag{5}$$

Note that (5) is backward in time, and it is impossible to obtain x_{k+1} at the current time instant k. Especially for the affine nonlinear characteristics of (1), DT HJB equation (5) cannot be solved exactly. To overcome these challenging difficulties, various RL methods including heuristic dynamic programming (HDP), action-dependent HDP (Q-learning), dual heuristic dynamic programming (DHP), action-dependent DHP, globalized DHP have been reported for approximating the optimal solution of DT HJB equation instead of solving the analytical optimal solution [28-32]. The followings introduce the Q-learning algorithm to approximately solve DT HJB equation (5).

III. ON-POLICY Q-LEARNING FORMULATION

This section focuses on three aspects: (a) review the on-policy Q-learning algorithm for finding the approximation value of the optimal control policy; (b) present a novel proof of

convergence of on-policy Q-learning algorithm; (c) show the bias of solution to DT HJB equation (5) if probing noises are added into the systems for enriching data.

A. Derivation of Q-learning algorithm

Define the optimal action-dependent Q-function as

$$Q^*(x_k,u_k)=x_k^{\mathrm{T}}Qx_k+u_k^{\mathrm{T}}Ru_k+V^*(x_{k+1}) \tag{6}$$

Then, one has

$$\begin{aligned}V^*(x_k)&=\min_{u_k}Q^*(x_k,u_k)\\&=Q^*(x_k,u_k^*)\end{aligned} \tag{7}$$

Combining with (3) yields the Q-function based DT HJB equation

$$Q^*(x_k,u_k^*)=x_k^{\mathrm{T}}Qx_k+(u_k^*)^{\mathrm{T}}Ru_k^*+Q^*(x_{k+1},u_{k+1}^*) \tag{8}$$

and the optimal control policy

$$u_k^*=-\frac{1}{2}R^{-1}g(x_k)^{\mathrm{T}}\frac{\partial Q^*(x_{k+1},u_{k+1}^*)}{\partial x_{k+1}} \tag{9}$$

Referring to value iteration algorithms [8, 30, 33], Algorithm 1 is given to learn the optimal control policy.

Algorithm 1: On-policy Q-learning

1: Initialize the optimal Q-function $Q^0(\cdot)=0$, and set the iteration index $i=0$;

2: Calculate the initial control u_k^0 by

$$u_k^0=\arg\min_{u_k}(x_k^{\mathrm{T}}Qx_k+u_k^{\mathrm{T}}Ru_k+Q^0(\cdot)) \tag{10}$$

3: Update the iterative Q-function

$$\begin{aligned}Q^1(x_k,u_k^0)&=x_k^{\mathrm{T}}Qx_k+(u_k^0)^{\mathrm{T}}Ru_k^0+Q^0(\cdot)\\&=x_k^{\mathrm{T}}Qx_k+(u_k^0)^{\mathrm{T}}Ru_k^0\end{aligned} \tag{11}$$

4: Update the sequence of action policies

$$\begin{aligned}u_k^i&=\arg\min_{u_k}(x_k^{\mathrm{T}}Qx_k+u_k^{\mathrm{T}}Ru_k+Q^i(x_{k+1},u_{k+1}^{i-1}))\\&=\arg\min_{u_k}Q^{i+1}(x_k,u_k)\\&=-\frac{1}{2}R^{-1}g(x_k)^T\frac{\partial Q^i(x_{k+1},u_{k+1}^{i-1})}{\partial x_{k+1}}\end{aligned} \tag{12}$$

and a sequence of Q-functions

$$\begin{aligned}Q^{i+1}(x_k,u_k^i)&=x_k^{\mathrm{T}}Qx_k+(u_k^i)^{\mathrm{T}}Ru_k^i\\&\quad+Q^i(x_{k+1},u_{k+1}^{i-1}(x_{k+1}))\end{aligned} \tag{13}$$

with $x_{k+1}=f(x_k)+g(x_k)u_k^i$.

5: If $\left\|Q^{i+1}(x_k,u_k^i)-Q^i(x_k,u_k^{i-1})\right\|\le\varepsilon$ $(\varepsilon>0)$, stop and obtain the approximate optimal control policy u_k^i; Otherwise, set $i=i+1$ and go back to step 4.

B. Convergence analysis of the on-policy Q-learning algorithm

The following two lemmas are given to use for the proof of convergence of Algorithm 1.

Definition 1: [31, 33] A feedback control u_n defined on Ω_x is said to be admissible with respect to (2) if u_n is continuous on a compact set $\Omega_u\in R^m$, $u(0)=0$, u_n stabilizes system (1) on Ω_x, and $J(x_0)$ is finite $\forall x_0\in\Omega_x$.

Lemma 1: Suppose the sequence $\{Q^{i+1}\}$ to be defined as in (13). If system (1) is controllable and $Q^0(\cdot)=0$, then the following conclusions hold.

(a). Let μ^i be an arbitrary sequence of control policies, function W^{i+1} be defined as

$$\begin{aligned}W^{i+1}(x_k,\mu^i)&=x_k^{\mathrm{T}}Qx_k+(\mu^i)^{\mathrm{T}}R\mu^i\\&\quad+W^i(f(x_k)+g(x_k)\mu^i,\mu^{i-1}(f(x_k)+g(x_k)\mu^i))\end{aligned} \tag{14}$$

and $W^0(\cdot)=0$, then $Q^{i+1}(x_k,u_k^i)\le W^{i+1}(x_k,\mu^i)$ can be satisfied;

(b). There exists an upper bound $Y(x_k)$ such that $Q^{i+1}(x_k,u_k^i)\le W^{i+1}(x_k,\mu^i)\le Y(x_k)$, where W^{i+1} is obtained by letting μ^i be an admissible control policy;

(c). If (8) is solvable, then $Q^{i+1}(x_k,u_k^i)\le Q^*(x_k,u^*(x_k))\le Y(x_k)$.

Proof: (a): Notice that $Q^{i+1}(x_k,u_k^i)$ is the result of minimizing the right-hand side of (13) by using u_k^i obtained from (12), while $W^{i+1}(x_k,\mu^i)$ is achieved under an arbitrary control input referring to (14), then $Q^{i+1}(x_k,u_k^i)\le W^{i+1}(x_k,\mu^i)$ can be derived

(b): Let $\mu^i=\eta(x_k)$ to be an admissible control policy, and let $Q^0(\cdot)=W^0(\cdot)=0$, one has the following difference

$$\begin{aligned}&W^{i+1}(x_k,\eta(x_k))-W^i(x_k,\eta(x_k))\\&\quad=W^i(x_{k+1},\eta(x_{k+1}))-W^{i-1}(x_{k+1},\eta(x_{k+1}))\\&\quad=W^{i-1}(x_{k+2},\eta(x_{k+2}))-W^{i-2}(x_{k+2},\eta(x_{k+2}))\\&\quad\vdots\\&\quad=W^2(x_{k+i-1},\eta(x_{k+i-1}))-W^1(x_{k+i-1},\eta(x_{k+i-1}))\\&\quad=W^1(x_{k+i},\eta(x_{k+i}))-W^0(x_{k+i},\eta(x_{k+i}))\\&\quad=W^1(x_{k+i},\eta(x_{k+i}))\end{aligned} \tag{15}$$

Rewriting (15) yields

$$\begin{aligned}W^{i+1}(x_k,\eta(x_k))&=W^1(x_{k+i},\eta(x_{k+i}))-W^i(x_k,\eta(x_k))\\&=W^1(x_{k+i},\eta(x_{k+i}))+W^1(x_{k+i-1},\eta(x_{k+i-1}))\\&\quad+W^{i-1}(x_k,\eta(x_k))\\&=W^1(x_{k+i},\eta(x_{k+i}))+W^1(x_{k+i-1},\eta(x_{k+i-1}))\\&\quad+W^1(x_{k+i-2},\eta(x_{k+i-2}))+W^{i-2}(x_k,\eta(x_k))\\&\quad\vdots\\&=W^1(x_{k+i},\eta(x_{k+i}))+W^1(x_{k+i-1},\eta(x_{k+i-1}))\\&\quad+W^1(x_{k+i-2},\eta(x_{k+i-2}))+\cdots+W^1(x_k,\eta(x_k))\\&=\sum_{n=0}^{i}x_{k+n}^{\mathrm{T}}Qx_{k+n}+\eta^{\mathrm{T}}(x_{k+n})R\eta(x_{k+n})\end{aligned} \tag{16}$$

Since $\eta(x_k)$ is an admissible control policy, one further has

$$W^{i+1}(x_k,\eta(x_k))\le Y(k) \tag{17}$$

where $Y(k)=\sum_{n=0}^{\infty}x_{k+n}^{\mathrm{T}}Qx_{k+n}+\eta^{\mathrm{T}}(x_{k+n})R\eta(x_{k+n})$. Combining with (a), (b) holds, i.e., $Q^{i+1}(x_k,u_k^i)\le W^{i+1}(x_k,\mu^i)\le Y(x_k)$.

(c): If $\eta(x_k)=u^*(x_k)$, then $Q^{i+1}(x_k,u_k^i)\le Q^*(x_k,u^*(x_k))\le Y(x_k)$ can be derived from (b). This completes the proof. □

Lemma 2*:* Suppose the sequences u^i and Q^i to be defined as in (12) and (13). If $Q^0(\cdot)=0$, then it follows $Q^i\le Q^{i+1}$.

Proof: By (a) of Lemma 1, we have $Q^i\le W^i$. Next we shall show $W^i\le Q^{i+1}$ by using induction.

Since μ^i is an arbitrary sequence of control policies, then we let $\mu^i=u_k^{i+1}$. First, when $i=0$, one has

$$Q^1(x_k,u_k^0)-W^0(x_k,u_k^0)=x_k^{\mathrm{T}}Qx_k+(u_k^0)^{\mathrm{T}}Ru_k^0\ge 0 \tag{18}$$

which means $Q^1\ge W^0$.

Suppose that $W^{i-1}(x_k,u_k^{i-1})\le Q^i(x_k,u_k^{i-1})$ holds, then one has

$$\begin{aligned}&Q^{i+1}(x_k,u_k^i)-W^i(x_k,u_k^i)\\&=Q^i(f(x_k)+g(x_k)u_k^i,u^{i-1}(f(x_k)+g(x_k)u_k^i))\\&-W^{i-1}(f(x_k)+g(x_k)u_k^i,u^{i-1}(f(x_k)+g(x_k)u_k^i))\\&=Q^i(x_{k+1},u^{i-1}(x_{k+1}))-W^{i-1}(x_{k+1},u^{i-1}(x_{k+1}))\ge 0\end{aligned} \tag{19}$$

By induction, it can conclude $W^i\le Q^{i+1}$. Since $Q^i\le W^i$, then $Q^i\le Q^{i+1}$ holds. This completes the proof. □

Theorem 1*:* For the iterative control policy u^i and the iterative Q-function Q^i respectively defined as in (12) and (13), if $Q^0(\cdot)=0$, then Q^i converges to the optimal value Q^* and u^i converges to the optimal control policy u^* as $i\to\infty$, i.e., $\lim_{i\to\infty}Q^i=Q^*$ and $\lim_{i\to\infty}u^i=u^*$.

Proof*:* From Lemma 1 and Lemma 2, one can conclude that the iterative Q-function Q^i converges, which leads to u^i converging as well. We are now in a position to prove that they respectively converge to the optimal value Q^* and the optimal control policy u^* as $i\to\infty$.

By (7), one has

$$Q^*(x_k,u_k^*)\le Q^{i+1}(x_k,u_k^i) \tag{20}$$

Combining (20) with the conclusion (c) of Lemma 1 yields

$$\lim_{i\to\infty}Q^i(x_k,u_k^{i-1})=Q^*(x_k,u_k^*) \tag{21}$$

Thus, one has $\lim_{i\to\infty}u^i=u^*$ by referring to (9) and (12). This completes the proof. □

Remark 1*:* Solving $Q^{i+1}(x_k,u_k^i)$ in terms of (13) when implementing Algorithm 1 generally needs to add probing noise for satisfying PE condition like [8-10, 14-26]. [19] has shown that incorrect solutions resulting in incorrect optimal control policy would be caused by probing noise if using on-policy HDP method for optimal control of linear DT systems. This conclusion will be proven to still hold by the sequel for the case of affine nonlinear systems with using on-policy Q-learning algorithm.

C. Bias of solution analysis for on-policy Q-learning algorithm

Lemma 3: Suppose that probing noise e_k is added to the control policy u_k^i in Algorithm 1. Let $\tilde{Q}^{i+1}$ be the solution to (13) with $\tilde{u}_k^i=u_k^i+e_k$, $e_k\ne 0$, then $\tilde{Q}^{i+1}$ is not the solution to (13) with $e_k=0$.

Proof: Let (13) be Bellman equation without probing noise, i.e. $e_k=0$. If probing noise is added into the system (1), i.e. $\tilde{u}_k^i=u_k^i+e_k$ $(e_k\ne 0)$ acts as control input to generate data using for performance evaluation, then (1) and (13) respectively become the forms below

$$\tilde{x}_{k+1}=f(x_k)+g(x_k)u_k^i+g(x_k)e_k \tag{22}$$

and

$$\begin{aligned}\tilde{Q}^{i+1}(x_k,u_k^i)&=x_k^{\mathrm{T}}Qx_k+(u_k^i)^{\mathrm{T}}Ru_k^i\\&+\tilde{Q}^i(\tilde{x}_{k+1},u_{k+1}^i(\tilde{x}_{k+1}))\end{aligned} \tag{23}$$

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

$$\begin{aligned}\tilde{Q}^{i+1}(x_k,u_k^i)&=x_k^{\mathrm{T}}Qx_k+(u_k^i)^{\mathrm{T}}Ru_k^i\\&+\tilde{Q}^i(x_{k+1}+g(x_k)e_k,u_{k+1}^i(x_{k+1}+g(x_k)e_k))\end{aligned} \tag{24}$$

Contrasting (13) with (24) shows that $\tilde{Q}^{i+1}$ is not the same as Q^{i+1}, which might lead to incorrect the control update since

$$\begin{aligned}u_k^{i+1}&=-\frac{1}{2}R^{-1}g(x_k)^{\mathrm{T}}\\&\cdot\frac{\partial\tilde{Q}^{i+1}(x_{k+1}+g(x_k)e_k,u_{k+1}^i(x_{k+1}+g(x_k)e_k))}{\partial x_{k+1}}\end{aligned} \tag{25}$$

This completes the proof. □

IV. Off-policy Q-learning technique

The basic target of this paper is to present an off-policy Q-learning method for achieving optimum of control performance of affine nonlinear DT systems. This section devotes to proposing an off-policy Q-learning algorithm and proving the convergence of the proposed off-policy Q-learning algorithm, as well as analyzing no bias of solution even though probing noise is added into the systems for reaching PE condition.

A. Off-policy and Q-learning

On-policy and off-policy are two kinds of RL methods. On-policy methods evaluate or improve the same policy as the one that is applied to the systems for generating data. While, in the off-policy methods, there exist two types of unrelated control policies, one is called behavior policy used to generate date for implementing learning, and the other is target or estimation policy, which is evaluated and improved to approximate the optimal control policy [4, 14-26].

Q-learning can be implemented by on-policy [8-10, 14-18] or off-policy approach [19-21, 26] depending on updating Q-function value by using data from a behaviour policy or the target policy. What is showed Q-learning in Algorithm 1 is actually an on-policy method because it updates its Q-values using the trajectories drawn from the evaluated action.

B. Derivation of off-policy Q-learning algorithm

Introducing an auxiliary variable u_k^i into system (1) yields

$$x_{k+1}=f(x_k)+g(x_k)u_k^i+g(x_k)(u_k-u_k^i) \tag{26}$$

where u_k is called the behavior policy and u_k^{i-1} is viewed as the target policy needed to be evaluated and improved. It is well known that (12) and (13) are respectively equivalent to

$$u_k^i = -\frac{1}{2}R^{-1}g(x_k)^{\mathrm{T}}\frac{\partial Q^i(f(x_k)+g(x_k)u_k^{i-1},u_{k+1}^{i-1})}{\partial(f(x_k)+g(x_k)u_k^{i-1})} \tag{27}$$

and

$$\begin{aligned} Q^{i+1}(x_k,u_k^i) &= x_k^{\mathrm{T}}Qx_k+(u_k^i)^{\mathrm{T}}Ru_k^i \\ &+Q^i(f(x_k)+g(x_k)u_k^i,u_{k+1}^{i-1}(f(x_k)+g(x_k)u_k^i)) \end{aligned} \tag{28}$$

Along the trajectory of (26), (27) and (28) can be respectively rewritten as

$$u_k^i = -\frac{1}{2}R^{-1}g(x_k)^{\mathrm{T}}\frac{\partial Q^i\{x_{k+1}^{i-1},u^{i-1}(x_{k+1}^{i-1})\}}{\partial(x_{k+1}^{i-1})} \tag{29}$$

and

$$\begin{aligned} &Q^{i+1}(x_k,u_k^i)-Q^i\{x_{k+1}-g(x_k) \\ &\cdot(u_k-u_k^i),u^{i-1}(x_{k+1}-g(x_k)(u_k-u_k^i))\} \\ &=x_k^{\mathrm{T}}Qx_k+(u_k^i)^{\mathrm{T}}Ru_k^i \end{aligned} \tag{30}$$

where $x_{k+1}^{i-1}=x_{k+1}-g(x_k)(u_k-u_k^{i-1})$. The following Algorithm 2 is to show how to implement off-policy learning for approximating the optimal control policy.

Algorithm 2: Off-policy Q-learning

1: Initialize the optimal Q-function $Q^i(\cdot)=0$, and set the iteration index $i=0$;

2: Calculate the initial control u_k^0 by

$$u_k^0=\arg\min_{u_k}(x_k^{\mathrm{T}}Qx_k+u_k^{\mathrm{T}}Ru_k+Q^0(x_{k+1}^0,u_{k+1}^{-1}(x_{k+1}^0))) \tag{31}$$

3: Update the iterative Q-function

$$Q^1(x_k,u_k^0)=x_k^{\mathrm{T}}Qx_k+(u_k^0)^{\mathrm{T}}Ru_k^0 \tag{32}$$

4: Update the sequence of action policies by (30) and the sequence of the iterative Q-functions by (29);

5: If $\left\|Q^{i+1}(x_k,u_k^i)-Q^i(x_k,u_k^{i-1})\right\|\le\varepsilon$, stop and obtain the approximate optimal control policy u_k^i; Otherwise, set $i=i+1$ and go to step 4.

Theorem 2: (Q^{i+1},u^i) is the solution of (12) and (13) if and only if it is the solution of (29) and (30).

Proof: One can find that if (Q^{i+1},u^i) is the solution of (12) and (13), then it also make (27) and (28) hold for $\forall x_k\in\Omega_x$ (Ω_x is a compact set). For the state x_k generated by (26), substituting $x_{k+1}-g(x_k)(u_k-u_k^i)=f(x_k)+g(x_k)u_k^i$ into (27) and (28) yields (29) and (30), so the solution (Q^{i+1},u^i) of (12) and (13) can satisfy (29) and (30) as well. Next, we shall prove that the solution of (29) and (30) is also the solution of (12) and (13). Substituting (26) into (29) and (30) yields (27) and (28), further gets (13) and (12). This completes the proof. □

Remark 2: Note that the solutions of Algorithm 1 and Algorithm 2 are equivalent as shown in Theorem 2. Moreover, the convergence of Algorithm 1 has been proved in Theorem 1, therefore, if (Q^{i+1},u^i) can be solved correctly from Algorithm 2, then $\lim_{i\to\infty}Q^i=Q^*$ and $\lim_{i\to\infty}u^i=u^*$ can be concluded.

Remark 3: The Q-learning in Algorithm 2 is definitely an off-policy approach, since the target control policy is updated but not to be applied to the real systems during learning due to the introduction of an arbitrary stabilizing behavior policy u_k used to generate data and enrich data exploration, which is a remarkable feature possessed by the off-policy learning as opposed to the on-policy learning [4, 14-26].

C. No bias of off-policy Q-learning algorithm

In [19], it was shown that adding probing noise does not result in biased solution for optimal control of linear DT systems using off-policy RL learning. Here, we extend that result to affine nonlinear DT systems for finding the optimal control policy by using off-policy Q-learning.

Theorem 3: Suppose that a probing noise e_k is added to the behavior policy in Algorithm 2. Let $(\bar{Q}^{i+1},\bar{u}^i)$ be the solution to (29) and (30) with $\bar{u}_k=u_k+e_k$, $e_k\neq 0$, then $(\bar{Q}^{i+1},\bar{u}^i)$ is also the solution to (29) and (30) with $e_k=0$.

Proof: A probing noise is added into the behavior control policy, that is, $\bar{u}_k=u_k+e_k$. By Algorithm 2, $\bar{u}_k^0=u_k^0$ and $\bar{Q}^1(x_k,\bar{u}_k^0)=Q^1(x_k,u_k^0)$ hold. We assume $\bar{u}_k^{i-1}=u_k^{i-1}$ and $\bar{Q}^i(x_k,\bar{u}_k^{i-1})=Q^i(x_k,u_k^{i-1})$ hold, and substituting $\bar{u}_k$ into (26) yields

$$\begin{aligned} \bar{x}_{k+1} &= f(x_k)+g(x_k)(u_k+e_k) \\ \bar{x}_{k+1}^{i-1} &= \bar{x}_{k+1}-g(x_k)(\bar{u}_k-\bar{u}_k^{i-1}) \\ &= f(x_k)+g(x_k)u_k^{i-1} \\ &= x_{k+1}^{i-1} \end{aligned} \tag{33}$$

By (29), one has

$$\bar{u}_k^i = -\frac{1}{2}R^{-1}g(x_k)^{\mathrm{T}}\frac{\partial\bar{Q}^i\{\bar{x}_{k+1}^{i-1},\bar{u}^{i-1}(\bar{x}_{k+1}^{i-1})\}}{\partial(\bar{x}_{k+1}^i)} \tag{34}$$

Due to $\bar{u}_k^{i-1}=u_k^{i-1}$ and $\bar{Q}^i(x_k,\bar{u}_k^{i-1})=Q^i(x_k,u_k^{i-1})$, (34) becomes

$$\bar{u}_k^i = -\frac{1}{2}R^{-1}g(x_k)^{\mathrm{T}}\frac{\partial Q^i\{x_{k+1}^{i-1},u^{i-1}(x_{k+1}^{i-1})\}}{\partial(x_{k+1}^{i-1})} \tag{35}$$

By comparing (29) and (35), one can conclude that $\bar{u}_k^i=u_k^i$ resulting in $\bar{x}_{k+1}^i=x_{k+1}^i$ by referring to (33). Note that

$$\begin{aligned} \bar{Q}^{i+1}(x_k,\bar{u}_k^i) &= x_k^{\mathrm{T}}Qx_k+(\bar{u}_k^i)^{\mathrm{T}}R\bar{u}_k^i \\ &+\bar{Q}^i(\bar{x}_{k+1}^i,\bar{u}^{i-1}(\bar{x}_{k+1}^i)) \end{aligned} \tag{36}$$

substituting $\bar{u}_k^i=u_k^i$ and $\bar{x}_{k+1}^i=x_{k+1}^i$ into (36), one has

$$\begin{aligned} \bar{Q}^{i+1}(x_k,u_k^i) &= x_k^{\mathrm{T}}Qx_k+(u_k^i)^{\mathrm{T}}Ru_k^i \\ &+\bar{Q}^i\{x_{k+1}-g(x_k)(u_k-u_k^i),u^i(x_{k+1}-g(x_k)(u_k-u_k^i))\} \end{aligned} \tag{37}$$

By comparing (30) with (37), one can conclude that $\bar{Q}^{i+1}=Q^{i+1}$. By mathematical induction, $(\bar{Q}^{i+1},\bar{u}^i)=(Q^{i+1},u^i)$ even though $e_k\neq 0$.

Therefore, adding the probing noise during implementing the proposed off-policy Q-learning Algorithm 2 cannot produce bias of solution. This completes the proof. □

Remark 4: In contrast to the off-policy Q-learning method [26], the developed off-policy Q-learning Algorithm 2 in this paper has a different learning strategy shown in (29) and (30). More importantly, even though probing noise is added to the system for satisfying PE condition, no bias of solution can be guaranteed and is proved for the first time from the perspective of Q-learning, whereas off-policy RL for linear DT systems was taken into account in [19-21].

V. NEURAL NETWORK-BASED OFF-POLICY INTERLEAVED Q-LEARNING

In this section, three neural networks are used to approximate $Q^i(x_k, u_k^{i-1})$, u_k^i and the affine nonlinear system (1) by function value approximation approach. Algorithm 2 is implemented based on interleaved-learning critic and actor structure by neural networks. Therefore, this is a data-driven approximate optimal control strategy without the knowledge of system model.

A. The model neural network

Note that updating Q-function and control policy in Algorithm 2 requires $g(x_k)$ to be known a priori, but it is difficult to know $g(x_k)$ in real applications. Actually $g(x_k) = \dfrac{\partial x_{k+1}}{\partial u_k}$, so the following three-lay NN [29, 33] is used to approximate the dynamics of system (1) for estimating $g(x_k)$ by using $\dfrac{\partial \hat{x}_{k+1}}{\partial u_k}$.

$$\hat{x}_{k+1} = \omega_x^T \sigma(v_x^T \begin{bmatrix} x_k \\ u_k \end{bmatrix}) \tag{38}$$

where ω_x and v_x are respectively the weights of the hidden layer to the output layer and the weights of the input layer to the hidden layer. $\sigma(v_x^T \begin{bmatrix} x_k \\ u_k \end{bmatrix}) \in R^l$ is the activation function vector, $[\sigma(z)]_l = (e^z - e^{-z})/(e^z + e^{-z})$ and l is the number of neurons in the hidden layer. To train the NN (38), the gradient descent algorithm is used to update the weight ω_x.

$$\omega_x(k+1) = \omega_x(k) - \eta_c \frac{\partial E_{xk}}{\partial \omega_x} \tag{39}$$

where e_{xk} and E_{xk} are respectively the approximate error and the squared error of model network, and they are defined as

$$E_{xk} = \frac{1}{2} e_{xk}^{\mathrm{T}} e_{xk},\ e_{xk} = \hat{x}_{k+1} - x_{k+1} \tag{40}$$

Thus, one has

$$\begin{aligned} \frac{\partial E_{xk}}{\partial \omega_x} &= \frac{\partial E_{xk}}{\partial e_{xk}} \frac{\partial e_{xk}}{\partial \hat{x}_{k+1}} \frac{\partial \hat{x}_{k+1}}{\partial \omega_x} \\ &= \sigma(v_x^{\mathrm{T}} \begin{bmatrix} x_k \\ u_k \end{bmatrix}) e_{xk}^{\mathrm{T}} \end{aligned} \tag{41}$$

B. The actor neural network

We employ the actor NN to approximate actor u_k^i given by

$$\hat{u}_k^i = (\hat{\omega}_{ak}^i)^{\mathrm{T}} \sigma(Z_a(k)) \tag{42}$$

where $Z_a(k) = v_a^{\mathrm{T}} x_k$, $\hat{\omega}_{ak}^i$ and v_a respectively are the weights of the hidden layer to the output layer and the weights of the input layer to the hidden layer. Training $\hat{\omega}_{ak}^i$ is implemented by using gradient descent algorithm.

$$\hat{\omega}_{ak}^{i+1} = \hat{\omega}_{ak}^i - \eta_a \frac{\partial E_{ak}^i}{\partial \hat{\omega}_{ak}^i} \tag{43}$$

where e_{ak}^i and E_{ak}^i respectively are the approximate error and the squared error of actor network, and they are defined as

$$E_{ak}^i = \frac{1}{2}(e_{ak}^i)^{\mathrm{T}} e_{ak}^i,\ e_{ak}^i = \hat{u}_k^i - u_k^i \tag{44}$$

and

$$\begin{aligned} \frac{\partial E_{ak}^i}{\partial \hat{\omega}_{ak}^i} &= \frac{\partial E_{ak}^i}{e_{ak}^i} \frac{e_{ak}^i}{\partial \hat{u}_k^i} \frac{\partial \hat{u}_k^i}{\partial \hat{\omega}_{ak}^i} \\ &= e_{ak}^i \sigma(Z_a(k)) \end{aligned}$$

$$u_k^i = -\frac{1}{2} R^{-1} (\frac{\partial \hat{x}_{k+1}}{\partial u_k})^{\mathrm{T}} \frac{\partial \hat{Q}^i \{x_{k+1}^{i-1}, u^{i-1}(x_{k+1}^{i-1})\}}{\partial (x_{k+1}^{i-1})} \tag{45}$$

C. The critic neural network

A critic neural network is used to allow approximate the iterative Q-function. The critic NN is given by the form of the three-layer neural network

$$\hat{Q}^{i+1}(x_k, \hat{u}_k^i) = (\hat{\omega}_{c,k}^{i+1})^{\mathrm{T}} \sigma(\hat{Z}_c(k)) \tag{46}$$

where $\hat{Z}_c^i(k) = v_c^{\mathrm{T}} \begin{bmatrix} x_k \\ \hat{u}_k^i \end{bmatrix}$, ω_{ci} and v_c respectively are the weights of the hidden layer to the output layer and the weights of the input layer to the hidden layer. (46) can be used to approximate $\hat{Q}^i(x_{k+1}, \hat{u}_{k+1}^{i-1})$ as

$$\hat{Q}^i \{x_{k+1}^i, \hat{u}^{i-1}(x_{k+1}^i)\} = (\hat{\omega}_{c,k}^i)^{\mathrm{T}} \sigma(v_c^{\mathrm{T}} \begin{bmatrix} x_{k+1}^i \\ \hat{u}^i(x_{k+1}^i) \end{bmatrix}) \tag{47}$$

where

$$\begin{aligned} &x_{k+1}^i = x_{k+1} - \frac{\partial \hat{x}_{k+1}}{\partial u_k}(u_k - \hat{u}_k^i), \\ &\frac{\partial \hat{x}_{k+1}}{\partial u_k} = \omega_x I_1 (\bar{I} - \sigma^2(v_x^{\mathrm{T}} \begin{bmatrix} x_k \\ u_k \end{bmatrix}), \\ &I_1 = \begin{bmatrix} 0_{n\times n} & 0_{n\times m} \\ 0_{m\times n} & I_{m\times m} \end{bmatrix},\ \bar{I} = \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix}_{(n+m)\times 1} \end{aligned} \tag{48}$$

Then, the squared error and approximate error of the critic network are respectively defined as E_{ck}^i and e_{ck}^i.

$$E_{ck}^i = \frac{1}{2}(e_{c,k}^i)^{\mathrm{T}} e_{ck}^i \tag{49}$$

$$\begin{aligned} e_{ck}^i &= \hat{Q}^i(x_k, \hat{u}_k^{i-1}) - Q^i(x_k, \hat{u}_k^{i-1}) \\ &= \hat{Q}^i(x_k, \hat{u}_k^{i-1}) - (x_k^{\mathrm{T}} Q x_k + (\hat{u}_k^{i-1})^{\mathrm{T}} R \hat{u}_k^{i-1} \\ &+ \hat{Q}^{i-1}\{x_{k+1}^{i-1}, \hat{u}^{i-2}(x_{k+1}^{i-1})\}) \end{aligned} \tag{50}$$

The gradient descent algorithm is used to update the weight for the critic network, which is given as follows

$$\hat{\omega}_{ck}^{i+1} = \hat{\omega}_{ck}^{i} - \eta_c \frac{\partial E_{ck}^{i}}{\partial \hat{\omega}_{ck}^{i}} \tag{51}$$

where

$$\begin{aligned} \frac{\partial E_{c,k}^{i}}{\partial \hat{\omega}_{c,k}^{i}} &= \frac{\partial E_{c,k}^{i}}{\partial e_{c,k}^{i}} \frac{\partial e_{c,k}^{i}}{\partial \hat{Q}^{i}(x_k, \hat{u}_k^{i})} \frac{\partial \hat{Q}^{i}(x_k, \hat{u}_k^{i-1})}{\partial \hat{\omega}_{c,k}^{i}} \\ &= e_{c,k}^{i} \sigma(\hat{Z}_c^{i-1}(k)) \end{aligned} \tag{52}$$

D. Interleaved Q-learning

The following presents an off-policy interleaved Q-learning algorithm based on interleaved-iteration critic and actor networks.

Algorithm 3 Off-policy interleaved Q-learning

1: Data collection: Collect system data x_k and store them in the sample sets by using the behavior control policy u_k;

2: Initialization: Initialize weight vector ω_x, v_x for the affine nonlinear neural network;

3: Train the model network

(1) Train weight in terms of (39) using the measured data until $e_{xk} \le \varepsilon_x$ $(\varepsilon_x > 0)$;

(2) Get the trained weight ω_x. Let $k = 0$;

4: Interleaved iteration

(1) Let the initial iterative Q-function $Q^0(\cdot) = 0$ and further calculate the initial control u_k^0 by (31). Let the iteration index $i = 0$;

(2) Train critic network: Calculate $\hat{Q}^{i+1}(x_k, \hat{u}_k^{i})$, $\hat{Q}^{i}\{x_{k+1}^{i}, \hat{u}^{i}(x_{k+1}^{i})\}$ and e_{ck}^{i} using (46), (47) and (50) to update the critic weight $\hat{\omega}_{ck}^{i+1}$ once using (51);

(3) Train actor network: Calculate $\hat{u}_k^{i}$, u_k^{i} and e_{ak}^{i} using (42) and (45) to update the actor weight $\hat{\omega}_{ak}^{i+1}$ once using (43);

(4) Check $\left\| \hat{Q}^{i}(x_k, \hat{u}_k^{i-1}) - \hat{Q}^{i+1}(x_k, \hat{u}_k^{i}) \right\| \le \varepsilon_Q$ $(\varepsilon_Q > 0)$, if it is not satisfied, go to (2) in step (4), otherwise get $\hat{u}_k^{i}$.

5: Set $k = k + 1$, and go back to step 4.

Remark 5: One can easily find that no information on dynamics of affine nonlinear systems is required when learning the optimal control policy by constructing three neural networks in Algorithm 3.

Remark 6*:* In Algorithm 1 and Algorithm 2 of this paper, the neural network weight $\hat{\omega}_{ck}^{i+1}$ is kept training with time k until it converges for each iterative index $i+1$ and then the actor weight $\hat{\omega}_{ak}^{i}$ is trained by the same approach, which is the traditional value iteration RL method [8, 30, 33]. While, in interleaved Q-learning Algorithm 3, for each time k, critic network and actor network are interleaved iterated with iterative index i until convergence, and they finally converge with increasing time k. Actually, the proposed interleaved Q-learning is a kind of variant of generalized value iteration [34, 35], is more easily implemented for the practical applications, which is another bright spot in this paper.

Remark 7: Notice that in Algorithm 3 the critic neural network for Q-function value is updated off-line by using an entire set of data under the PE condition, instead of on-line updating it. This idea is basically the same as Neural Fitted Q (NFQ) Iteration [2].

Theorem 4*:* Let the optimal performance index function and the optimal control policy be expressed by

$$Q^*(x_k, u_k^*) = (\omega_{ck}^*)^{\mathrm{T}} \sigma(Z_c(k)) \tag{53}$$

and

$$u_k^* = (\omega_{ak}^*)^{\mathrm{T}} \sigma(Z_a(k)) \tag{54}$$

respectively, where $Z_c(k) = v_c^{\mathrm{T}} \begin{bmatrix} x_k \\ u_k^* \end{bmatrix}$. Let the actor and critic networks be regulated by (43) and (51), respectively. Let $\bar{\omega}_{ck}^{i} = \hat{\omega}_{ck}^{i} - \omega_{ck}^{*}$, $\bar{\omega}_{ak}^{i} = \hat{\omega}_{ak}^{i} - \omega_{ak}^{*}$, if there exist $W_c > 0$ and $W_a > 0$ satisfying

$$\begin{aligned} &\eta_c < \frac{1}{\left\| \sigma(\hat{Z}_c^{i}(k)) \right\|^2}, \ \eta_a < \frac{1}{\left\| \sigma(Z_a(k)) \right\|^2}, \\ &\left\| e_{ck}^{i} \right\|^2 > \frac{W_c \left\| \sigma(\hat{Z}_c^{i}(k)) \right\|^2}{\eta_q}, \\ &\left\| e_{ak}^{i} \right\|^2 > \frac{W_a \left\| \sigma(Z_a(k)) \right\|^2}{\eta_u} \end{aligned} \tag{55}$$

then the errors $\bar{\omega}_{ck}^{i}$ and $\bar{\omega}_{ak}^{i}$ both converge to zero, as $i \to \infty$.

where $\eta_q = 2 - \eta_c \left\| \sigma(\hat{Z}_c(k+1)) \right\|^2 - \left\| \sigma(\hat{Z}_c(k+1)) \right\|^2$, $\eta_u = 2 - \eta_a \left\| \sigma(Z_a(k)) \right\|^2 - \left\| \sigma(Z_a(k)) \right\|^2$.

Proof: By (43) and (51), one has

$$\bar{\omega}_{ck}^{i+1} = \bar{\omega}_{ck}^{i} - \eta_c e_{ck}^{i} \sigma(\hat{Z}_c^{i-1}(k)) \tag{56}$$

and

$$\bar{\omega}_{ak}^{i+1} = \bar{\omega}_{ak}^{i} - \eta_a e_{ak}^{i} \sigma(Z_a(k)) \tag{57}$$

Choose a Lyapunov function candidate as

$$V(\bar{\omega}_{c,k}^{i}, \bar{\omega}_{a,k}^{i}) = \mathrm{tr}((\bar{\omega}_{c,k}^{i})^{\mathrm{T}} \bar{\omega}_{c,k}^{i} + (\bar{\omega}_{a,k}^{i})^{\mathrm{T}} \bar{\omega}_{a,k}^{i}) \tag{58}$$

Let $V_1(i) = (\bar{\omega}_{ck}^{i})^{\mathrm{T}} \bar{\omega}_{ck}^{i}$ and $V_2(i) = (\bar{\omega}_{a,k}^{i})^{\mathrm{T}} \bar{\omega}_{a,k}^{i}$, so the following holds

$$\begin{aligned} \Delta V_1(i) &= \mathrm{tr}((\bar{\omega}_{ck}^{i+1})^{\mathrm{T}} \bar{\omega}_{ck}^{i+1} - (\bar{\omega}_{ck}^{i})^{\mathrm{T}} \bar{\omega}_{ck}^{i}) \\ &= \eta_c^2 (e_{ck}^{i} \sigma(\hat{Z}_c(k+1)))^{\mathrm{T}} e_{ck}^{i} \sigma(\hat{Z}_c(k+1)) \\ &\quad - 2\eta_c (e_{ck}^{i} \sigma(\hat{Z}_c^{i-1}(k)))^{\mathrm{T}} \bar{\omega}_{ck}^{i}) \\ &= \eta_c^2 \left\| e_{ck}^{i} \right\|^2 \left\| \sigma(\hat{Z}_c^{i-1}(k)) \right\|^2 \\ &\quad - 2\eta_c (e_{ck}^{i} \sigma(\hat{Z}_c^{i-1}(k)))^{\mathrm{T}} (\hat{\omega}_{ck}^{i} - \omega_{ck}^{i} + \omega_{ck}^{i} - \omega_{ck}^{*})) \end{aligned} \tag{59}$$

and

$$\begin{aligned} \Delta V_2(i) &= \mathrm{tr}((\bar{\omega}_{ak}^{i+1})^{\mathrm{T}} \bar{\omega}_{ak}^{i+1} - (\bar{\omega}_{ak}^{i})^{\mathrm{T}} \bar{\omega}_{ak}^{i}) \\ &= \eta_a^2 (e_{ak}^{i} \sigma(Z_a(k))^{\mathrm{T}} e_{ak}^{i} \sigma(Z_a(k) \\ &\quad - 2\eta_a (e_{ak}^{i} \sigma(Z_a(k))^{\mathrm{T}} \bar{\omega}_{ak}^{i}) \\ &= \eta_a^2 \left\| e_{ak}^{i} \right\|^2 \left\| \sigma(Z_a(k)) \right\|^2 \\ &\quad - 2\eta_a e_{ak}^{i} \sigma(Z_a(k))^{\mathrm{T}} (\hat{\omega}_{ak}^{i} - \omega_{ak}^{i} + \omega_{ak}^{i} - \omega_{ak}^{*})) \end{aligned} \tag{60}$$

We assume $Q^{i}\{x_{k+1}^{i}, \hat{u}^{i-1}(x_{k+1}^{i})\} = (\omega_{ck}^{i})^{\mathrm{T}} \sigma(v_c^{\mathrm{T}} \begin{bmatrix} x_{k+1}^{i} \\ \hat{u}^{i-1}(x_{k+1}^{i}) \end{bmatrix})$, and

$u_k^i = (\omega_{ak}^i)^{\mathrm{T}} Z_a(k)$, then we have $e_{ck}^i = \hat{Q}^i(x_k, \hat{u}_k^{i-1}) - Q^i(x_k, \hat{u}_k^{i-1}) = \sigma(\hat{Z}_c^{i-1}(k))^{\mathrm{T}}(\hat{\omega}_{ck}^i - \omega_{ck}^i)$ and $e_{ak}^i = \hat{u}_k^i - u_k^i = \sigma(Z_a(k))^{\mathrm{T}}(\hat{\omega}_{ak}^i - \omega_{ak}^i)$. By the analysis in Remark 2, one can know that $\lim_{i\to\infty}(\omega_{ck}^i - \omega_{ck}^*) = 0$ and $\lim_{i\to\infty}(\omega_{ak}^i - \omega_{ak}^*) = 0$. So there must exist $W_c > 0$ and $W_a > 0$, such that $\left\|\omega_{ck}^i - \omega_{ck}^*\right\|^2 \le W_c$, $\left\|\omega_{ak}^i - \omega_{ak}^*\right\|^2 \le W_a$ hold. Thus, one has

$$\begin{aligned}\Delta V_1(i) &= -\eta_c \left\|e_{ck}^i\right\|^2 (2 - \eta_c \left\|\sigma(\hat{Z}_c^{i-1}(k))\right\|^2) \\ &\quad + 2(e_{ck}^i \sigma^{\mathrm{T}}(\hat{Z}_c^{i-1}(k))(\omega_{ck}^* - \omega_{ck}^i)) \\ &\le -\eta_c \left\|e_{ck}^i\right\|^2 (2 - \eta_c \left\|\sigma(\hat{Z}_c^{i-1}(k))\right\|^2) \\ &\quad + \eta_c (\left\|e_{ck}^i\right\|^2 + \left\|\sigma^{\mathrm{T}}(\hat{Z}_c^{i-1}(k))(\omega_{ck}^* - \omega_{ck}^i)\right\|^2) \\ &= \eta_c (\left\|e_{ck}^i\right\|^2 \eta_q - W_c \left\|\sigma(\hat{Z}_c^{i-1}(k))\right\|^2)\end{aligned} \tag{61}$$

and

$$\begin{aligned}\Delta V_2(i) &= -\eta_a \left\|e_{ak}^i\right\|^2 (2 - \eta_a \left\|\sigma(Z_a(k))\right\|^2) \\ &\quad + 2(e_{ak}^i \sigma^{\mathrm{T}}(Z_a(k))(\omega_{ak}^* - \omega_{ak}^i)) \\ &\le -\eta_a \left\|e_{ak}^i\right\|^2 (2 - \eta_a \left\|\sigma(Z_a(k))\right\|^2) \\ &\quad + \eta_a (\left\|e_{ak}^i\right\|^2 + \left\|\sigma^T(Z_a(k))(\omega_{ak}^* - \omega_{ak}^i)\right\|^2) \\ &\le -\eta_a (\left\|e_{ak}^i\right\|^2 \eta_u - W_a \left\|\sigma(Z_a(k))\right\|^2)\end{aligned} \tag{62}$$

If (55) holds, then $\Delta V(i) = \Delta V_1(i) + \Delta V_2(i) < 0$. Hence, $\lim_{i\to\infty} \bar{\omega}_{ck}^i = 0$ and $\lim_{i\to\infty} \bar{\omega}_{ak}^i = 0$. This completes the proof. □

Remark 8: Since the analytical solution (Q^{i+1}, u^i) is quite hard to achieve, neural network approximation is necessary for presenting a numerical solution of them. But it has to point out that the reconstruction errors inherently exist due to the facts of $Q^*(x_k, u_k^*) = (\omega_{ck}^*)^{\mathrm{T}} \sigma(Z_c(k)) + \varepsilon_1(x_k)$ and $u_k^* = (\omega_{ak}^*)^{\mathrm{T}} \sigma(Z_a(k)) + \varepsilon_2(x_k)$, where $\varepsilon_1(x_k)$ and $\varepsilon_2(x_k)$ are bounded reconstruction errors. This means that $\hat{\omega}_{ck}^i - \omega_{ck}^*$ and $\hat{\omega}_{ak}^i - \omega_{ak}^*$ are both bounded, whose details can be seen in [36]. Hence, we claim that an approximate optimal solution of the HJB equation (8) is actually obtained instead of the exact optimal one.

D. For linear system using off-policy Q-learning

For linear DT system given as

$$x_{k+1} = A x_k + B u_k \tag{63}$$

Actually, the optimal Q-function is a quadratic form

$$Q^*(x_k, u_k) = (\begin{bmatrix} x_k \\ u_k \end{bmatrix}^{\mathrm{T}} \otimes \begin{bmatrix} x_k \\ u_k \end{bmatrix}^{\mathrm{T}}) \mathrm{vec}(H) \tag{64}$$

since $J(x_k) = (x_k^{\mathrm{T}} \otimes x_k^{\mathrm{T}}) \mathrm{vec}(P)$ if $u_k = -K x_k$, where $H \ge 0$, $P \ge 0$ and

$$H = \begin{bmatrix} H_{xx} & H_{xu} \\ (H_{xu})^{\mathrm{T}} & H_{uu} \end{bmatrix} = \begin{bmatrix} A^{\mathrm{T}} P A + Q & A^{\mathrm{T}} P B \\ B^{\mathrm{T}} P A & B^{\mathrm{T}} P B + R \end{bmatrix} \tag{65}$$

$$P = \begin{bmatrix} I \\ -K \end{bmatrix}^{\mathrm{T}} H \begin{bmatrix} I \\ -K \end{bmatrix} \tag{66}$$

DT HJB equation (8) is reduced as

$$(z_k^{\mathrm{T}} \otimes z_k^{\mathrm{T}}) \mathrm{vec}(H) = x_k^{\mathrm{T}} Q x_k + u_k^{\mathrm{T}} R u_k + (z_{k+1}^{\mathrm{T}} \otimes z_{k+1}^{\mathrm{T}}) \mathrm{vec}(H) \tag{67}$$

where $z_k = [x_k^{\mathrm{T}} \; u_k^{\mathrm{T}}]^{\mathrm{T}}$. (9) correspondingly becomes

$$u_k^* = -H_{uu}^{-1} (H_{xu})^{\mathrm{T}} x_k \tag{68}$$

Thus, (13) and (12) are correspondingly rewritten as

$$(z_k^{\mathrm{T}} \otimes z_k^{\mathrm{T}}) \mathrm{vec}(H^{i+1}) = x_k^{\mathrm{T}} Q x_k + (u_k^i)^{\mathrm{T}} R u_k^i + (z_{k+1}^{\mathrm{T}} \otimes z_{k+1}^{\mathrm{T}}) \mathrm{vec}(H^i) \tag{69}$$

and

$$u_k^i = -(H_{uu}^i)^{-1} (H_{xu}^i)^{\mathrm{T}} x_k \tag{70}$$

To implement the off-policy Q-learning algorithm for linear system (63), (26) is correspondingly changed into

$$x_{k+1} = A_c x_k + B(u_k - u_k^i) \tag{71}$$

where $A_c = A - BK^i$ and $u_k^j = -K^i x_k$. Notice that (69) is equivalent to the following form

$$\begin{aligned}\begin{bmatrix} I \\ -K^i \end{bmatrix}^{\mathrm{T}} H^{i+1} \begin{bmatrix} I \\ -K^i \end{bmatrix} &= Q + (K^i)^{\mathrm{T}} R K^i \\ &\quad + (A - BK^i)^{\mathrm{T}} \begin{bmatrix} I \\ -K^i \end{bmatrix}^{\mathrm{T}} H^i \begin{bmatrix} I \\ -K^i \end{bmatrix} (A - BK^i)\end{aligned} \tag{72}$$

Thus, (29) is correspondingly changed into

$$\begin{aligned}&Q^{i+1}(x_k, u_k^i) \\ &\quad - x_k^{\mathrm{T}} A_c^{\mathrm{T}} \begin{bmatrix} I \\ -K^j \end{bmatrix}^{\mathrm{T}} H^i \begin{bmatrix} I \\ -K^j \end{bmatrix} A_c x_k \\ &\quad = x_k^{\mathrm{T}} \begin{bmatrix} I \\ -K^i \end{bmatrix}^{\mathrm{T}} H^{i+1} \begin{bmatrix} I \\ -K^i \end{bmatrix} x_k - (x_{k+1} \\ &\quad - B(u_k - u_k^i))^{\mathrm{T}} \begin{bmatrix} I \\ -K^i \end{bmatrix}^{\mathrm{T}} H^i \begin{bmatrix} I \\ -K^i \end{bmatrix} \\ &\quad \cdot (x_{k+1} - B(u_k - u_k^i)) \\ &\quad = x_k^{\mathrm{T}} (Q + (K^i)^{\mathrm{T}} R K^i) x_k\end{aligned} \tag{73}$$

Since P^{i+1} and H^{i+1} have the relationship shown in (65) and (66), then the following off-policy Q-function based Bellman equation holds

$$\begin{aligned}&x_k^{\mathrm{T}} \begin{bmatrix} I \\ -K^i \end{bmatrix}^{\mathrm{T}} H^{i+1} \begin{bmatrix} I \\ -K^i \end{bmatrix} x_k \\ &\quad - x_{k+1}^{\mathrm{T}} \begin{bmatrix} I \\ -K^i \end{bmatrix}^{\mathrm{T}} H^i \begin{bmatrix} I \\ -K^i \end{bmatrix} x_{k+1} \\ &\quad + 2 x_k^{\mathrm{T}} H_{xu}^i (u_k + K^i x_k) + u_k^{\mathrm{T}} (H_{uu}^i - R)(u_k + K^i x_k) \\ &\quad - (K^i x_k)^{\mathrm{T}} (H_{uu}^i - R)(u_k + K^i x_k) \\ &\quad = x_k^{\mathrm{T}} (Q + (K^i)^{\mathrm{T}} R K^i) x_k\end{aligned} \tag{74}$$

Properly manipulating (74) yields the following form

$$\theta^i(k) \mathrm{vec}(H^{i+1}) = \rho_k^i \tag{75}$$

where

$$\begin{aligned}\rho_k^i &= x_k^{\mathrm{T}} Q x_k + u_k^{\mathrm{T}} R u_k \\ &\quad + \left(x_{k+1}^{\mathrm{T}} \begin{bmatrix} I \\ -K^i \end{bmatrix}^{\mathrm{T}} \right) \otimes \left(x_{k+1}^{\mathrm{T}} \begin{bmatrix} I \\ -K^i \end{bmatrix}^{\mathrm{T}} \right) vec(H^i)\end{aligned}$$

Table 1 Comparisons between on-policy and off-policy learning

	on-policy Q-learning			off-policy interleaved Q-learning		
Probing noise	Controller gain	mean	variation	Controller gain	mean	variation
1	N	N	N	$K^{10}=[0.3441 \quad -1.0017]$	$6.7075e^{-4}$	$1.7353e^{-5}$
2	$K^{1}=[0.4839 \quad -1.0025]$	N	N	$K^{10}=[0.3445 \quad -1.0016]$	$3.2443e^{-4}$	$1.7353e^{-5}$
3	$K^{3}=[0.1669 \quad -0.7327]$	N	N	$K^{10}=[0.3445 \quad -1.0016]$	$3.2849e^{-4}$	$1.7353e^{-5}$

$$-2(x_k^{\mathrm{T}} \otimes (u_k + K^i x_k)^{\mathrm{T}}) vec(H_{xu}^i)$$
$$-(u_k^{\mathrm{T}} \otimes u_k^{\mathrm{T}}) vec(H_{uu}^i)$$
$$\theta^i(k) = \left(x_k^{\mathrm{T}} \begin{bmatrix} I \\ -K^i \end{bmatrix}^{\mathrm{T}} \right) \otimes \left(x_k^{\mathrm{T}} \begin{bmatrix} I \\ -K^i \end{bmatrix}^{\mathrm{T}} \right)$$

When finding H^{i+1}, thus K^{j+1} can be calculated as

$$K^{i+1} = (H_{uu}^{i+1})^{-1} (H_{xu}^{i+1})^{\mathrm{T}} \tag{76}$$

Algorithm 3 is reduced as follows for the case of linear system.

Algorithm 4 Off-policy interleaved Q-learning for linear systems

1: Data collection: Collect system data x_k and store them in the sample sets $\theta^i(k)$ and ρ^i by using the behavior control policy u_k;

2: Initiation: Choose the initial stabilizing gains K^0, and let the initial iterative matrix H^0. Set $i=0$, $k=0$;

3: Implementing Q-learning: Calculate H^{i+1} in (75) using the collected data in Step 1, and then K^{i+1} can be updated in terms of (76);

4: If $\left\| K^{i+1} - K^i \right\| \le l$ (l is some small positive number), then stop the iteration, and let $k = k+1$, go back to Step 3, and thus the optimal control policy has been obtained. Otherwise, let $i = i+1$ and go back to Step 3.

Remark 9: A distinctive feature existed in Algorithm 4 for the special case of linear systems is to approximate the optimal control policy gain without knowing system matrices A and B, even no need of identifying system model using neural networks or something similar.

VI. Simulation Results

In this section, the proposed off-policy interleaved Q-learning algorithm is applied to two representative examples to show its effectiveness. Simulations are operated to show the no bias of solutions when adding probing noise to systems if we use this developed off-policy Q-learning algorithm. Moreover, simulations show the implementation and control performance of the proposed algorithm.

Case 1: Consider the following open-loop unstable system:

$$x_{k+1} = \begin{bmatrix} -1 & 2 \\ 2.2 & 1.7 \end{bmatrix} x_k + \begin{bmatrix} 2 \\ 1.6 \end{bmatrix} u_k \tag{77}$$

Choose $Q=6$ and $R=1$. First, the optimal solution P^* was calculated by using command "dare" in Matlab. Thus, the optimal Q-function matrix H^* and the optimal controller gain K^* can be respectively obtained in terms of (65) and (68).

$$H^* = \begin{bmatrix} 96.4653 & -95.5203 & -96.9205 \\ -95.5203 & 289.2950 & 281.7826 \\ -96.9205 & 281.7826 & 281.3409 \end{bmatrix}$$
$$K^* = \begin{bmatrix} 0.3445 & -1.0016 \end{bmatrix} \tag{78}$$

Using three different probing noises, the unbiasedness of the off-policy Q-learning algorithm is verified compared with the on-policy Q-learning algorithm. The probing noise is respectively considered as

1:

$$e_k = 1.1(0.5\sin^2(2.0k)\cos(10.1k) + 0.9\sin^2(1.102k)\cos(4.001k)) \tag{79}$$

2:

$$e_k = 2.97(0.5\sin^2(2.0k)\cos(10.1k) + 0.9\sin^2(1.102k)\cos(4.001k)) \tag{80}$$

3:

$$e_k = 3.2(0.5\sin^2(2.0k)\cos(10.1k) + 0.9\sin^2(1.102k)\cos(4.001k)) \tag{81}$$

Table 1 respectively lists the convergence results of the iterative controller gain and means and variances of their differences from the theoretical optimal controller gains by using the on-policy Q-learning algorithm [8, 10, 14, 15] and the developed off-policy Q-learning algorithm under the above-mentioned three cases. In Table 1, 'N' denotes unavailable. For probing noise 1, the PE condition is not satisfied when implementing the on-policy Q-learning Algorithm 1, thus this algorithm cannot work. For probing noise 2 and 3, the PE condition is satisfied only at the first iteration and the third iteration, respectively. The learned controller gain shown in Table 1 cannot stabilize system (77) (see Fig. 2(a) with using K^3 of probing noise 3). It shows that the learned controller gains are incorrect. However, for all three probing noises, the controller gains can converge to the theoretical optimal values when implementing off-policy interleaved Q-learning after 10 iterations, which shows that adding probing noise cannot produce bias on learning solution of LQT problem unlike on-policy Q-learning. Fig. 2(b) and Fig.

2(c) show the state trajectories of the system and cost variation under the learned optimal control policy, respectively.

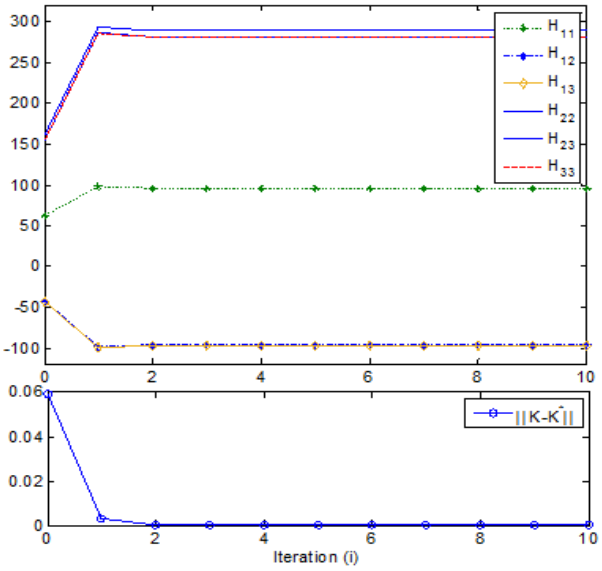


Fig. 1 Convergence of matrices H^i and K^i

Case 2: Now the developed off-policy interleaved Q-learning algorithm is verified in the following inverted pendulum system [37]:

$$\begin{bmatrix} x_{1(k+1)} \\ x_{2(k+1)} \end{bmatrix} = \begin{bmatrix} x_{1k} + \Delta t x_{2k} \\ \frac{g}{\iota}\Delta t \sin(x_{1k}) + (1 - \kappa\tau\Delta t)x_{2k} \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{\Delta t}{m\iota^2} \end{bmatrix} u_k \tag{82}$$

where the sampling interval $\Delta t = 0.1\text{s}$, $m = 0.5\text{kg}$ and $\iota = 1.0545\text{m}$ are the mass and length of the pendulum bar, respectively. Let $\kappa = 8.5415$ and $g = 3.1002\text{m}/(\text{s}^2)$ be the frictional factor and the gravitational acceleration, respectively. Let the initial state be $x_0 = [0.3 \ \ -0.3]^{\mathrm{T}}$, the structures of the inverted pendulum network, the critic and action networks be 3-6-1, 3-8-1 and 2-2-1, respectively. Choose $Q = \text{diag}(1,1)$ and $R = 0.1$.

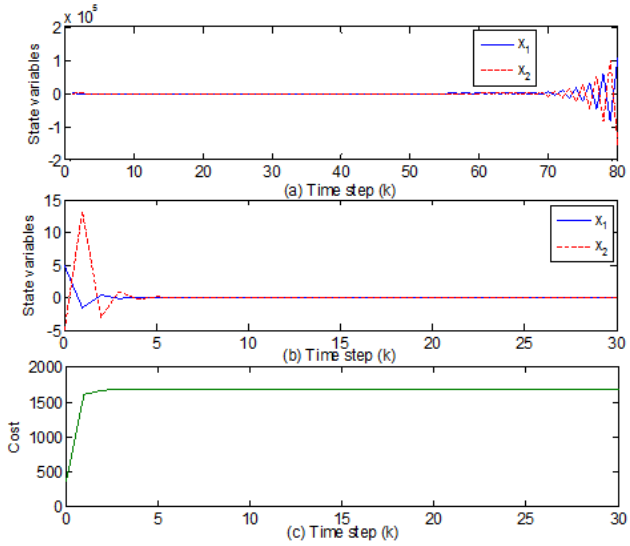


Fig. 2 The curves of state trajectories (a) using on-policy Q-learning, (b) using off-policy Q-learning and cost (c) using off-policy Q-learning

Let the learning rates of the inverted pendulum network, the critic and action networks respectively be 0.1, 0.3, 0.1. Let the training errors be 0.02 for these three neural networks. Fig. 3(a) shows the results of regulating neural network weights. Implementing the off-policy interleaved Q-learning Algorithm 3 yields the training or iteration results of the critic and actor networks as shown in Fig. 3(b) and Fig. 3(c). Thus, the approximate optimal control policy is learned, Fig. 4(a) presents the approximation of the optimal Q-function $Q^*(x_k, u^*(x_k))$. In the real operation of the inverted pendulum system, external disturbance and measurement errors are not completely avoided, so they are combined and assumed as $0.2e^{-0.0001k}\sin([2k \ \ 0]^{\mathrm{T}})$ and put it into (82). Fig. 4(b) and Fig. 4(c) are given to show the system sates under the approximate optimal control policy and the trajectory of the approximate optimal control policy, respectively. The performance $J^*(x_0)$ along with the system trajectories under the learned optimal control policy is plotted in Fig. 4(d) .

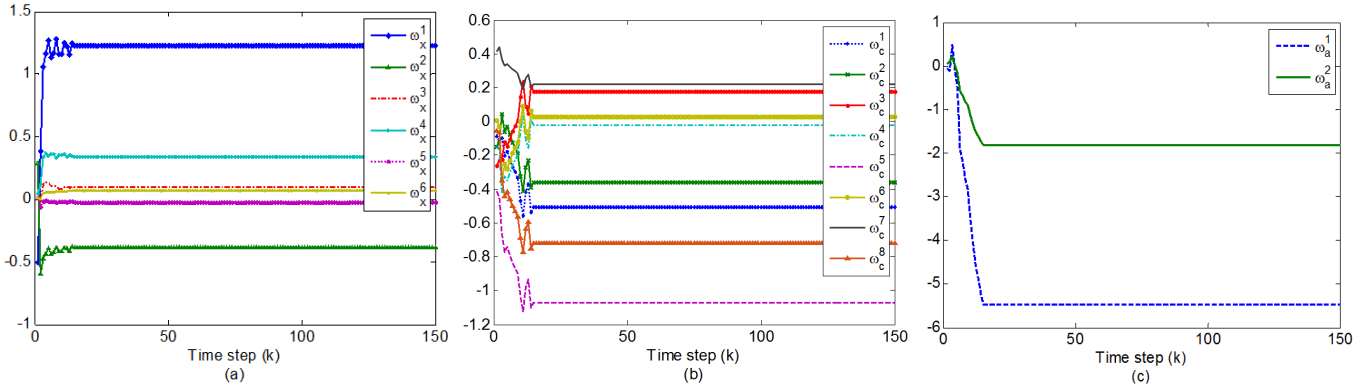


Fig. 3. The updating process of weights (a) of the inverted pendulum neural network; (b) of the critic neural network; and (c) the actor neural network

Case 3: Consider the following three-dimensional DT affine nonlinear system [33]:

$$\begin{bmatrix} x_{1(k+1)} \\ x_{2(k+1)} \\ x_{2(k+1)} \end{bmatrix} = \begin{bmatrix} x_{1k}x_{2k} \\ x_{1k}^2 - 0.5\sin(x_{2k}) \\ x_{3k} \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \\ -1 \end{bmatrix} u_k \tag{83}$$

Let $x_0 = [-0.5 \ 0.5 \ 1]^{\mathrm{T}}$, $Q = \mathrm{diag}(1,1,1)$ and $R = 0.1$. The model network, the critic network and the action network are built with the structures 4-3, 4-8-1 and 3-1, respectively. And the learning rates of these three networks are all set to be 0.1. For the critic network, the entries of input layer to hidden layer weight matrix are randomly generated in $[-0.15, 0.05]$ and then kept unchanged.

Under the probing noise $e_k = \mathrm{rand}(1,1)*0.2$, the performance is tested on 5 trials under the same scenario (the same initial neural network weights ω_x, $\hat{\omega}_{ak}^i$, $\hat{\omega}_{ck}^i$ and v_c). The results of 5 trials by using off-policy interleaved Q-learning Algorithm 3 are listed in Table 2. Approximate optimal control policy is quite hard to be found by using on-policy learning as not only neural network approximation but also adding probing noise might produce biased iterative Q-function solutions, as shown in these 5 trials wherein four testings are failed and the not good performance is obtained in one successful testing compared with the off-policy Q-learning method. Whereas adding probing noise wouldn't take any effect on precise solution of iterative Q-function and adequate exploration can be satisfied by using arbitrary behaviour control policy if the off-policy Q-learning algorithm is employed. Additionally, the iterative target control policy with probing noise has to act on the real system to learn the optimal control policy when running on-policy learning, which inevitably produces negative impact on performance of systems. Fig. 5(a) and Fig. 5(b) give the curves of state trajectories and the approximate optimal control laws that make the accumulated cost respectively reach 5.6789 and 195.1209 by using off-policy interleaved Q-learning and on-policy interleaved Q-learning.

Table 2 Performance comparisons of on-policy vs. off-policy learnings

Off-policy interleaved Q-learning		
	Approximate optimal cost $J(x_0)$	Operation time
1	3.4618	1.692s
2	7.8311	1.662s
3	5.6789	1.293s
4	5.7520	1.248 s
5	6.1557	1.505s
Average	5.7757	1.48s
standard deviation	1.5599	0.2046s
On-policy interleaved Q-learning		
1	195.1209	1.018s

VII. Conclusion

This paper focuses on presenting a novel off-policy interleaved Q-learning method for approximating the optimal control policy to achieve the optimum of affine nonlinear DT systems without knowing the dynamics of models. Based on the existing on-policy Q-learning methods for solving the optimal control problem, an off-policy Q-learning algorithm is developed and further the critic and actor structure based off-policy interleaved Q-learning algorithm is proposed. The rigorously theoretical proofs on the less sensitivity of solution of optimality problem to probing noise and the convergence of the proposed off-policy interleaved Q-learning are presented. Simulation results have demonstrated the effectiveness of the proposed method.

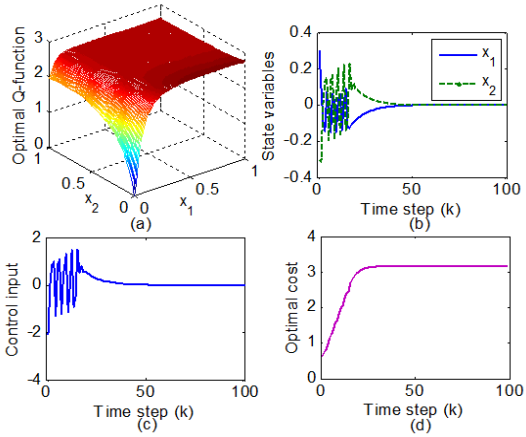


Fig. 4. Simulation results under the approximate optimal control policy

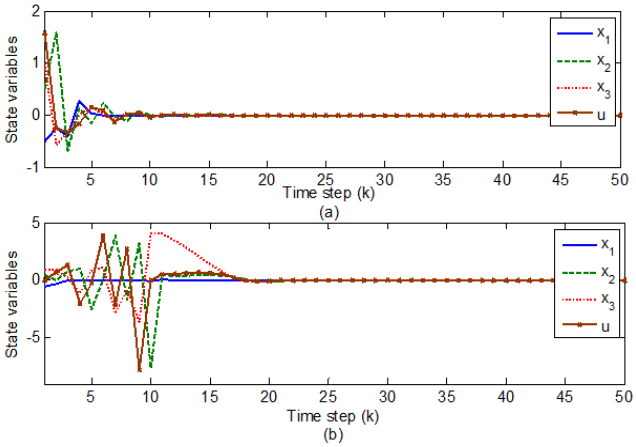


Fig. 5. The curves of state and control input (a) using off-policy interleaved Q-learning and (b) using on-policy interleaved Q-learning

References

[1]. F. L. Lewis and D. Liu, "Reinforcement learning and approximate dynamic programming for feedback control," *IEEE Circuits & Systems Magazine*, vol. 9, no. 3, pp. 32-50, Aug. 2009.

[2]. M. Riedmiller, "Neural fitted Q iteration-first experiences with a data efficient neural reinforcement learning method," *European Conference on Machine Learning*, 2005, pp. 317-328.

[3]. J. Fu, H. He, and X. Zhou, "Adaptive learning and control for MIMO system based on adaptive dynamic programming," *IEEE Transactions on Neural Networks*, vol. 22, no. 7, pp. 1133-1148, Jul. 2011.

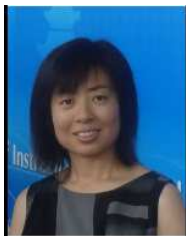
[4]. R. Munos, T. Stepleton, A. Harutyunyan and M. G. Bellemare, "Safe and efficient off-policy reinforcement learning," *Conference on Neural Information Processing Systems*, 2016, pp. 1-17.

[5]. T. Y. Chai, S. J. Qin, and H. Wang, "Optimal operational control for complex industrial processes," *Annual Reviews in Control*, vol. 38, no. 1, pp. 81-92, Feb. 2014.

[6]. T. Y. Chai, J. L. Ding, and F. Wu, "Hybrid intelligent control for optimal operation of shaft furnace roasting process," *Control Engineering Practice*, vol. 19, no. 3, pp. 264-275, Mar. 2011.

[7]. D. Wang, H. B. He, X. N. Zhong, and D. R. Liu, "Event-driven nonlinear discounted optimal regulation involving a power system application," *IEEE Transactions on Industrial Electronics*, vol. 64, no. 10, pp. 8177-8186, Oct.

2017.
[8]. Q. L. Wei, D. R. Liu, and G. Shi, "A novel dual iterative Q-learning method for optimal battery management in smart residential environments," *IEEE Transactions on Industrial Electronics*, vol. 62, no. 4, pp. 2509-2518, Apr. 2015.
[9]. F. L. Lewis, H. Modares, A. Karimpour, M. B. Naghibisistani, and B. Kiumarsi, "Reinforcement Q-learning for optimal tracking control of linear discrete-time systems with unknown dynamics," *Automatica*, vol. 50, no. 4, pp. 1167-1175, Apr. 2014.
[10]. A. Al-Tamimi, F. L. Lewis, and M. Abu-Khalaf, "Model-free Q -learning designs for linear discrete-time zero-sum games with application to H-infinity control," *Automatica*, vol. 43, no. 3, pp. 473-481, Mar. 2007.
[11]. C. J. C. H. Watkins and P. Dayan, "Q-learning," *Machine Learning*, vol. 8, no. 3-4, pp. 279-292, 1992.
[12]. J. N. Tsitsiklis, "Asynchronous stochastic approximation and Q-learning," *Machine Learning*, vol. 16, no. 3, pp. 185-202, Sep. 1993.
[13]. S. Doltsinis, P. Ferreira, and N. Lohse, "An MDP model-based reinforcement learning approach for production station ramp-up optimization: Q-learning analysis," *IEEE Transactions on Systems Man & Cybernetics Systems*, vol. 44, no. 9, pp. 1125-1138, Sep. 2014.
[14]. J. H. Kim and F. L. Lewis, "Model-free H∞ control design for unknown linear discrete-time systems via Q-learning with LMI," *Automatica*, vol. 46, no. 8, pp. 1320-1326, Aug. 2010.
[15]. Yi Jiang, Jialu Fan, Tianyou Chai, Frank L. Lewis, and J. N. Li, "Tracking control for linear discrete-time Networked control systems with unknown dynamics and dropout," IEEE Transactions on Neural Networks and Learning Systems, to be published, Doi: 10.1109/TNNLS.2017.2771459, Dec. 2017.
[16]. J. Y. Lee, J. B. Park, and Y. H. Choi, "Integral Q-learning and explorized policy iteration for adaptive optimal control of continuous-time linear systems," *Automatica*, vol. 48, no. 11, pp. 2850-2859, Nov. 2012.
[17]. K. G. Vamvoudakis, "Q-learning for continuous-time linear systems: A model-free infinite horizon optimal control approach," *Systems & Control Letters* , vol. 100, pp. 14-20, 2017.
[18]. K. G. Vamvoudakis, "Q-learning for continuous-time graphical games on large networks with completely unknown linear system dynamics," International Journal of Robust & Nonlinear Control, Doi: 10.1002/rnc.3719, Nov. 2016.
[19]. B. Kiumarsi, F. L. Lewis, and Z. P. Jiang, "H∞ control of linear discrete-time systems: Off-policy reinforcement learning," *Automatica*, vol. 37, no. 1, pp. 144-152, Apr. 2017.
[20]. J. N. Li, B. Kiumarsi, T. Y. Chai, F. L. Lewis, and J. L. Fan, "Off-policy reinforcement learning: optimal operational control for two-time-scale industrial processes," *IEEE Transactions on Cybernetics*, vol. 47, no. 12, pp. 4547- 4558, Dec. 2017.
[21]. J. N. Li, T. Y. Chai, F. L. Lewis, J. L. Fan, Z. T. Ding, and Z. L. Ding, "Off-policy Q-learning: set-point design for optimizing dual-rate rougher flotation operational processes," *IEEE Transactions on Industrial Electronics*, vol. 65, no. 5, pp. 4092-4102, May 2018.
[22]. R. Song, F. L. Lewis, Q. Wei, and H. Zhang, "Off-policy actor-critic structure for optimal control of unknown systems with disturbances," *IEEE Transactions on Cybernetics*, vol. 46, no. 5, pp. 1041-1050, Apr. 2015.
[23]. B. Luo, H. N. Wu, and T. Huang, "Off-policy reinforcement learning for H∞ control design," *IEEE Transactions on Cybernetics*, vol. 45, no. 1, pp. 65-76, Jan. 2015.
[24]. Y. Jiang and Z. P. Jiang, "Computational adaptive optimal control for continuous-time linear systems with completely unknown dynamics," *Automatica*, vol. 48, no. 10, pp. 2699-2704, 2012.
[25]. J. N. Li, H. Modares, T. Y. Chai, F. L. Lewis, and L. H. Xie, "Off-policy reinforcement learning for synchronization in multi-agent graphical games," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 28, no. 10, pp. 2434-2445, Oct. 2017.
[26]. B. Luo, D. R. Liu, T. Huang, and D. Wang, "Model-free optimal tracking control via critic-only Q-learning," *IEEE Transactions on Neural Networks & Learning Systems*, vol. 27, no. 10, pp. 2134-2144, Oct. 2016.
[27]. Y. L. Wang, Q. L. Han, M. R. Fei, and C. Peng, "Network-based T-S fuzzy dynamic positioning controller design for unmanned marine vehicles," *IEEE Transactions on Cybernetics*, Doi: 10.1109/TCYB.2018.2829730, Apr. 2018.
[28]. P. J. Werbos, *Approximate dynamic programming for real-time control and neural modeling*. In Handbook of Intelligent Control Neural Fuzzy & Adaptive Approaches, New York, USA: Van Nostrand Reinhold, 1992.
[29]. D. R. Liu and Q. L. Wei, "Policy iteration adaptive dynamic programming algorithm for discrete-time nonlinear systems," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 25, no. 3, pp. 621-634, March 2014.
[30]. D. Wang, D. R. Liu, C. X. Mu, and Y. Zhang, "Neural network learning and robust stabilization of nonlinear systems with dynamic uncertainties," IEEE Transactions on Neural Networks and Learning Systems, vol. 29, no. 4, pp. 1342-1351, Apr. 2018.
[31]. D. V. Prokhorov, R. A. Santiago, and D. C. W. Ii, "Adaptive critic designs: A case study for neurocontrol," Neural Networks, vol. 8, no. 9, pp. 1367-1372, 1995.
[32]. D. Wang, H. B. He, and D. R. Liu, "Adaptive critic nonlinear robust control: A survey," IEEE T ransactions on Cybernetics, vol. 47, no. 10, pp. 3429-3451, Jul. 2017.
[33]. C. X. Mu, D. Wang, and H. B. He, "Novel iterative neural dynamic programming for data-based approximate optimal control design," *Automatica*, vol. 81, pp. 240-252, Apr. 2017.
[34]. F. L. Lewis, D. Vrabie, and K. G. Vamvoudakis, "Reinforcement learning and feedback control: Using natural decision methods to design optimal adaptive controllers," *IEEE Control Syst*., vol. 32, no. 6, pp. 76-105, Dec. 2012.
[35]. Y. Jiang, J. L. Fan, T. Y. Chai, J. N. Li, and F. L. Lewis, "Data-Driven Flotation Industrial Process Operational Optimal Control Based on Reinforcement Learning," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 5, pp. 1974-1989, May 2018.
[36]. K. G. Vamvoudakis, Q. Yang, and S. Jagannathan, "Reinforcement learning controller design for affine nonlinear discrete-time systems using online approximators," *IEEE Transactions on Systems Man & Cybernetics Part B*, vol. 42, no. 2, pp. 377-390, Apr. 2012.
[37]. R. W. Beard, "Improving the closed-loop performance of nonlinear systems," Ph. D. dissertation, Dept. Electr. Eng., Rensselaer Polytech. Inst., Troy, NY, USA, 1995.



Jinna Li received the M.S. degree and the Ph. D. degree from Northeastern University, Shenyang, China, 2006 and 2009, respectively. She is an associate professor at Shenyang University of Chemical Technology, Shenyang, China.

From April 2009 to April 2011, she was a postdoctor with the Lab of Industrial Control Networks and Systems, Shenyang Institute of Automation, Chinese Academy of Sciences. From June 2014 to June 2015, she was a Visiting Scholar granted by China Scholarship Council with Energy Research Institute, Nanyang Technological University, Singapore. From September 2015 to June 2016, she was a Domestic Young Core Visiting Scholar granted by Ministry of Education of China with State Key Lab of Synthetical Automation for Process Industries, Northeastern University. From Jan. 2017 to Jul. 2017, she was a Visiting Scholar with the School of Electrical and Electronic Engineering, the University of Manchester, UK. Her current research interests include neural networks, reinforcement learning, optimal operational control, distributed optimization control and data-based control.

Tianyou Chai (F'08) received the Ph. D. degree in control theory and engineering in 1985 from Northeastern University, Shenyang, China, where he became a Professor in 1988. He is the founder and Director of the Center of Automation, which became a National Engineering and Technology Research Center and a State Key Laboratory. He is a member of Chinese Academy of Engineering, IFAC Fellow and IEEE Fellow, director of Department of Information Science of National Natural Science Foundation of China. His current research interests include modeling, control, optimization and integrated automation of complex industrial processes. He has published 144 peer reviewed international journal papers. He has developed control technologies with applications to various industrial processes. For his contributions, he has won 4 prestigious awards of National Science and Technology Progress and National Technological Innovation, the 2007 Industry Award for Excellence in Transitional Control Research from IEEE Multiple-conference on Systems and Control.

Frank. L. Lewis (F'94) received the bachelor's degree in physics/electrical engineering and the M.S. degree in electrical engineering from Rice University, Houston, TX, USA, the M.S. degree in aeronautical engineering from the University of West Florida, Pensacola, FL, USA, and the Ph.D. degree from the Georgia Institute of Technology, Atlanta, GA, USA.

He is currently a U.K. Chartered Engineer, the IEEE Control Systems Society Distinguished Lecturer, a University of Texas at Arlington Distinguished Scholar Professor, a UTA Distinguished Teaching Professor, and a Moncrief-O'Donnell Chair with the University of Texas at Arlington Research Institute, Fort Worth, TX, USA. He is a Qian Ren Thousand Talents Consulting Professor with Northeastern University, Shenyang, China. He is involved in feedback control, reinforcement learning, intelligent systems, and distributed control systems. He has authored six U.S. patents, 301 journal papers, 396 conference papers, 20 books, 44 chapters, and 11 journal special issues. Dr. Lewis is a fellow of the International Federation of Automatic Control, the U.K. Institute of Measurement and Control, and Professional Engineer at Texas. He was a recipient of the IEEE Computational Intelligence Society Neural Networks Pioneer Award in 2012, the Distinguished Foreign Scholar from the Nanjing University of Science and Technology, the 111 Project Professor at Northeastern University, China, the Outstanding Service Award from the Dallas IEEE Section, an Engineer of the Year from the Fort Worth IEEE Section. He was listed in Fort Worth Business Press Top 200 Leaders in Manufacturing. He was also a recipient of the 2010 IEEE Region Five Outstanding Engineering Educator Award. He also received the IEEE Control Systems Society Best Chapter Award (as a Founding Chairman of DFW Chapter) in 1996.

Zhengtao Ding (SM'03) received the B. Eng. degree from Tsinghua University, Beijing, China, and the M.Sc. degree in systems and control and the Ph. D. degree in control systems from the University of Manchester Institute of Science and Technology, Manchester, U.K. After working as a Lecturer with Ngee Ann Polytechnic, Singapore, for ten years, in 2003, he joined The University of Manchester, Manchester, U.K., where he is currently a professor of control engineering with the School of Electrical and Electronic Engineering. He is the author of the book Nonlinear and Adaptive Control Systems (IET, 2013) and a number of journal papers. His research interests include nonlinear and adaptive control theory and their applications. Dr. Ding serves as an Associate Editor for the IEEE Transactions on Automatic Control, Transactions of the Institute of Measurement and Control, Control Theory and Technology, Mathematical Problems in Engineering, Unmanned Systems and the International Journal of Automation and Computing.

Yi Jiang (S'14) was born in Hubei Province, China. He received the B. Eng. degree in automation and M.S. degree in control theory and control engineering from information science and engineering college and State Key Laboratory of Synthetical Automation for Process Industries in Northeastern University, Shenyang, Liaoning, China in 2014 and 2016, respectively, where he is currently working toward the Ph.D. degree. From January to July, 2017, he was a Visiting Scholar with the UTA Research Institute, University of Texas at Arlington, TX, USA. From March 2018 to March 2019, he is a Research Assistant with the University of Alberta, Edmonton, Canada. His research interests include networked control systems, industrial process opertaionl control and reinforcement learning.