IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS

A PUBLICATION OF THE IEEE COMPUTATIONAL INTELLIGENCE SOCIETY

http://cis.ieee.org/tnnls

IEEE Computational Intelligence Society

OCTOBER 2016 VOLUME 27 NUMBER 10 ITNNEP (ISSN 2162-237X)

REGULAR PAPERS

Why Deep Learning Works: A Manifold Disentanglement Perspective .... *P. P. Brahma, D. Wu, and Y. She* 1997

Feedback Solution to Optimal Switching Problems With Switching Cost .... *A. Heydari* 2009

Geometric Bioinspired Networks for Recognition of 2-D and 3-D Low-Level Structures and Transformations .... *E. Bayro-Corrochano, E. Vazquez-Santacruz, E. Moya-Sanchez, and E. Castillo-Muñis* 2020

Classifying Stress From Heart Rate Variability Using Salivary Biomarkers as Reference .... *W. S. Liew, M. Seera, C. K. Loo, E. Lim, and N. Kubota* 2035

Space Structure and Clustering of Categorical Data .... *Y. Qian, F. Li, J. Liang, B. Liu, and C. Dang* 2047

Kohonen's Map Approach for the Belief Mass Modeling ..... *I. Hammami, G. Mercier, A. Hamouda, and J. Dezert* 2060

Multiple Ordinal Regression by Maximizing the Sum of Margins .... *O. C. Hamsici and A. M. Martinez* 2072

Adaptive Filter Design Using Type-2 Fuzzy Cerebellar Model Articulation Controller .... *C.-M. Lin, M.-S. Yang, F. Chao, X.-M. Hu, and J. Zhang* 2084

Directional Clustering Through Matrix Factorization .... *T. Blumensath* 2095

Learning With Jensen–Tsallis Kernels .... *D. Ghoshdastidar, A. P. Adsul, and A. Dukkipati* 2108

Tensor LRR and Sparse Coding-Based Subspace Clustering .... *Y. Fu, J. Gao, D. Tien, Z. Lin, and X. Hong* 2120

Model-Free Optimal Tracking Control via Critic-Only Q-Learning .... *B. Luo, D. Liu, T. Huang, and D. Wang* 2134

IEEE

Model-Free Optimal Tracking Control via Critic-Only Q-Learning

Biao Luo, *Member, IEEE*, Derong Liu, *Fellow, IEEE*, Tingwen Huang, and Ding Wang, *Member, IEEE*

***Abstract*— Model-free control is an important and promising topic in control fields, which has attracted extensive attention in the past few years. In this paper, we aim to solve the model-free optimal tracking control problem of nonaffine nonlinear discrete-time systems. A critic-only Q-learning (CoQL) method is developed, which learns the optimal tracking control from real system data, and thus avoids solving the tracking Hamilton–Jacobi–Bellman equation. First, the Q-learning algorithm is proposed based on the augmented system, and its convergence is established. Using only one neural network for approximating the Q-function, the CoQL method is developed to implement the Q-learning algorithm. Furthermore, the convergence of the CoQL method is proved with the consideration of neural network approximation error. With the convergent Q-function obtained from the CoQL method, the adaptive optimal tracking control is designed based on the gradient descent scheme. Finally, the effectiveness of the developed CoQL method is demonstrated through simulation studies. The developed CoQL method learns with off-policy data and implements with a critic-only structure, thus it is easy to realize and overcome the inadequate exploration problem.**

***Index Terms*— Critic-only Q-learning (CoQL), model-free, nonaffine nonlinear systems, optimal tracking control.**

I. Introduction

REINFORCEMENT learning (RL) is an important topic in the machine learning community, which mainly aims at solving the optimal control problem of Markov decision process (MDP) [1]–[6]. RL technique refers to an actor or agent that interacts with its environment and aims to learn the optimal control policy, by observing their responses from the environment. Over the past years, RL has appeared as a powerful tool for solving control problems [7]–[10], [12]–[15], [17]–[33]. However, it is noted that most of the RL-based control approaches focused on the regulation control problem, a few of which are [7]–[9], [11], [12], [15], [18], [19], and [22]. For many practical systems, such as hypersonic aircraft [35]–[37], spacecraft [38], [39], motion tracking [40], etc., it is required to design a controller such that the desired reference trajectories can be tracked and the optimal performance can be achieved. By considering these two goals, the optimal tracking control problem [14], [41]–[54] has received increasing attention in recent years.

Some works have been reported for solving the optimal tracking problem based on the expected control. The expected control is first derived with the desired reference trajectories and the system model. Then, the expected control and state errors are employed to define the performance index. Hence, the optimal tracking control problem is then reformulated as the optimal regulation problem of the error system with respect to the performance index, and then some RL approaches were proposed, such as, heuristic dynamic programming algorithm for nonlinear discrete-time systems [53] and with time delays [50]. By considering approximation errors, value iteration algorithms were presented to obtain the optimal tracking control [54]. An approximate dynamic programming method [44] was proposed for nonlinear continuous-time systems, and its stability was proved. In [49], a prior model identification procedure was conducted for nonlinear continuous-time systems, and then model-based adaptive methods were used for optimal tracking control design. Note that the analytical expression of the expected control is required, which depends on the system model, and thus all these methods are model based.

To avoid using the expected control explicitly, an augmented system can be obtained with the error system and the command system of the desired reference trajectories. By introducing a discounted performance index, the optimal regulation problem of the augmented system is reformulated and then be solved with RL-based approaches. Without requiring the internal system dynamics, online policy iteration methods were employed for the optimal tracking control design of linear continuous-time systems [45] and nonlinear continuous-time systems with control constraints [43], nonlinear discrete-time systems with control constraints [51], and nonlinear time-varying discrete-time systems [55]. Without a complete system model, the optimal tracking control problem of linear discrete-time systems was solved with input–output data [48]. For the linear continuous-time systems [56], the original performance index was employed, which requires a stable command system.

As one of the powerful RL methods, Q-learning has been studied for a long time in the machine learning community for

Manuscript received February 25, 2016; revised April 16, 2016; accepted June 19, 2016. Date of publication July 12, 2016; date of current version September 15, 2016. This work was supported in part by the National Natural Science Foundation of China under Grant 61233001, Grant 61273140, Grant 61304086, Grant 61374105, Grant 61503377, Grant 61533017, and Grant U1501251, in part by the Early Career Development Award of the State Key Laboratory of Management and Control for Complex Systems, and in part by the National Priorities Research Program through the Qatar National Research Fund (a member of Qatar Foundation) under Grant NPRP 7-1482-1-278. The acting Editor-in-Chief who handled the review of this paper was Cristiano Cervellera.

B. Luo and D. Wang are with the State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China (e-mail: biao.luo@hotmail.com; ding.wang@ia.ac.cn).

D. Liu is with the School of Automation and Electrical Engineering, University of Science and Technology Beijing, Beijing 100083, China (e-mail: derong@ustb.edu.cn).

T. Huang is with Texas A&M University at Qatar, Doha 23874, Qatar (e-mail: tingwen.huang@qatar.tamu.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TNNLS.2016.2585520

the problem of MDP [57]–[63]. However, till most recently, only a few Q-learning techniques have been introduced for solving control problems [22], [42], [64]–[68]. For linear discrete-time systems, Q-learning methods were proposed to solve the H_∞ control problem [64], [65] and the optimal tracking control problem [42]. For linear continuous-time systems, online Q-learning algorithms [66], [68] were investigated to solve the linear quadratic regulation problem. It is noted that most of these works are just for simple linear systems [42], [64]–[66], [68]. However, the model-free optimal tracking control of general nonlinear systems has rarely been studied with the Q-learning method.

The motivation of this paper aims to achieve the following three important goals simultaneously for control design.

1) Solve the model-free optimal tracking control problem of general nonaffine nonlinear systems. This problem still remains an open issue, which has rarely been studied.
2) Controller design with off-policy data. For model-free control design problems, how to collect system data and make efficient use of them for learning are important tasks. On-policy and off-policy learning schemes are two main RL frameworks for control design. Compared with the on-policy scheme, the off-policy learning [69]–[75] has several merits, which is regarded as more practical and efficient. Off-policy schemes permit the use of any control policy to generate data, while the exact target policy must be employed for on-policy schemes that is usually difficult to operate. This means that collecting off-policy data is easier than the on-policy data. Thus, the use of off-policy data for model-free tracking control design is the important goal we aim to achieve.
3) Use critic-only implementation structure. For the model-free control problem of general nonaffine nonlinear systems, it is still difficult to use the critic-only implementation structure where only one NN is required.

To the best of our knowledge, the model-free optimal tracking control problem involving the above three goals has not been studied yet. The main contribution of this paper is the development of the critic-only Q-learning (CoQL) method, which achieves the above three goals simultaneously for the optimal tracking control design. The detailed contributions compared with the existing related works will be analyzed in Section IV.

The rest of this paper is organized as follows. Section II presents the optimal tracking control problem and the tracking Hamilton–Jacobi–Bellman equation (HJBE). Q-learning algorithm is proposed in Section III and the CoQL method is developed for model-free optimal tracking control design in Section IV. Simulation results are demonstrated in Section V and brief conclusions and future works are presented in Section VI.

II. Optimal Tracking Control Problem

In this section, the optimal tracking control problem for general nonaffine nonlinear discrete-time systems is presented. Theoretically, the problem is converted to solve a tracking HJBE.

A. Problem Description

Let us consider the following nonaffine nonlinear discrete-time systems:

$$x(k+1) = f(x(k), u(k)) \tag{1}$$

where $x(k) \in \mathbb{R}^n$ is the state and $u(k) \in \mathbb{R}^m$ is the control input. It is assumed that the system (1) is stabilizable on the set $\mathcal{X}$ and $f(0,0) = 0$.

Let $r(k) \in \mathbb{R}^n$ be the desired reference trajectory. For the optimal tracking control problem, the objective is to design the control input $u(k)$ for the system (1), such that the state $x(k)$ tracks $r(k)$ and minimizes the performance index. Assume that $r(k)$ is bounded and generated by the command system

$$r(k+1) = h(r(k)) \tag{2}$$

where $h(r)$ is a Lipschitz continuous vector function with $h(0) = 0$. Denoting the tracking error as $e(k) \triangleq x(k) - r(k)$, it follows from (1) and (2) that:

$$e(k+1) = f(e(k) + r(k), u(k)) - h(r(k)). \tag{3}$$

Define the state of the augmented system as $y(k) \triangleq [e^{\mathsf{T}}(k) \quad r^{\mathsf{T}}(k)]^{\mathsf{T}}$. Then, combining (2) and (3) yields the following augmented system:

$$y(k+1) = F(y(k), u(k)) \tag{4}$$

where $y(0) = [e^{\mathsf{T}}(0) \quad r^{\mathsf{T}}(0)]^{\mathsf{T}}$ and

$$F(y(k), u(k)) \triangleq \begin{bmatrix} f(e(k) + r(k), u(k)) - h(r(k)) \\ h(r(k)) \end{bmatrix}. \tag{5}$$

In this paper, we consider the model-free optimal tracking control problem of the system (1) with the following discounted performance index:

$$J(y(0), u) \triangleq \sum_{l=0}^{\infty} \gamma^l \mathcal{R}(y(l), u(l)) \tag{6}$$

where $0 < \gamma \leqslant 1$ is the discount factor and $\mathcal{R}(y,u) \triangleq W(e) + R(u)$ with $W(e)$ and $R(u)$ positive definite functions, i.e., $W(e) > 0$, $R(u) > 0$ for $\forall e \neq 0$, $u \neq 0$, and $W(e) = 0$, $R(u) = 0$ only when $e = 0$, $u = 0$. Then, the optimal tracking control problem of the system (1) is converted into an optimal regulation control problem, i.e., finding the following optimal control:

$$u^*(y) \triangleq \arg\min_u J(y(0), u) \tag{7}$$

with respect to the augmented system (4) and the performance index (6).

Remark 1: For the considered model-free optimal tracking control problem, the meaning of the term model-free involves two aspects.

1) The mathematical models of the system (1) and the command system (2) are unknown, i.e., $f(x,u)$ and $h(r)$ are unknown. Thus, it follows from (5) that the mathematical model of $F(y,u)$ is unknown.
2) The explicit expression of $\mathcal{R}(y,u)$ in the performance index (6) is unknown.

Remark 2: In [42], [43], [45], [48], [51], and [55], the discounted performance index has also been used to study the optimal tracking problem. For the discount factor γ in the performance index (6), it can be discussed from two aspects. On the one hand, $\gamma = 1$ can be only employed if one knows *a priori* that the reference trajectory is generated by an asymptotically stable command system (6). On the other hand, if the desired reference trajectory r is a general bounded signal, it is required to choose a γ such that $0 < \gamma < 1$, which will result in a bounded performance index (6). Thus, the optimization of the performance index (6) does not require $r \to 0$ and $y \to 0$ as time increases.

B. Tracking HJBE

Let $y \in \mathcal{Y}$, $u \in \mathcal{U}$, where $\mathcal{Y}$ and $\mathcal{U}$ are two compact sets, and denote $\mathcal{D} \triangleq \{(y,u)|y \in \mathcal{Y}, u \in \mathcal{U}\}$. For an admissible control policy $u(y)$, define its value function as

$$V_u(y(k)) \triangleq \sum_{l=k}^{\infty} \gamma^{l-k} \mathcal{R}(y(l), u(l)) \tag{8}$$

which can be rewritten as the following Bellman equation:

$$\begin{aligned} V_u(y(k)) &= \mathcal{R}(y(k), u(k)) + \sum_{l=k+1}^{\infty} \gamma^{l-k} \mathcal{R}(y(l), u(l)) \\ &= \mathcal{R}(y(k), u(k)) + \gamma \sum_{l=k+1}^{\infty} \gamma^{l-(k+1)} \mathcal{R}(y(l), u(l)) \\ &= \mathcal{R}(y(k), u(k)) + \gamma V_u(y(k+1)). \end{aligned} \tag{9}$$

The optimal control law (7) can be rewritten as

$$u^*(y(k)) \triangleq \arg\min_u V_u(y(k)). \tag{10}$$

Denoting the optimal value function as $V^*(y) \triangleq V_{u^*}(y)$, the tracking HJBE is given as follows:

$$\begin{aligned} V^*(y(k)) &= \min_u \{\mathcal{R}(y(k), u(k)) + \gamma V^*(y(k+1))\} \\ &= \mathcal{R}(y(k), u^*(k)) + \gamma V^*(y(k+1)) \end{aligned} \tag{11}$$

which is a nonlinear difference equation. Note that the optimal control policy $u^*(y)$ depends on the solution of the tracking HJBE (11), which is difficult to solve for nonlinear systems. Even worse, the unavailability of $F(y,u)$ and $\mathcal{R}(y,u)$ prevents using model-based methods to solve the tracking HJBE for the optimal value function V^*. To overcome these difficulties, we developed a model-free CoQL method for the direct optimal tracking control design with real system data.

III. Q-Learning for Tracking Control

In this section, a Q-learning algorithm is proposed and its convergence theory is established.

A. Q-Learning Algorithm

For an admissible control policy $u(y)$, define its Q-function as

$$Q_u(y(k), a) \triangleq \mathcal{R}(y(k), a) + \sum_{l=k+1}^{\infty} \gamma^{l-k} \mathcal{R}(y(l), u(y(k))) \tag{12}$$

where $Q_u(0,0) = 0$. From (9) and (12)

$$\begin{aligned} Q_u(y(k), a) &= \mathcal{R}(y(k), a) + \gamma \sum_{l=k+1}^{\infty} \gamma^{l-(k+1)} \mathcal{R}(y(l), u(l)) \\ &= \mathcal{R}(y(k), a) + \gamma V_u(y(k+1)). \end{aligned} \tag{13}$$

The Q-function $Q_u(y,a)$ is an action-state value function, which represents the value of the performance metric obtained when control action a is used at state y and the control policy u is pursued thereafter. For the optimal control policy $u^*(y)$, it follows from (13) that the associated optimal Q-function $Q^*(y,a) \triangleq Q_{u^*}(y,a)$ is given by

$$Q^*(y(k), a) = \mathcal{R}(y(k), a) + \gamma V^*(y(k+1)). \tag{14}$$

According to (10) and (14), the optimal control policy $u^*(y)$ can also be represented as

$$u^*(y) = \arg\min_u V_u(y) = \arg\min_a Q^*(y,a). \tag{15}$$

To avoid solving the tracking HJBE (11) and using system model, the following Q-learning algorithm is proposed to learn the optimal Q-function $Q^*(y,a)$ and optimal tracking control $u^*(y)$ from real system data.

```
Algorithm 1 Q-Learning
▶ Step 1: Let $u^{(0)}(y)$ be an initial admissible control policy, and $i = 0$;
▶ Step 2: (Policy evaluation) Solve the equation
      $Q^{(i)}(y(k), a) = \mathcal{R}(y(k), a) + \gamma Q^{(i)}(y(k+1), u^{(i)})$   (16)
  for the unknown Q-function $Q^{(i)} \triangleq Q_{u^{(i)}}$;
▶ Step 3: (Policy improvement) Update control policy with
      $u^{(i+1)}(y) = \arg\min_a Q^{(i)}(y, a)$;   (17)
▶ Step 4: Let $i = i + 1$, go back to Step 2 and continue.
```

B. Convergence Analysis of Q-Learning

From (8), denote

$$\begin{aligned} V^{(i)}(y(k)) &\triangleq V_{u^{(i)}}(y(k)) \\ &= \sum_{l=k}^{\infty} \gamma^{l-k} \mathcal{R}(y(l), u^{(i)}(l)). \end{aligned} \tag{18}$$

According to (13) and (18), we have

$$\begin{aligned} Q^{(i)}(y(k), u^{(i)}) &= \mathcal{R}(y(k), u^{(i)}) + \gamma V^{(i)}(y(k+1)) \\ &= V^{(i)}(y(k)). \end{aligned} \tag{19}$$

Note that the Q-learning Algorithm 1 generates the sequences $\{Q^{(i)}(y,a)\}$ and $\{u^{(i)}(y)\}$, which are proved to converge to $Q^*(y,a)$ and $u^*(y)$, respectively, in the following Theorem 1.

Theorem 1: For $\forall (y,a) \in \mathcal{Y} \times \mathcal{U}$, the sequences $\{Q^{(i)}(y,a)\}$ and $\{u^{(i)}(y)\}$ are generated by Algorithm 1. Then

1) $Q^{(i)}(y,a) \geqslant Q^{(i+1)}(y,a) \geqslant Q^*(y,a)$;
2) $\lim_{i\to\infty} Q^{(i)}(y,a) = Q^*(y,a)$ and $\lim_{i\to\infty} u^{(i)}(y) = u^*(y)$.

Proof:

1) For $\forall(y(k), a) \in \mathcal{Y} \times \mathcal{U}$, based on (13) and (16), we obtain

$$\begin{aligned}
&Q^{(i+1)}(y(k), a)\\
&= \mathcal{R}(y(k), a) + \gamma\, Q^{(i+1)}(y(k+1), u^{(i+1)})\\
&= \mathcal{R}(y(k), a) + \gamma\, V^{(i+1)}(y(k+1))\\
&= \mathcal{R}(y(k), a) + \gamma\, [\mathcal{R}(y(k+1), u^{(i+1)})\\
&\quad + \gamma\, V^{(i+1)}(y(k+2))]\\
&= \mathcal{R}(y(k), a) + \gamma\, [\mathcal{R}(y(k+1), u^{(i+1)}) + \gamma\, V^{(i)}(y(k+2))]\\
&\quad - \gamma^2 V^{(i)}(y(k+2)) + \gamma^2 V^{(i+1)}(y(k+2))\\
&= \mathcal{R}(y(k), a) + \gamma\, Q^{(i)}(y(k+1), u^{(i+1)})\\
&\quad - \gamma^2 V^{(i)}(y(k+2)) + \gamma^2 V^{(i+1)}(y(k+2)). \qquad (20)
\end{aligned}$$

Based on (17)

$$\begin{aligned}
Q^{(i)}(y(k+1), u^{(i+1)}) &= \min_a Q^{(i)}(y(k+1), a)\\
&\leqslant Q^{(i)}(y(k+1), u^{(i)})\\
&= V^{(i)}(y(k+1)). \qquad (21)
\end{aligned}$$

Combining (20) and (21) yields

$$\begin{aligned}
&Q^{(i+1)}(y(k), a)\\
&\quad\leqslant \mathcal{R}(y(k), a) + \gamma\, V^{(i)}(y(k+1))\\
&\qquad - \gamma^2 V^{(i)}(y(k+2)) + \gamma^2 V^{(i+1)}(y(k+2))\\
&\quad= \mathcal{R}(y(k), a) + \gamma\, \mathcal{R}(y(k+1), u^{(i)})\\
&\qquad + \gamma^2 V^{(i+1)}(y(k+2))\\
&\quad\leqslant \mathcal{R}(y(k), a) + \gamma\, \mathcal{R}(y(k+1), u^{(i)})\\
&\qquad + \gamma^2 \mathcal{R}(y(k+2), u^{(i)}) + \gamma^3 V^{(i+1)}(y(k+3))\\
&\quad\leqslant \mathcal{R}(y(k), a) + \gamma\, \mathcal{R}(y(k+1), u^{(i)})\\
&\qquad + \gamma^2 \mathcal{R}(y(k+2), u^{(i)}) + \gamma^3 \mathcal{R}(y(k+3), u^{(i)}) + \cdots\\
&\quad= \mathcal{R}(y(k), a) + \gamma \sum_{l=k+1}^{\infty} \gamma^{\,l-(k+1)} \mathcal{R}(y(l), u^{(i)})\\
&\quad= \mathcal{R}(y(k), a) + \gamma\, V^{(i)}(y(k+1))\\
&\quad= Q^{(i)}(y(k), a) \qquad (22)
\end{aligned}$$

that is, $Q^{(i+1)}(y, a) \leqslant Q^{(i)}(y, a)$ for $\forall(y, a) \in \mathcal{Y} \times \mathcal{U}$. Note that $Q^*(y, a) \leqslant Q^{(i)}(y, a)$ for $\forall i$. Thus, $Q^{(i)}(y, a) \geqslant Q^{(i+1)}(y, a) \geqslant Q^*(y, a)$.

2) Part 1) of Theorem 1 shows that $\{Q^{(i)}(y, a)\}$ is a nonincreasing sequence, which is bounded below by $Q^*(y, a)$. Since a bounded monotone sequence always has a limit, denote $Q^{(\infty)}(y, a) \triangleq \lim_{i\to\infty} Q^{(i)}(y, a)$ and $u^{(\infty)}(y) \triangleq \lim_{i\to\infty} u^{(i)}(y)$. Taking limits on (16) and (17) yields

$$\begin{aligned}
Q^{(\infty)}(y(k), a) &= \mathcal{R}(y(k), a) + \gamma\, Q^{(\infty)}(y(k+1), u^{(\infty)})\\
&= \mathcal{R}(y(k), a) + \gamma\, V^{(\infty)}(y(k+1)) \qquad (23)\\
u^{(\infty)}(y) &= \arg\min_a Q^{(\infty)}(y, a). \qquad (24)
\end{aligned}$$

Letting $V^{(\infty)}(y)$ be the cost function of the control policy $u^{(\infty)}(y)$, from (23) and (24), we get

$$\begin{aligned}
V^{(\infty)}(y(k)) &= Q^{(\infty)}(y(k), u^{(\infty)})\\
&= \min_a\{\mathcal{R}(y(k), a) + \gamma\, Q^{(\infty)}(y(k+1), u^{(\infty)})\}\\
&= \min_a\{\mathcal{R}(y(k), a) + \gamma\, V^{(\infty)}(y(k+1))\}. \qquad (25)
\end{aligned}$$

It is observed that (25) is the tracking HJBE, i.e., $V^{(\infty)}(y) = V^*(y)$. Thus, it follows from (23) that:

$$\begin{aligned}
Q^{(\infty)}(y(k), a) &= \mathcal{R}(y(k), a) + \gamma\, V^*(x_{k+1})\\
&= Q^*(y(k), a).
\end{aligned}$$

Then, $u^{(\infty)}(y) = u^*(y)$ based on (24). The proof is complete. □

IV. Adaptive Tracking Control With Critic-Only Q-Learning

In this section, a CoQL method is developed for adaptive tracking control design based on Algorithm 1. Critic-actor and critic-only are two important structures of RL. In the critic-only structure, only critic NN is required to approximate the value function. In the critic-actor structure, both critic NN and action NN are required to approximate the value function and control policy, respectively.

A. Critic-Only Q-Learning

It is known that NNs are universal approximators [76], [77] for approximating continuous function. To solve (16), a critic NN is employed for estimating the unknown Q-function $Q^{(i)}(y, a)$ on $\mathcal{D}$. Then, the Q-function $Q^{(i)}(y, a)$ can be given by

$$Q^{(i)}(y, a) = \sum_{j=1}^{L} \theta_j^{(i)} \psi_j(y, a) + e^{(i)}(y, a) \qquad (26)$$

where $\theta^{(i)} \triangleq [\theta_1^{(i)}, \ldots, \theta_L^{(i)}]^{\mathsf{T}}$ is the ideal constant NN weight vector, $\Psi_L(x, a) \triangleq [\psi_1(y, a), \ldots, \psi_L(y, a)]^{\mathsf{T}}$ is the critic NN activation function vector, and $e^{(i)}(y, a)$ is the NN estimation error that satisfies $\lim_{L\to\infty} e^{(i)}(y, a) = 0$. Although $\theta^{(i)}$ provides the best approximation for the Q-function $Q^{(i)}(y, a)$, it is usually unknown and difficult to obtain. For real applications, the output of the critic NN is

$$\hat{Q}^{(i)}(y, a) = \sum_{j=1}^{L} \hat{\theta}_j^{(i)} \psi_j(y, a) = \Psi_L^{\mathsf{T}}(y, a)\hat{\theta}^{(i)} \qquad (27)$$

where $\hat{\theta}^{(i)} \triangleq [\hat{\theta}_1^{(i)}, \ldots, \hat{\theta}_L^{(i)}]^{\mathsf{T}}$ is the estimation of the ideal constant weight vector $\theta^{(i)}$. With $\hat{Q}^{(i)}(y, a)$, it follows from (17):

$$\hat{u}^{(i+1)}(y) = \arg\min_a \hat{Q}^{(i)}(y, a). \qquad (28)$$

For $\forall y \in \mathcal{Y}$, based on the gradient descent method, we have

$$\begin{aligned}
\hat{u}^{(i+1)}(y) &= \hat{u}^{(i)}(y) - \alpha \left.\frac{\partial \hat{Q}^{(i)}(y, a)}{\partial a}\right|_{a=\hat{u}^{(i)}(y)}\\
&= \hat{u}^{(i)}(y) - \alpha \left.\frac{\partial \Psi_L^{\mathsf{T}}(y, a)}{\partial a}\right|_{a=\hat{u}^{(i)}(y)} \hat{\theta}^{(i)} \qquad (29)
\end{aligned}$$

where $\alpha > 0$.

Remark 3: The selection of NN activation function $\Psi_L(x, a)$ is important to achieve a good approximation performance for the Q-function. However, it still remains a difficult issue in the NN function approximation and few efficient methods have been reported. For different systems, the choices of $\Psi_L(x,a)$ are often different, and thus it is difficult to develop a general method for all systems. In a word, for a specific system, the engineers' prior experiences would be helpful for the choice of $\Psi_L(x,a)$. □

To compute $\hat{\theta}^{(i)}$ for $\hat{Q}^{(i)}(y,a)$, a least-square scheme is developed using real system data. For notation simplicity, denote $(y, a, y', \mathcal{R}(y,a))$ as a data measured from the real system (4), where y' represents the next state under the control action a at state y, i.e., $y' = F(y,a)$. For real implementation of the CoQL algorithm, y' is measured from the real system without requiring the mathematical system model F. With (27) and (28), it follows from (16) that:

$$\begin{aligned}\epsilon^{(i)}(y,a) &= \hat{Q}^{(i)}(y,a) - \gamma\,\hat{Q}^{(i)}(y',\hat{u}^{(i)}) - \mathcal{R}(y,a)\\ &= [\Psi_L(y,a) - \gamma\,\Psi_L(y',\hat{u}^{(i)})]^\mathsf{T}\hat{\theta}^{(i)} - \mathcal{R}(y,a)\end{aligned} \tag{30}$$

where $\epsilon^{(i)}(y,a)$ is the residual error due to the critic NN approximation error $e^{(i)}$. Based on (30), real system data is employed to compute the unknown critic NN weight vector $\hat{\theta}^{(i)}$. The system data set is denoted as $\mathcal{S}_M \triangleq \{(y_{[l]}, a_{[l]}, y'_{[l]}, \mathcal{R}_{[l]}) | (y_{[l]}, a_{[l]}) \in \mathcal{D}, l = 1, 2, \ldots, M\}$ with its size M. Before starting the CoQL algorithm, the data set $\mathcal{S}_M$ should be collected from the measurements of sensors during the operations of the real system. For each data $(y_{[l]}, a_{[l]}, y'_{[l]}, \mathcal{R}_{[l]})$ in $\mathcal{S}_M$, the residual error (30) is given by

$$\epsilon_{[l]}^{(i)} = [\Psi_L(y_{[l]}, a_{[l]}) - \gamma\,\Psi_L(y'_{[l]}, \hat{u}^{(i)}(y'_{[l]}))]^\mathsf{T}\hat{\theta}^{(i)} - \mathcal{R}_{[l]} \tag{31}$$

where $\epsilon_{[l]}^{(i)} \triangleq \epsilon^{(i)}(y_{[l]}, a_{[l]})$ and $\mathcal{R}_{[l]} \triangleq \mathcal{R}(y_{[l]}, a_{[l]})$. The critic NN weight vector $\hat{\theta}^{(i)}$ can be computed with a least-square scheme by minimizing the sum of residual errors, that is

$$\min \sum_{l=1}^{M} (\epsilon_{[l]}^{(i)})^2. \tag{32}$$

Then, the least-square scheme is given by

$$\hat{\theta}^{(i)} = [(Z^{(i)})^\mathsf{T} Z^{(i)}]^{-1}[Z^{(i)}]^\mathsf{T}\eta \tag{33}$$

where $\eta \triangleq [\mathcal{R}_{[1]} \cdots \mathcal{R}_{[M]}]^\mathsf{T}$ and $Z^{(i)} \triangleq [z_{[1]}^{(i)} \cdots z_{[M]}^{(i)}]^\mathsf{T}$, with $z_{[l]}^{(i)} \triangleq \Psi_L(y_{[l]}, a_{[l]}) - \gamma\,\Psi_L(y'_{[l]}, \hat{u}^{(i)}(y'_{[l]}))$. The least-square scheme is a general method [78] that was widely used to update NN weights by minimizing the sum of squared residuals.

By using the least-square scheme (33), the following CoQL algorithm is developed to learn the optimal Q-function.

Remark 4: By giving an initial admissible control policy $u^{(0)}(y)$, the CoQL algorithm uses the least-squares scheme (33) for updating the critic NN weight vector $\hat{\theta}^{(i)}$ iteratively until the termination condition is satisfied. Then, the convergent Q-function will be employed for further adaptive control design. Note that the implementation procedure of the CoQL algorithm is extremely simple, where only the critic NN is required.

B. Theoretical Analysis for CoQL Algorithm

In this section, with the consideration of approximation error in the critic NN, the convergence of CoQL Algorithm 2 will be analyzed in the following theorem.

Algorithm 2 Critic-Only Q-Learning

```
► Step 1: Let $\hat{u}^{(0)} = u^{(0)}(y)$ be an initial admissible control
  policy, and $i = 0$;
► Step 2: Compute critic NN weight vector $\hat{\theta}^{(i)}$ with (33).
► Step 3: If $i \geqslant 1$ and $\|\hat{\theta}^{(i)} - \hat{\theta}^{(i-1)}\| \leqslant \varepsilon$ ($\varepsilon > 0$ is a
  small parameter), stop iteration; else, $i = i + 1$, go back
  to Step 2 and continue.
```

Theorem 2: For $\forall(y,a) \in \mathcal{Y} \times \mathcal{U}$, let $\{\hat{Q}^{(i)}(y,a)\}$ be the sequence generated by the CoQL Algorithm 2. Assume that there exist constants $\bar{M} > 0$ and $\delta > 0$, such that for $\forall M \geqslant \bar{M}$

$$\frac{1}{M}\sum_{l=1}^{M} z_{[l]}^{(i)}[z_{[l]}^{(i)}]^\mathsf{T} \geqslant \delta I_M. \tag{34}$$

Then, $\lim_{i,L\to\infty} \hat{Q}^{(i)}(y,a) = Q^*(y,a)$.

Proof: Denote $\bar{Q}^{(i)}(y,a)$ as the Q-function of control $\hat{u}^{(i)}$, that is

$$\bar{Q}^{(i)}(y,a) \triangleq \mathcal{R}(y,a) + \gamma\,\bar{Q}^{(i)}(y',\hat{u}^{(i)}). \tag{35}$$

Similar to (26), $\bar{Q}^{(i)}(y,a)$ can be expressed by

$$\bar{Q}^{(i)}(y,a) = \sum_{j=1}^{L} \bar{\theta}_j^{(i)}\psi_j(y,a) + \bar{e}^{(i)}(y,a) \tag{36}$$

where $\bar{e}^{(i)}(y,a)$ is the NN estimation error. From (35) and (36)

$$\sum_{j=1}^{L} \bar{\theta}_j^{(i)}[\gamma\,\psi_j(y',\hat{u}^{(i)}) - \psi_j(y,a)] + \mathcal{R}(y,a) + \bar{\epsilon}^{(i)}(y,a) = 0. \tag{37}$$

where $\bar{\epsilon}^{(i)}(y,a) \triangleq \gamma\,\bar{e}^{(i)}(y',\hat{u}^{(i)}) - \bar{e}^{(i)}(y,a)$.

Define

$$\widetilde{\theta}_j^{(i)} \triangleq \hat{\theta}_j^{(i)} - \bar{\theta}_j^{(i)} \tag{38}$$

which can be written as a vector $\widetilde{\theta}^{(i)} \triangleq [\widetilde{\theta}_1^{(i)} \cdots \widetilde{\theta}_L^{(i)}]^\mathsf{T}$. Based on (30), (37), and (38)

$$\begin{aligned}\bar{\epsilon}^{(i)}(y,a) &= \sum_{j=1}^{L} \big(\hat{\theta}_j^{(i)} - \widetilde{\theta}_j^{(i)}\big)[\psi_j(y,a) - \gamma\,\psi_j(y',\hat{u}^{(i)})]\\ &\quad - \mathcal{R}(y,a)\\ &= [\Psi_L(y,a) - \gamma\,\Psi_L(y',\hat{u}^{(i)})]^\mathsf{T}\hat{\theta}^{(i)} - \mathcal{R}(y,a)\\ &\quad + [\Psi_L(y,a) - \gamma\,\Psi_L(y',\hat{u}^{(i)})]^\mathsf{T}\widetilde{\theta}^{(i)}\\ &= \epsilon^{(i)}(y,a) + [\Psi_L(y,a) - \gamma\,\Psi_L(y',\hat{u}^{(i)})]^\mathsf{T}\widetilde{\theta}^{(i)}.\end{aligned} \tag{39}$$

For each data $(y_{[l]}, a_{[l]}, y'_{[l]}, \mathcal{R}_{[l]})$ in $\mathcal{S}_M$, it follows from (39) that:

$$\bar{\epsilon}_{[l]}^{(i)} = \epsilon_{[l]}^{(i)} + [z_{[l]}^{(i)}]^\mathsf{T}\widetilde{\theta}^{(i)} \tag{40}$$

where $\bar{\epsilon}_{[l]}^{(i)} \triangleq \bar{\epsilon}^{(i)}(y_{[l]}, a_{[l]})$. Then

$$[\widetilde{\theta}^{(i)}]^\mathsf{T} z_{[l]}^{(i)}[z_{[l]}^{(i)}]^\mathsf{T}\widetilde{\theta}^{(i)} = \big(\bar{\epsilon}_{[l]}^{(i)} - \epsilon_{[l]}^{(i)}\big)^2. \tag{41}$$

Using (34)

$$\sum_{l=1}^{M}[\widetilde{\theta}^{(i)}]^{\mathsf{T}} z_{[l]}^{(i)}\big[z_{[l]}^{(i)}\big]^{\mathsf{T}}\widetilde{\theta}^{(i)} \geqslant \delta M\|\widetilde{\theta}^{(i)}\|^2. \tag{42}$$

Based on (41) and (42)

$$\|\widetilde{\theta}^{(i)}\|^2 \leqslant \frac{1}{\delta M}\sum_{l=1}^{M}\big(\bar{\epsilon}_{[l]}^{(i)} - \epsilon_{[l]}^{(i)}\big)^2. \tag{43}$$

Note that the critic NN weight vector $\hat{\theta}^{(i)}$ is computed with least-square scheme (33) by minimizing (32). Then

$$\sum_{l=1}^{M}\big(\epsilon_{[l]}^{(i)}\big)^2 \leqslant \sum_{l=1}^{M}\big(\bar{\epsilon}_{[l]}^{(i)}\big)^2. \tag{44}$$

It follows from (43) and (44) that:

$$\begin{aligned}\|\widetilde{\theta}^{(i)}\|^2 &\leqslant \frac{1}{\delta M}\sum_{l=1}^{M}\big(\bar{\epsilon}_{[l]}^{(i)} - \epsilon_{[l]}^{(i)}\big)^2\\ &\leqslant \frac{1}{\delta M}\sum_{l=1}^{M}4\big(\varepsilon_{max}^{(i)}\big)^2\\ &= \frac{4}{\delta}\big(\varepsilon_{max}^{(i)}\big)^2\end{aligned} \tag{45}$$

where $\varepsilon_{max}^{(i)} \triangleq \max_l |\bar{\epsilon}_{[l]}^{(i)}|$. From the definition of $\bar{\epsilon}^{(i)}$, we have $\lim_{L\to\infty}\varepsilon_{max}^{(i)} = 0$. According to (45)

$$\lim_{L\to\infty}\|\widetilde{\theta}^{(i)}\| = 0. \tag{46}$$

Thus, it follows from (27), (36), and (46):

$$\begin{aligned}&\hat{Q}^{(i)}(y,a) - \bar{Q}^{(i)}(y,a)\\ &\quad= \sum_{j=1}^{L}\big(\hat{\theta}_j^{(i)} - \bar{\theta}_j^{(i)}\big)\psi_j(y,a) - \sum_{j=L+1}^{\infty}\bar{\theta}_j^{(i)}\psi_j(y,a)\\ &\quad= \Psi_L^{\mathsf{T}}(y,a)\widetilde{\theta}^{(i)} - \sum_{j=L+1}^{\infty}\bar{\theta}_j^{(i)}\psi_j(y,a).\end{aligned} \tag{47}$$

From (46) and (47)

$$\lim_{L\to\infty}\hat{Q}^{(i)}(y,a) - \bar{Q}^{(i)}(y,a) = 0$$

that is

$$\lim_{L\to\infty}\hat{Q}^{(i)}(y,a) = \bar{Q}^{(i)}(y,a). \tag{48}$$

Next, we will use the method of mathematical induction to prove that $\lim_{L\to\infty}\bar{Q}^{(i)}(y,a) = Q^{(i)}(y,a)$ for $\forall i$.

1) From Algorithm 2, $\hat{u}^{(0)} = u^{(0)}$. Thus, it follows from (16) and (35) that $\bar{Q}^{(0)}(y,a) = Q^{(0)}(y,a)$.
2) Assume that $\lim_{L\to\infty}\bar{Q}^{(i-1)}(y,a) = Q^{(i-1)}(y,a)$ for $\forall i > 0$. Then, it follows from (48) that $\lim_{L\to\infty}\hat{u}^{(i)} = u^{(i)}$. According to (35)

$$\begin{aligned}\lim_{L\to\infty}\bar{Q}^{(i)}(y,a) &= \mathcal{R}(y(k),a) + \gamma\lim_{L\to\infty}\bar{Q}^{(i)}(y',\hat{u}^{(i)})\\ &= \mathcal{R}(y,a) + \gamma\lim_{L\to\infty}V_{\hat{u}^{(i)}}(y')\\ &= \mathcal{R}(y,a) + \gamma V_{u^{(i)}}(y')\\ &= Q^{(i)}(y,a).\end{aligned} \tag{49}$$

Based on the above two steps

$$\lim_{L\to\infty}\bar{Q}^{(i)}(y,a) = Q^{(i)}(y,a) \tag{50}$$

for $\forall i$. From (48) and (50)

$$\lim_{L\to\infty}\hat{Q}^{(i)}(y,a) = Q^{(i)}(y,a). \tag{51}$$

Then, according to part 2) of Theorem 1, $\lim_{i,L\to\infty}\hat{Q}^{(i)}(y,a) = Q^{*}(y,a)$. □

Remark 5: In Theorem 2, the condition (34) is related to the issue of exploration. To realize the condition (34), it requires to increase the richness of the system data set $\mathcal{S}_M$. During practical applications, there are some potential methods to increase the richness of $\mathcal{S}_M$, such as collecting system data with different initial states, using arbitrary or even randomized exploratory behavior control action a. □

C. Adaptive Tracking Control

After the CoQL Algorithm 2 is terminated, the convergent Q-function is used for adaptive tracking control design. Denote the convergent critic NN weight vector as θ_c and the convergent Q-function as $Q_c(y,a) = \Psi_L^{\mathsf{T}}(y,a)\theta_c$. Then, according to (15), the tracking control law is given by

$$u(k) = \arg\min_a Q_c(y(k),a). \tag{52}$$

To solve the optimization problem (52) at each time instant k, the one-step gradient descent method can be employed, and then the adaptive tracking control can be given as

$$\begin{aligned}u(k) &= u(k-1) - \alpha\frac{\partial Q_c(y(k),a)}{\partial a}\bigg|_{a=u(k-1)}\\ &= u(k-1) - \alpha\frac{\partial\Psi_L^{\mathsf{T}}(y(k),a)}{\partial a}\bigg|_{a=u(k-1)}\theta_c.\end{aligned} \tag{53}$$

As stated in the Introduction, the motivation of this paper is to achieve three important goals simultaneously. Accordingly, compared with existing works, the strengths of the developed CoQL method and the contributions of this paper can be analyzed from three aspects as follows.

1) The CoQL method is developed for solving the model-free optimal tracking control problem of general nonaffine nonlinear systems. It is known that results reported for solving this problem are scarce. Note that most of the existing works on optimal tracking control were reported for affine nonlinear systems [43], [44], [51], [53]–[55] or linear systems [42], [45], [48]. Moreover, these results are completely model based [44], [50], [53] or partially model based [43], [45], [51], [55].
2) The CoQL method learns the tracking control policy with off-policy data. Off-policy data is arbitrary data collected from the daily operations of the real system. Note that most of the existing related works [12], [24], [42], [51], [66], [68], [79], [80] belong to the on-policy learning framework or use on-policy implementation. For the on-policy learning scheme, to evaluate the value function of a target control policy u, the system data should be generated using the target control policy u. This means that during the learning process, the

learned control policy should be applied to generate data before its convergence, which makes the data collection much more difficult. That is to say, on-policy methods cannot learn from the off-policy data. Even worse, learning with on-policy scheme has the problem of inadequate exploration [19]. Unlike the on-policy methods, off-policy methods are able to evaluate a target policy while executing other behavior policies, which means that any off-policy data is useful during the learning process.

3) The developed CoQL method uses the critic-only implementation structure. For the model-free tracking control problem of general nonaffine nonlinear systems, the use of the critic-only structure is still a difficult issue and no results have been reported according to the best of our knowledge. Compared with the actor-critic structure, the critic-only implementation structure may reduce the computational effort but at the price of losing the accuracy to some extent.

Remark 6: It is worth pointing out that the developed CoQL method suits for solving extensive general model-free optimal tracking control problems. The term *general* can be reflected from the following aspects.

1) The system (1) is *general* and the full system model $f(x, u)$ is not required. Nearly all time-invariant systems can be described by (1). Many existing works are mainly restricted to linear systems [42], [45], [48], [65], [66], [68] or affine nonlinear systems [43], [44], [51], [53]–[55], which are special forms of the system (1).
2) The desired trajectory $r(k)$ is a *general* bounded signal since $h(r)$ is unknown.
3) $\mathcal{R}(y, u)$ in the performance index (6) is *general*, where its explicit expression is not required and thus no restrictions are imposed on its form. The one-step cost $\mathcal{R}(y, u)$ is the real instantaneous cost generated at state y with control action u during the practical operation. For many existing works, $\mathcal{R}(y, u)$ is usually restricted to quadratic form [42], [44], [45], [48], [55], [66], [68] or at least quadratic form in control [82].
4) The developed CoQL method is an off-policy learning approach, where the system data required for learning is *general*. For the off-policy learning approach, the target policy can be evaluated with data generated with any other behavior policies, and thus any data collected from real system is useful. □

Remark 7: Similar to the developed CoQL method, the single network adaptive critic (SNAC) [8], [16], [83] uses only one critic NN to approximate the costate, which was proved to be a promising method for control design. However, there are three main differences between the CoQL and SNAC methods.

1) The developed CoQL is a model-free method by learning with off-policy data, while the SNAC is a model-based method that requires the system mathematical equation.
2) The CoQL method is developed for solving the optimal tracking control problem in this paper, while the SNAC methods were designed for optimal regulation problems.
3) The CoQL and SNAC methods belong to different frameworks. The CoQL method is developed based on the action-state value function, i.e., Q-function Q, while the SNAC method is designed based on the state value function V. Thus, different approaches may be required to analyze the two methods. □

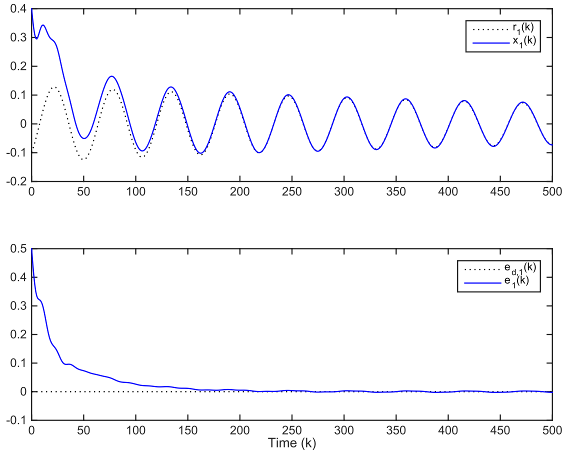

Fig. 1. Trajectories of $r_1(k)$, state $x_1(k)$, and tracking error $e_1(k)$ generated by CoQL-based adaptive tracking control.

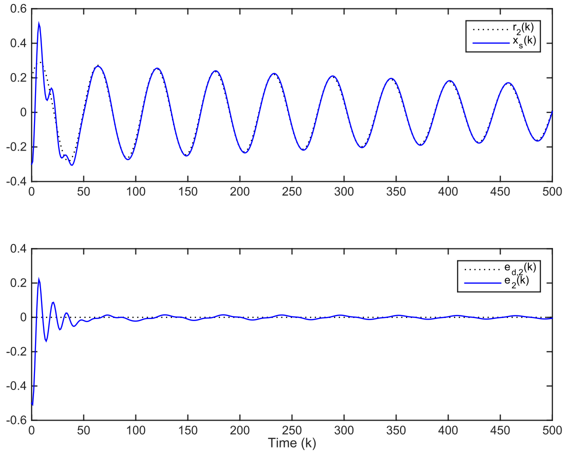

Fig. 2. Trajectories of $r_2(k)$, state $x_2(k)$, and tracking error $e_2(k)$ generated by CoQL-based adaptive tracking control.

V. Simulation Studies

In this section, the effectiveness of the developed CoQL method is verified through simulation studies on a nonlinear system. Consider the following system:

$$\begin{cases} x_1(k+1) = 0.9950\tanh(x_1(k)) + 0.0499\tanh(x_2(k)) \\ x_2(k+1) = -0.2996\tanh(x_1(k)) + 0.9925\tanh(x_2(k)) \\ \qquad\qquad\quad + \sin(u) \end{cases} \tag{54}$$

with $x(0) = [0.4, -0.3]^{\mathsf{T}}$. Let the desired trajectory $r(k)$ be generated by the following command system:

$$\begin{cases} r_1(k+1) = 0.9963r_1(k) + 0.0498r_2(k) \\ r_2(k+1) = -0.2492r_1(k) + 0.9888r_2(k) \end{cases} \tag{55}$$

with $r(0) = [-0.1, 0.2]^{\mathsf{T}}$. The desired trajectory $r(k)$ are sinusoidal signals, which are shown in Figs. 1 and 2 with

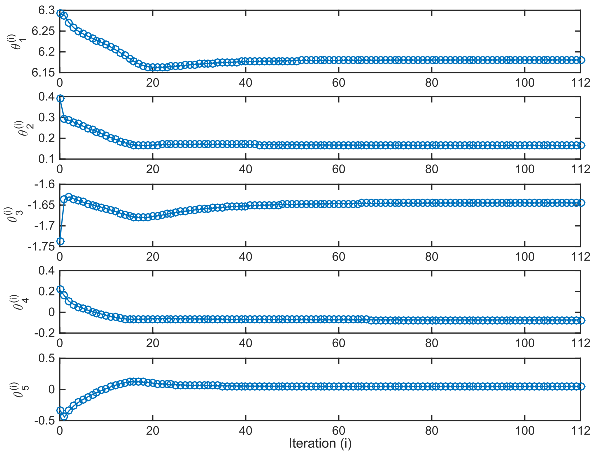

Fig. 3. Critic NN weights $\hat{\theta}_1^{(i)} - \hat{\theta}_5^{(i)}$ at each iteration.

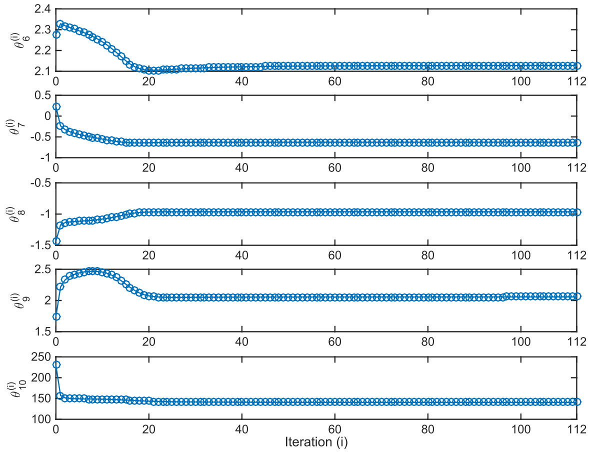

Fig. 4. Critic NN weights $\hat{\theta}_6^{(i)} - \hat{\theta}_{10}^{(i)}$ at each iteration.

black dotted lines. For the discounted performance index (6), let $\mathcal{R}(y(l), u(l)) = e_1^2(l) + 2e_2^2(l) + u^2$ and $\gamma = 0.99$. To show the real cost generated using control u, define the cost with respect to time as

$$J(k) \triangleq \sum_{l=0}^{k} \gamma^l \mathcal{R}(y(l), u(l)). \tag{56}$$

To learn the Q-function with the developed CoQL method (i.e., Algorithm 2), let the initial control $u^{(0)}(y) = [0.0906, 0.0187, 0.0936, 0.0322]y$, the termination condition $\varepsilon = 10^{-5}$, and the critic NN activation function $\Psi_L(x, a) = [\tanh^2(e_1), \tanh(e_1)\tanh(e_2), \tanh(e_1)r_1, \tanh(e_1)r_2, \tanh(e_1)\sin(a), \tanh^2(e_2), \tanh(e_2)r_1, \tanh(e_2)r_2, \tanh(e_2)\sin(a), r_1^2, r_1r_2, r_1\sin(a), r_2^2, r_2\sin(a), \sin^2(a)]^{\mathsf{T}}$. Using these parameters, simulation is conducted with the CoQL algorithm and it terminates at the 112th iterations. Figs. 3–5 show the critic NN weights at each iteration, where $\hat{\theta}^{(i)}$ converges to $\theta_c = [6.1806, 0.1657, -1.6448, -0.0743, 0.0481, 2.1282, -0.6271, -0.9628, 2.0581, 140.4417, 2.5552, -0.2362, 26.7717, -0.1004, 0.5357]^{\mathsf{T}}$. With the convergent critic NN weight vector θ_c, the CoQL-based adaptive control law (53) is then employed for the system (54). The simulation results are given in Figs. 1, 2, 6, and 7, where the dotted lines denote the desired trajectories and the solid lines are generated by the developed CoQL-based adaptive tracking control. Figs. 1 and 2 demonstrate the trajectories of system state $x(k)$ and tracking error $e(k)$. Note that the CoQL-based adaptive tracking control achieves a good tracking performance. The trajectory of the CoQL-based adaptive control

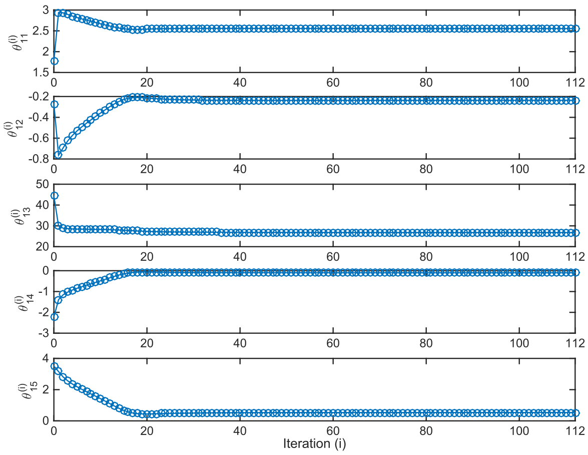

Fig. 5. Critic NN weights $\hat{\theta}_{11}^{(i)} - \hat{\theta}_{15}^{(i)}$ at each iteration.

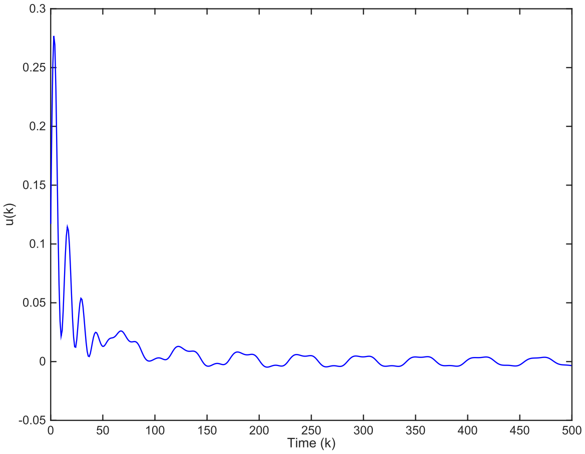

Fig. 6. Trajectory of the CoQL-based adaptive tracking control.

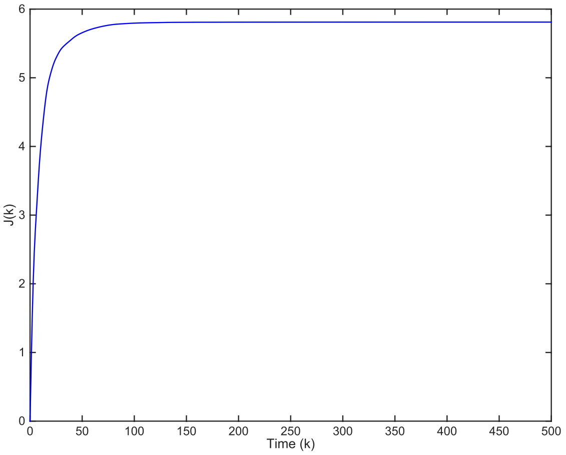

Fig. 7. Trajectory of $J(k)$ generated by the CoQL-based adaptive tracking control.

is shown in Fig. 6. In Fig. 7, the real cost $J(k)$ of the CoQL-based adaptive tracking control converges to 5.8109.

Remark 8: For the developed CoQL method, most of its execution time is consumed on learning the Q-function by computing the critic NN weight vector with (33) iteratively. It is noted from (33) that the time will increase as the dimension of the matrix $Z^{(i)}$ increases. Note that the dimension of the matrix $Z^{(i)}$ is determined by the parameters L and M, where L is the size of the critic NN and M is the size of the data set. The parameters L and M are usually affected by the dimension and the complexity of the system. That is to say, for complex or higher dimensional systems, more hidden-layer nodes (i.e., larger L) of the critic NN are required to approximate the Q-function and more system data (i.e., larger M) is needed to learn the Q-function. Therefore, the dimension and the complexity of the system are the two main factors that affect the execution time of the developed CoQL method. □

VI. Conclusion

The CoQL method was developed for model-free optimal tracking control design of general nonlinear discrete-time systems. The optimal tracking control problem was first reformulated as an optimal regulation control problem of the argumented system, and then the Q-learning algorithm was proposed to learn the optimal Q-function without requiring system model. Q-learning generates a nonincreasing Q-function sequence, which was proved converging to the optimal Q-function. For implementation purposes, the CoQL method was developed, where only critic NN was required for approximating the Q-function. By involving NN estimation error, the convergence of the CoQL method is proved. After the converge Q-function is obtained from the CoQL method, the adaptive tracking control was designed based on a gradient descent scheme. The developed CoQL method is general and simple to implement, which suits for solving the model-free optimal tracking control problem of many practical systems. To verify the effectiveness of the developed adaptive optimal tracking control method, simulation studies were conducted on a nonlinear system and good tracking performance was achieved.

There are some related interesting but not trivial issues worth further investigation, but beyond the scope of this paper. These are as follows.

1) The selection of parameters in the developed CoQL method are still experience based. Thus, it is required to conduct further studies on the parameters selection such that the performance of the CoQL method can be improved.
2) Extend the developed methods to handle control problems with input constraints, delays, uncertainties, etc.
3) Apply the developed methods to solve real application problems.

References

[1] D. P. Bertsekas and J. N. Tsitsiklis, *Neuro-Dynamic Programming*. Belmont, MA, USA: Athena Scientific, 1996.

[2] L. P. Kaelbling, M. L. Littman, and A. W. Moore, "Reinforcement learning: A survey," *J. Artif. Intell. Res.*, vol. 4, no. 1, pp. 237–285, Jan. 1996.

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

[4] D. P. Bertsekas, "Approximate policy iteration: A survey and some new methods," *J. Control Theory Appl.*, vol. 9, no. 3, pp. 310–335, Aug. 2011.

[5] D. P. Bertsekas, "Temporal difference methods for general projected equations," *IEEE Trans. Autom. Control*, vol. 56, no. 9, pp. 2128–2139, Sep. 2011.

[6] G. Chowdhary, M. Liu, R. Grande, T. Walsh, J. How, and L. Carin, "Off-policy reinforcement learning with Gaussian processes," *IEEE/CAA J. Autom. Sinica*, vol. 1, no. 3, pp. 227–238, Jul. 2014.

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

[8] R. Padhi, N. Unnikrishnan, X. Wang, and S. N. Balakrishnan, "A single network adaptive critic (SNAC) architecture for optimal control synthesis for a class of nonlinear systems," *Neural Netw.*, vol. 19, no. 10, pp. 1648–1660, 2006.

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

[10] H.-N. Wu and B. Luo, "Neural network based online simultaneous policy update algorithm for solving the HJI equation in nonlinear H_∞ control," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 23, no. 12, pp. 1884–1895, Dec. 2012.

[11] T. Dierks and S. Jagannathan, "Online optimal control of affine nonlinear discrete-time systems with unknown internal dynamics by using time-based policy update," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 23, no. 7, pp. 1118–1129, Jul. 2012.

[12] D. Wang, D. Liu, Q. Wei, D. Zhao, and N. Jin, "Optimal control of unknown nonaffine nonlinear discrete-time systems based on adaptive dynamic programming," *Automatica*, vol. 48, no. 8, pp. 1825–1832, 2012.

[13] B. Luo and H. N. Wu, "Approximate optimal control design for nonlinear one-dimensional parabolic PDE systems using empirical eigenfunctions and neural network," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 42, no. 6, pp. 1538–1549, Dec. 2012.

[14] Z. Ni, H. He, and J. Wen, "Adaptive learning in tracking control based on the dual critic network design," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 24, no. 6, pp. 913–928, Jun. 2013.

[15] F. L. Lewis and D. Liu, Eds., *Reinforcement Learning and Approximate Dynamic Programming for Feedback Control*, vol. 17. Hoboken, NJ, USA: Wiley, 2013.

[16] A. Heydari and S. N. Balakrishnan, "Finite-horizon control-constrained nonlinear optimal control using single network adaptive critics," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 24, no. 1, pp. 147–157, Jan. 2013.

[17] X. Xu, Z. Hou, C. Lian, and H. He, "Online learning control using adaptive critic designs with sparse kernel machines," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 24, no. 5, pp. 762–775, May 2013.

[18] Y. Jiang and Z.-P. Jiang, "Robust adaptive dynamic programming and feedback stabilization of nonlinear systems," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 25, no. 5, pp. 882–893, May 2014.

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

[20] R. Kamalapurkar, J. R. Klotz, and W. E. Dixon, "Concurrent learning-based approximate feedback-Nash equilibrium solution of N-player nonzero-sum differential games," *IEEE/CAA J. Autom. Sinica*, vol. 1, no. 3, pp. 239–247, Jul. 2014.

[21] A. Heydari and S. N. Balakrishnan, "Optimal switching and control of nonlinear switching systems using approximate dynamic programming," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 25, no. 6, pp. 1106–1117, Jun. 2014.

[22] J. Y. Lee, J. B. Park, and Y. H. Choi, "Integral reinforcement learning for continuous-time input-affine nonlinear systems with simultaneous invariant explorations," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 5, pp. 916–932, May 2015.

[23] B. Xu, C. Yang, and Z. Shi, "Reinforcement learning output feedback NN control using deterministic learning technique," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 25, no. 3, pp. 635–641, Mar. 2014.

[24] X. Zhong, H. He, H. Zhang, and Z. Wang, "Optimal control for unknown discrete-time nonlinear Markov jump systems using adaptive dynamic programming," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 25, no. 12, pp. 2141–2155, Dec. 2015.

[25] Z. Ni, H. He, X. Zhong, and D. V. Prokhorov, "Model-free dual heuristic dynamic programming," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 8, pp. 1834–1839, Aug. 2013.

[26] B. Luo, H.-N. Wu, and H.-X. Li, "Data-based suboptimal neuro-control design with reinforcement learning for dissipative spatially distributed processes," *Ind. Eng. Chem. Res.*, vol. 53, no. 19, pp. 8106–8119, 2014.

[27] B. Luo, H.-N. Wu, and H.-X. Li, "Adaptive optimal control of highly dissipative nonlinear spatially distributed processes with neuro-dynamic programming," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 4, pp. 684–696, Apr. 2015.

[28] Q. Wei and D. Liu, "Numerical adaptive learning control scheme for discrete-time non-linear systems," *IET Control Theory Appl.*, vol. 7, no. 11, pp. 1472–1486, Jul. 2013.

[29] Q. Wei and D. Liu, "Adaptive dynamic programming for optimal tracking control of unknown nonlinear systems with application to coal gasification," *IEEE Trans. Autom. Sci. Eng.*, vol. 11, no. 4, pp. 1020–1036, Oct. 2014.

[30] Q. Wei and D. Liu, "Data-driven neuro-optimal temperature control of water–gas shift reaction using stable iterative adaptive dynamic programming," *IEEE Trans. Ind. Electron.*, vol. 61, no. 11, pp. 6399–6408, Nov. 2014.

[31] Q. Wei, F.-Y. Wang, D. Liu, and X. Yang, "Finite-approximation-error-based discrete-time iterative adaptive dynamic programming," *IEEE Trans. Cybern.*, vol. 44, no. 12, pp. 2820–2833, Dec. 2014.

[32] Q. Wei and D. Liu, "A novel iterative θ-adaptive dynamic programming for discrete-time nonlinear systems," *IEEE Trans. Autom. Sci. Eng.*, vol. 11, no. 4, pp. 1176–1190, Oct. 2014.

[33] D. Liu, H. Li, and D. Wang, "Neural-network-based zero-sum game for discrete-time nonlinear systems via iterative adaptive dynamic programming algorithm," *Neurocomputing*, vol. 110, no. 13, pp. 92–100, 2013.

[34] Y. Tang, H. He, J. Wen, and J. Liu, "Power system stability control for a wind farm based on adaptive dynamic programming," *IEEE Trans. Smart Grid*, vol. 6, no. 1, pp. 166–177, Jan. 2015.

[35] Z. D. Wilcox, W. MacKunis, S. Bhat, R. Lind, and W. E. Dixon, "Lyapunov-based exponential tracking control of a hypersonic aircraft with aerothermoelastic effects," *J. Guid., Control, Dyn.*, vol. 33, no. 4, pp. 1213–1224, 2010.

[36] B. Xu, C. Yang, and Y. Pan, "Global neural dynamic surface tracking control of strict-feedback systems with application to hypersonic flight vehicle," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 10, pp. 2563–2575, Oct. 2015.

[37] B. Xu and Z. Shi, "An overview on flight dynamics and control approaches for hypersonic vehicles," *Sci. China Inf. Sci.*, vol. 58, no. 7, pp. 1–19, Jul. 2015.

[38] K. Lu and Y. Xia, "Adaptive attitude tracking control for rigid spacecraft with finite-time convergence," *Automatica*, vol. 49, no. 12, pp. 3591–3599, 2013.

[39] W. Luo, Y.-C. Chu, and K.-V. Ling, "Inverse optimal adaptive control for attitude tracking of spacecraft," *IEEE Trans. Autom. Control*, vol. 50, no. 11, pp. 1639–1654, Nov. 2005.

[40] A. Mannava, S. N. Balakrishnan, L. Tang, and R. G. Landers, "Optimal tracking control of motion systems," *IEEE Trans. Control Syst. Technol.*, vol. 20, no. 6, pp. 1548–1558, Nov. 2012.

[41] B. Wang, Z.-H. Guan, and F.-S. Yuan, "Optimal tracking and two-channel disturbance rejection under control energy constraint," *Automatica*, vol. 47, no. 4, pp. 733–738, Apr. 2011.

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

[43] H. Modares and F. L. Lewis, "Optimal tracking control of nonlinear partially-unknown constrained-input systems using integral reinforcement learning," *Automatica*, vol. 50, no. 7, pp. 1780–1792, Jul. 2014.

[44] R. Kamalapurkar, H. Dinh, S. Bhasin, and W. E. Dixon, "Approximate optimal trajectory tracking for continuous-time nonlinear systems," *Automatica*, vol. 51, no. 1, pp. 40–48, Jan. 2015.

[45] H. Modares and F. L. Lewis, "Linear quadratic tracking control of partially-unknown continuous-time systems using reinforcement learning," *IEEE Trans. Autom. Control*, vol. 59, no. 11, pp. 3051–3056, Nov. 2014.

[46] J. Na and G. Herrmann, "Online adaptive approximate optimal tracking control with simplified dual approximation structure for continuous-time unknown nonlinear systems," *IEEE/CAA J. Autom. Sinica*, vol. 1, no. 4, pp. 412–422, Oct. 2014.

[47] Q. Wei and D. Liu, "A novel iterative-adaptive dynamic programming for discrete-time non-linear systems," *IEEE Trans. Autom. Sci. Eng.*, vol. 11, no. 4, pp. 1176–1190, Oct. 2014.

[48] B. Kiumarsi, F. L. Lewis, M.-B. Naghibi-Sistani, and A. Karimpour, "Optimal tracking control of unknown discrete-time linear systems using input-output measured data," *IEEE Trans. Cybern.*, vol. 45, no. 12, pp. 2770–2779, Dec. 2015.

[49] H. Zhang, L. Cui, X. Zhang, and Y. Luo, "Data-driven robust approximate optimal tracking control for unknown general nonlinear systems using adaptive dynamic programming method," *IEEE Trans. Neural Netw.*, vol. 22, no. 12, pp. 2226–2236, Dec. 2011.

[50] H. Zhang, R. Song, Q. Wei, and T. Zhang, "Optimal tracking control for a class of nonlinear discrete-time systems with time delays based on heuristic dynamic programming," *IEEE Trans. Neural Netw.*, vol. 22, no. 12, pp. 1851–1862, Dec. 2011.

[51] B. Kiumarsi and F. L. Lewis, "Actor–critic-based optimal tracking for partially unknown nonlinear discrete-time systems," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 1, pp. 140–151, Jan. 2015.

[52] Y.-J. Liu, L. Tang, S.-C. Tong, C. L. P. Chen, and D.-J. Li, "Reinforcement learning design-based adaptive tracking control with less learning parameters for nonlinear discrete-time MIMO systems," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 1, pp. 165–176, Jan. 2015.

[53] H. Zhang, Q. Wei, and Y. Luo, "A novel infinite-time optimal tracking control scheme for a class of discrete-time nonlinear systems via the greedy HDP iteration algorithm," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 38, no. 4, pp. 937–942, Aug. 2008.

[54] Q. Wei and D. Liu, "Neural-network-based adaptive optimal tracking control scheme for discrete-time nonlinear systems with approximation errors," *Neurocomputing*, vol. 149, no. 3, pp. 106–115, Feb. 2015.

[55] B. Kiumarsi, F. L. Lewis, and D. S. Levine, "Optimal control of nonlinear discrete time-varying systems using a new neural network approximation structure," *Neurocomputing*, vol. 156, pp. 157–165, May 2015.

[56] C. Qin, H. Zhang, and Y. Luo, "Online optimal tracking control of continuous-time linear systems with unknown dynamics by using adaptive dynamic programming," *Int. J. Control*, vol. 87, no. 5, pp. 1000–1009, 2014.

[57] C. J. C. H. Watkins and P. Dayan, "Q-learning," *Mach. Learn.*, vol. 8, nos. 3–4, pp. 279–292, 1992.

[58] J. N. Tsitsiklis, "Asynchronous stochastic approximation and Q-learning," *Mach. Learn.*, vol. 16, no. 3, pp. 185–202, Sep. 1994.

[59] J. Peng and R. J. Williams, "Incremental multi-step Q-learning," *Mach. Learn.*, vol. 22, nos. 1–3, pp. 283–290, 1996.

[60] E. Even-Dar and Y. Mansour, "Learning rates for Q-learning," *J. Mach. Learn. Res.*, vol. 5, pp. 1–25, Jan. 2003.

[61] S. Bhatnagar and K. M. Babu, "New algorithms of the Q-learning type," *Automatica*, vol. 44, no. 4, pp. 1111–1119, Apr. 2008.

[62] H. R. Maei and R. S. Sutton, "GQ(λ): A general gradient algorithm for temporal-difference prediction learning with eligibility traces," in *Proc. 3rd Conf. Artif. General Intell.*, vol. 1. 2010, pp. 91–96.

[63] S. Doltsinis, P. Ferreira, and N. Lohse, "An MDP model-based reinforcement learning approach for production station ramp-up optimization: Q-learning analysis," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 44, no. 9, pp. 1125–1138, Sep. 2014.

[64] 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, 2007.

[65] 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.

[66] 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.

[67] Q. Wei, D. Liu, and G. Shi, "A novel dual iterative Q-learning method for optimal battery management in smart residential environments," *IEEE Trans. Ind. Electron.*, vol. 62, no. 4, pp. 2509–2518, Apr. 2015.

[68] M. Palanisamy, H. Modares, F. L. Lewis, and M. Aurangzeb, "Continuous-time Q-learning for infinite-horizon discounted cost linear quadratic regulator problems," *IEEE Trans. Cybern.*, vol. 45, no. 2, pp. 165–176, Feb. 2015.

[69] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. Cambridge, MA, USA: MIT Press, 1998.

[70] D. Precup, R. S. Sutton, and S. Dasgupta, "Off-policy temporal-difference learning with function approximation," in *Proc. 18th Int. Conf. Mach. Learn.*, 2001, pp. 417–424.

[71] H. R. Maei, C. Szepesvári, S. Bhatnagar, and R. S. Sutton, "Toward off-policy learning control with function approximation," in *Proc. 27th Int. Conf. Mach. Learn.*, 2010, pp. 719–726.

[72] B. Luo, H.-N. Wu, and T. Huang, "Off-policy reinforcement learning for H_∞ control design," *IEEE Trans. Cybern.*, vol. 45, no. 1, pp. 65–76, Jan. 2015.

[73] H. Modares, F. L. Lewis, and Z.-P. Jiang, "H_∞ tracking control of completely unknown continuous-time systems via off-policy reinforcement learning," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 10, pp. 2550–2562, Oct. 2015.

[74] B. Luo, H.-N. Wu, T. Huang, and D. Liu, "Reinforcement learning solution for HJB equation arising in constrained optimal control problem," *Neural Netw.*, vol. 71, pp. 150–158, Nov. 2015.

[75] B. Luo, T. Huang, H. Wu, and X. Yang, "Data-driven H_∞ control for nonlinear distributed parameter systems," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 11, pp. 2949–2961, Nov. 2015.

[76] J. T. Spooner, M. Maggiore, R. Ordóñez, and K. M. Passino, *Stable Adaptive Control and Estimation for Nonlinear Systems: Neural and Fuzzy Approximator Techniques*, vol. 43. New York, NY, USA: Wiley, 2004.

[77] K. Hornik, M. Stinchcombe, and H. White, "Universal approximation of an unknown mapping and its derivatives using multilayer feedforward networks," *Neural Netw.*, vol. 3, no. 5, pp. 551–560, 1990.

[78] S. O. Haykin, *Neural Networks and Learning Machines*, vol. 3. Upper Saddle River, NJ, USA: Pearson Education, 2009.

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

[80] 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.

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

[82] Q. Zhao, H. Xu, and S. Jagannathan, "Neural network-based finite-horizon optimal control of uncertain affine nonlinear discrete-time systems," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 3, pp. 486–499, Mar. 2015.

[83] J. Ding, A. Heydari, and S. N. Balakrishnan, "Single network adaptive critics networks—Development, analysis and applications," in *Proc. Reinforcement Learn. Approx. Dyn. Program. Feedback Control*, 2013, pp. 98–118.

Biao Luo (M'15) received the B.E. and M.E. degrees from Xiangtan University, Xiangtan, China, in 2006 and 2009, respectively, and the Ph.D. degree from Beihang University, Beijing, China, in 2014.

He was a Research Assistant with the Department of System Engineering and Engineering Management, City University of Hong Kong, Hong Kong, in 2013. He was a Research Assistant/Associate with the Department of Mathematics and Science, Texas A&M University at Qatar, Doha, Qatar, in 2013, 2014, and 2015. He is currently an Assistant Professor with the Institute of Automation, Chinese Academy of Sciences, Beijing. His current research interests include distributed parameter systems, optimal control, data-based control, fuzzy/neural modeling and control, hypersonic entry/reentry guidance, learning and control from big data, reinforcement learning, approximate dynamic programming, and evolutionary computation.

Dr. Luo was a recipient of the Chinese Association of Automation Outstanding Ph.D. Dissertation Award in 2015. He serves as an Associate Editor of the *Artificial Intelligence Review*. He was the Secretariat of the 12th World Congress on Intelligent Control and Automation 2016, the Secretariat of the 5th International Conference on Information Science and Technology 2015, and the Publication Chair of the 13th International Symposium on Neural Networks 2016.

Derong Liu (S'91–M'94–SM'96–F'05) received the Ph.D. degree in electrical engineering from the University of Notre Dame, Notre Dame, IN, USA, in 1994.

He was a Staff Fellow with the General Motors Research and Development Center, Warren, MI, USA, from 1993 to 1995. He was an Assistant Professor with the Department of Electrical and Computer Engineering, Stevens Institute of Technology, Hoboken, NJ, USA, from 1995 to 1999. He joined the University of Illinois at Chicago, Chicago, IL, USA, in 1999, and became a Full Professor of Electrical and Computer Engineering and of Computer Science in 2006. He was selected for the 100 Talents Program by the Chinese Academy of Sciences in 2008. He served as the Associate Director of the State Key Laboratory of Management and Control for Complex Systems with the Institute of Automation, Chinese Academy of Sciences, Beijing, from 2010 to 2015. He is currently a Full Professor with the School of Automation and Electrical Engineering, University of Science and Technology Beijing, Beijing. He has authored 15 books (six research monographs and nine edited volumes).

Prof. Liu is an elected AdCom Member of the IEEE Computational Intelligence Society. He is a fellow of the International Neural Network Society. He received the Faculty Early Career Development Award from the National Science Foundation in 1999, the University Scholar Award from the University of Illinois from 2006 to 2009, the Overseas Outstanding Young Scholar Award from the National Natural Science Foundation of China in 2008, and the Outstanding Achievement Award from the Asia Pacific Neural Network Assembly in 2014. He was the Editor-in-Chief of the IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS. He was the General Chair of the 2014 IEEE World Congress on Computational Intelligence and is the General Chair of the 2016 World Congress on Intelligent Control and Automation.

Tingwen Huang received the B.S. degree from Southwest University, Chongqing, China, in 1990, the M.S. degree from Sichuan University, Chengdu, China, in 1993, and the Ph.D. degree from Texas A& M University, College Station, TX, USA, in 2002.

He was a Visiting Assistant Professor with Texas A&M University, after graduation. Then, he joined Texas A& M University at Qatar, Doha, Qatar, as an Assistant Professor in 2003, where he is currently a Professor. He has authored or co-authored over 100 refereed journal papers. His current research interests include neural networks, chaotic dynamical systems, complex networks, and optimization and control.

Ding Wang (M'15) received the B.S. degree in mathematics from the Zhengzhou University of Light Industry, Zhengzhou, China, in 2007, the M.S. degree in operations research and cybernetics from Northeastern University, Shenyang, China, in 2009, and the Ph.D. degree in control theory and control engineering from the Institute of Automation, Chinese Academy of Sciences, Beijing, China, in 2012.

He has been a Visiting Scholar with the Department of Electrical, Computer, and Biomedical Engineering, University of Rhode Island, Kingston, RI, USA, since 2015. He is currently an Associate Professor with the State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences. He has authored over 60 journal and conference papers, and co-authored two monographs. His current research interests include adaptive and learning systems, intelligent control, and neural networks.

Dr. Wang is a member of the Asia-Pacific Neural Network Society and the Chinese Association of Automation (CAA). He was a recipient of the Excellent Doctoral Dissertation Award of the Chinese Academy of Sciences in 2013, and a nomination of the Excellent Doctoral Dissertation Award of CAA in 2014. He was the Registration Chair of the 4th International Conference on Intelligent Control and Information Processing in 2013and the 5th International Conference on Information Science and Technology in 2015, and the Secretariat of the 2014 IEEE World Congress on Computational Intelligence in 2014. He served as the Program Committee Member of several international conferences. He is the Finance Chair of the 12th World Congress on Intelligent Control and Automation in 2016. He serves as an Associate Editor of the IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS and *Neurocomputing*.