Adaptive Constrained Optimal Control Design for Data-Based Nonlinear Discrete-Time Systems With Critic-Only Structure

Biao Luo, Member, IEEE, Derong Liu, Fellow, IEEE, and Huai-Ning Wu

Abstract—Reinforcement learning has proved to be a powerful tool to solve optimal control problems over the past few years. However, the data-based constrained optimal control problem of nonaffine nonlinear discrete-time systems has rarely been studied yet. To solve this problem, an adaptive optimal control approach is developed by using the value iteration-based Q-learning (VIQL) with the critic-only structure. Most of the existing constrained control methods require the use of a certain performance index and only suit for linear or affine nonlinear systems, which is unreasonable in practice. To overcome this problem, the system transformation is first introduced with the general performance index. Then, the constrained optimal control problem is converted to an unconstrained optimal control problem. By introducing the action-state value function, i.e., Q-function, the VIQL algorithm is proposed to learn the optimal Q-function of the data-based unconstrained optimal control problem. The convergence results of the VIQL algorithm are established with an easy-to-realize initial condition $Q^0(x, a) = 0$. To implement the VIQL algorithm, the critic-only structure is developed, where only one neural network is required to approximate the Q-function. The converged Q-function obtained from the critic-only VIQL method is employed to design the adaptive constrained optimal controller based on the gradient descent scheme. Finally, the effectiveness of the developed adaptive control method is tested on three examples with computer simulation.

Index Terms—Adaptive control, adaptive dynamic programming, constraints, critic-only, data-based, optimal control, Q-learning.

I. INTRODUCTION

Over the past few years, many research works [1]–[17] have been reported to solve the optimal control problems with reinforcement learning or approximate dynamic programming. Theoretically, the optimal control problems rely on the solutions of Hamilton–Jacobi–Bellman equation (HJBE) [18], [19] or Hamilton–Jacobi–Isaacs equation (HJIE) [20].

However, the solutions of the HJBE and the HJIE are not an easy task. Moreover, for the data-based optimal control problems considered in this paper, the problem becomes even more difficult.

Reinforcement learning is a famous method in machine learning community and has been widely studied. In recent years, the thoughts of reinforcement learning have been introduced to design approximate dynamic programming methods for solving the optimal control problems in control community [10], [11], [14], [21]–[51]. To discuss a few, the convergence of the value iteration-based heuristic dynamic programming algorithm was proved in [47] for the optimal control of affine nonlinear discrete-time systems. The optimal tracking control problem was considered in [46], where after system transformation, a greedy heuristic dynamic programming iteration algorithm was suggested. Based on online approximators, the state feedback and output feedback adaptive critic controllers were designed in [43]. By using the measured input/output data, value iteration and policy iteration were proposed for the output feedback control of linear discrete-time systems [45]. Without the internal dynamics of the affine nonlinear discrete-time systems, Dierks and Jagannathan [40] proposed the time-based approximate dynamic programming to solve the HJBE forward-in-time. In [39], the globalized dual heuristic programming was suggested, where three neural networks (NNs) are required to approximate system model, value function, and costate function. For the optimal tracking control of linear discrete-time systems, the Q-learning method [35] was used to learn the solution of the augmented algebraic Riccati equation without the knowledge of system dynamics. In [24], a type of the fuzzy system structure is applied to the heuristic dynamic programming algorithm to solve the discrete-time HJBE. It is noted that most of these works concerned linear or affine nonlinear systems.

For the optimal control problem of discrete-time systems, the optimal control is given by $u^*(x(k)) = \arg \min_u \{ R(x(k), u(x(k))) + V^*(x(k+1)) \}$, where $R(x, u)$ is the utility function that represents the one-step cost and $V^*(x)$ is the optimal value function. For affine nonlinear systems $x(k+1) = f(x(k)) + g(x(k))u(k)$ and linear systems $x(k+1) = Ax(k) + Bu(k)$, the optimal control policy can be explicitly given by $u^*(x(k)) = -(1/2)R^{-1}g(x(k))(\partial V^*(x(k+1))/\partial x(k+1))$. However, for nonaffine nonlinear systems, its optimal control policy is an...
implicit function of the value function, which cannot be given explicitly. Therefore, the HJBE of nonaffine nonlinear systems becomes more complicated, and its solution becomes more difficult. In [52], the optimal control problem of nonaffine nonlinear discrete-time systems was studied, and a policy gradient approximate dynamic programming method was proposed, which uses gradient descent approach for policy improvement instead of greedy policy improvement. The finite-horizon optimal control problem was considered in [53], and an iterative approximate dynamic programming algorithm was suggested to obtain the optimal control policy by involving an $\epsilon$-error bound in the performance index. The convergence and stability properties of policy iteration method were investigated in [54]. An iterative $\theta$-adaptive dynamic programming [55] was proposed, which avoids the use of an initial admissible control policy. In [56], the reinforcement learning method was employed to solve the optimal control problem of the nonaffine nonlinear discrete-time system represented by the nonlinear autoregressive moving average with an exogenous input form. It is noted that the control constraints were not considered in these works.

In real practice, the actuator saturation is unavoidable. Hence, it is meaningful to consider the constrained optimal control problems. In this aspect, some results have been reported for affine nonlinear systems [17], [57]–[60]. Generally, a certain nonquadratic performance index [61] is employed, and then, the optimal control policy can be represented in an explicit form. For example, an off-policy reinforcement learning [17] was proposed to solve the optimal control problem of affine nonlinear continuous-time systems with input constraints. Modares et al. [58] provided an integral reinforcement learning with the use of an experience replay technique. The event-triggered optimal control problem was considered in [59], and an adaptive dynamic programming method was used with three NNs to estimate the system model, optimal value function, and optimal control policy. The single network-based adaptive critics method [60] was proposed for finite-horizon nonlinear constrained optimal control design. However, for the nonaffine nonlinear systems with control constraints, it is difficult to obtain the explicit form for the optimal control policy if using the certain performance index in [61]. Moreover, the data-based constrained optimal control problem of nonaffine nonlinear systems has rarely been studied yet. This remains an open issue, requiring further investigation, which motivates the research of this paper.

To solve this problem, an adaptive constrained optimal control design approach with critic-only value iteration-based Q-learning (CoVIQL) is developed. The constrained control problem is first converted to an unconstrained control problem through system transformation. Then, the VIQL algorithm is proposed to learn the optimal Q-function, and its convergence results are established. By using only one NN to estimate the Q-function, the implementation procedure of the CoVIQL is developed to learn the critic NN weights from real system data. After the convergence of the CoVIQL, the Q-function is applied to design the adaptive constrained optimal controller based on the gradient descent scheme. The numerical simulation studies are conducted to show its effectiveness.

The rest of this paper is arranged as follows. Section II gives some preliminaries. In Section III, the VIQL algorithm is proposed, and its convergence results are established. In Section IV, the adaptive constrained optimal controller is designed based on the VIQL algorithm using the critically stable structure. Section V provides the simulation results, and Section VI gives the conclusions.

II. Preliminaries

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

$$x(k + 1) = f(x(k), u(k))$$

where $x(k) = [x_1(k), \ldots, x_n(k)]^T \in \mathbb{R}^n$ is the state and $u(k) = [u_1(k), \ldots, u_m(k)]^T \in \mathbb{R}^m$ is the control input constrained by $|u_i| \leq \pi_i, \pi_i > 0, \forall i.$

Let $x \in \mathcal{X}, u \in \mathcal{U}$, where $\mathcal{X}$ and $\mathcal{U}$ are two compact sets. For system (1), it is assumed that Assumption 1 holds throughout this paper.

Assumption 1:

1) $f(x, u)$ is Lipschitz continuous on $\mathcal{X} \times \mathcal{U}$ that contains the origin, $f(0, 0) = 0$, and $x = 0$ is the unique equilibrium on $\mathcal{X}$.

2) System (1) is stabilizable on $\mathcal{X}$, i.e., there exists a continuous control function $u(x)$, such that the system is asymptotically stable on a domain $\mathcal{X}$.

Remark 1: For the considered general nonaffine nonlinear discrete-time system (1), Assumption 1 contains the basic assumptions for control design, although it may not be given explicitly in all works. Note that system (1) is a general form, which can be used to describe almost all real systems. Linear and affine nonlinear systems can be viewed as a specific kind of system (1). Thus, there exist similar assumptions, such as Assumption 1, for linear and affine nonlinear systems. For linear systems, i.e., $x(k + 1) = Ax(k) + Bu(k)$, it usually assumes that $(A, B)$ is stabilizable, which is similar to part 2) of Assumption 1, and part 1) of Assumption 1 is obvious. For affine nonlinear systems [14], [57], i.e., $x(k + 1) = f(x(k)) + g(x(k))u(k)$, part 1) of Assumption 1 can be represented as: $f(x)$ and $g(x)$ are Lipschitz continuous on the set $\mathcal{X}$ that contains the origin, $f(0) = g(0) = 0$, and $x = 0$ is the unique equilibrium on $\mathcal{X}$. Part 2) of Assumption 1 remains the same.

Denote $\mathcal{U} = \text{diag}[\mathcal{U}_1, \mathcal{U}_2, \ldots, \mathcal{U}_m]$. To handle the input constraints in the optimal control problem, most of the existing works [17], [57]–[60] use the following performance index:

$$J(x(0), u) = \sum_{l=0}^{\infty} \{S(x(l)) + W(u(l))\}$$

where $S(x)$ and $W(u)$ are positive definite functions, i.e., $S(x) > 0, W(u) > 0$ for $\forall x, u \neq 0$, and $S(x) = 0, W(u) = 0$ only when $x = 0, u = 0$. In many works [17], [57]–[60], $W(u)$ is usually chosen as the following form proposed in [61]:

$$W(u(l)) = 2 \int_0^{u(l)} [\varphi^{-1}(\mathcal{U}^{-1}s)]^T \mathcal{U} \mathcal{R} ds$$

(3)
where \( s = [s_1, \ldots, s_m]^T \in \mathbb{R}^m \), \( \varphi(s) = [\varphi(s_1), \ldots, \varphi(s_m)]^T \), \( \varphi^{-1}(s) = [\varphi^{-1}(s_1), \ldots, \varphi^{-1}(s_m)]^T \), and \( R = \text{diag}(r_1, \ldots, r_m) > 0 \) is assumed to be diagonal for the simplicity of analysis. \( \varphi(\cdot) \) is a bounded one-to-one function satisfying \( |\varphi(\cdot)| \leq 1 \) and belonging to \( C^p(\mathcal{U}) \) and \( L_2(\mathcal{U}) \). Moreover, it is a monotonic odd function with its first derivative bounded by a constant. The well-known hyperbolic tangent function \( \varphi(\cdot) = \tanh(\cdot) \) is an example of such function.

III. DATA-BASED VALUE ITERATION Q-LEARNING

In this section, the motivations of this paper are analyzed first. Then, the VIQL algorithm is proposed, and its convergence results are established.

A. Motivations

It is noted that most of the existing constrained control methods [14], [34], [57], [60] require the use of a certain performance index as in (2), which may be impractical in some cases. In this paper, we aim to solve the data-based constrained optimal control problem of system (1), i.e., the mathematical expression \( f(x, u) \) is unknown. Before starting, we try to analyze several limitations in using the performance index (2), which motivate the research of this paper.

1) First and the most important limitation is that the form of the performance index is restricted to (2) with \( W(u) \) given by (3), which is predetermined by designers artificially. This is generally unreasonable for many practical systems. The performance index in optimal control problem reflects the real cost consumed during the control process, such as fuel consumption of aircraft, which is determined by the real system rather than given by the designers. In other words, the form of the performance index in real systems may not satisfy (2) and \( W(u) \) may not have the form of (3).

2) The main purpose to use the from (3) for \( W(u) \) is to obtain the analytical expression for the optimal control form such as \( u^*(x(k)) = \int \frac{1}{2}R^{-1}g(x(k)) \circ \nabla^* x(k + 1) / \circ x(k + 1) \). This control form can only be given for linear systems \( x(k + 1) = Ax(k) + Bu(k) \) and affine nonlinear systems \( x(k + 1) = f(x(k)) + g(x(k))u(k) \), which is unavailable for nonaffine nonlinear systems. Thus, the use of the performance index (2) with \( W(u) \) be given by (3) is restricted to linear systems, affine nonlinear systems, or nonaffine nonlinear systems with prior system identification to an affine nonlinear representation.

3) Due to the use of explicit expression of the optimal control policy, a full system model or a partial system model is often required. For nonaffine nonlinear systems, the problem becomes more complicated. Thus, data-based approaches are direct for control design to overcome this problem, which avoids the requirement of the mathematical model by using real system data.

To overcome the above-mentioned problems, we develop a VIQL method to solve the data-based constrained optimal control problem of system (1) by using a general performance index through system transformation.

B. System Transformation With General Performance Index

In this paper, the performance index is not restricted to the form (2) with \( W(u) \) given by (3). Consider the following general performance index:

\[
J(x(0), u) \triangleq \sum_{l=0}^{\infty} \mathcal{R}(x(l), u(l))
\]

where \( \mathcal{R}(x, u) \) is a positive definite function, i.e., \( \mathcal{R}(x, u) > 0 \) for \( x, u \neq 0 \), and \( \mathcal{R}(x, u) = 0 \) only when \( x = 0, u = 0 \). \( \mathcal{R}(x, u) \) is measurable with unknown mathematical expression. We consider the following state feedback control:

\[
u(x) = U \tanh(U^{-1}v(x))
\]

where \( v = [v_1, \ldots, v_m]^T \in \mathbb{R}^m \). Based on (5), system (1) can be rewritten as

\[
x(k + 1) = f(x, U \tanh(U^{-1}v))
\]

and the performance index (4) can be rewritten as

\[
J(x(0), u) \triangleq \sum_{l=0}^{\infty} C(x(l), v(l))
\]

where \( C(x, v) \triangleq \mathcal{R}(x, U \tanh(U^{-1}v)) \). Note that the mathematical expressions of \( f(x, u) \) and \( \mathcal{R}(x, u) \) are unknown, which mean that the mathematical expressions \( f(x, U \tanh(U^{-1}v)) \) and \( C(x, v) \) are unknown. The problem to design the constrained control \( u \) is converted to design the unconstrained control \( v \). The unconstrained optimal control problem can be reformulated as

\[
v^*(x) \triangleq \arg \min_v J(x(0), v)
\]

subjected to the transformed system (6). From (5), the constrained optimal control is given by

\[
u^*(x) = U \tanh(U^{-1}v^*(x)).
\]

Remark 2: By using the function \( \tanh(\cdot) \) to give the controller form (5), the constrained optimal control problem is reformulated as an unconstrained control problem. It is worth mentioning that other continuous bounded functions can also be used to replace \( \tanh(\cdot) \), such as \( 2 \log \text{sigmoid}(\cdot) - 1 \), where \( \log \text{sigmoid}(\cdot) \) represents the log-sigmoid transfer function. With different functions, the controlled system may behave differently. In this paper, \( \tanh(\cdot) \) is one choice to give the controller form (5), and the general performance index (4) can be minimized under the controller form (5). Similarly, the controller form (5) can also be given with other functions and the performance index can be minimized with respect to new controller forms, and the method developed in this paper is still valid. Just as analyzed in Section III-A, the motivations of this paper aim to overcome the three difficulties due the use of predetermined performance index (2) with (3). It is unreasonable for real system because its performance index should reflect the real cost consumed during the control process that is determined by the real system rather than given by the designers. For example, the real fuel consumption of an aircraft may not satisfy the predetermined performance index (2) with (3) and cannot be given by the designers.
C. Value Iteration-Based Q-Learning

To solve the data-based unconstrained optimal control problem of the transformed system (6) with the performance index (7), we proposed a VIQL algorithm. Let \( v \in \mathcal{V} \), where \( \mathcal{V} \) is a compact set and denote \( \mathcal{D} \triangleq \{(x, v) | x \in \mathcal{X}, v \in \mathcal{V}\} \). For a stabilizing control policy \( v(x) \), define its state value function as follows:

\[
V_v(x(k)) \triangleq \sum_{l=k}^\infty C(x(l), v(x(l)))
\]

where \( V_v(0) = 0 \). Based on (10), the state-value function \( V_v(x) \) satisfies the following equation:

\[
V_v(x(k)) = C(x(k), v(x(k))) + V_v(x(k + 1))
\]

and the optimal state-value function is represented as

\[
V^*(x) \triangleq V_{v^*}(x) = \min_{v} V_v(x).
\]

The optimal state-value function \( V^*(x) \) satisfies the following HJBE:

\[
V^*(x(k)) = \min_{\mu} [C(x(k), \mu) + V^*(x(k + 1))]
\]

and the optimal control policy is given by

\[
v^*(x(k)) = \arg \min_{\mu} [C(x(k), \mu) + V^*(x(k + 1))].
\]

For the development of the VIQL algorithm, it requires to define the action-state function, i.e., the Q-function, as follows:

\[
Q_v(x(k), a) = C(x(k), a) + V_v(x(k + 1))
\]

where \( Q_v(0, 0) = 0 \). For the optimal control policy \( v^* \), denote its optimal Q-function as \( Q^*(x, a) \) that is given by

\[
Q^*(x(k), a) = C(x(k), a) + V^*(x(k + 1)).
\]

According to (14) and (16), we have that

\[
v^*(x(k)) = \arg \min_a Q^*(x(k), a).
\]

Thus, the design of the optimal control policy \( v^* \) can be converted to finding the optimal Q-function \( Q^*(x, a) \).

For the data-based optimal control problem of system (6), the VIQL algorithm (see Algorithm 1) is proposed to learn the optimal Q-function \( Q^*(x, a) \).

Note that Algorithm 1 is a completely data-based method, which does not require the mathematical model of system dynamics \( f \) and the utility function \( R(x, u) \) in the performance index.

**Algorithm 1 VIQL**

1. **Step 1:** Let \( Q^{(0)}(x, a) \) be an initial Q-function. Let \( i = 0 \).
2. **Step 2:** Update control policy with:

\[
v^{(i)}(x) = \arg \min_a Q^{(i)}(x, a).
\]
3. **Step 3:** Solve the iterative equation

\[
Q^{(i+1)}(x(k), a) = C(x(k), a) + Q^{(i)}(x(k + 1), \hat{v}^{(i)}(x(k + 1)))
\]

for the unknown Q-function \( Q^{(i)} \).
4. **Step 4:** If \( |Q^{(i+1)}(x, a) - Q^{(i)}(x, a)| \leq \epsilon \) for all \( (x, a) \in \mathcal{D} \) and \( \epsilon > 0 \), terminate the iteration; else, let \( i = i + 1 \), go back to Step 2 and continue.

**D. Convergence of VIQL Algorithm**

In this section, the convergence of Algorithm 1 is established to show that the iterative sequence \( \{Q^{(i)}(x, a)\} \) will converge to the optimal Q-function \( Q^*(x, a) \).

**Theorem 1:** For \( \forall (x, a) \in \mathcal{D} \), let \( \{Q^{(i)}(x, a)\} \) and \( \{\hat{v}^{(i)}(x)\} \) be the sequences generated by Algorithm 1 with an initial function \( Q^{(0)}(x, a) = 0 \), that is

\[
\hat{v}^{(i)}(x) = \arg \min_a Q^{(i)}(x, a)
\]

\[
Q^{(i+1)}(x(k), a)
\]

\[
= C(x(k), a) + Q^{(i)}(x(k + 1), \hat{v}^{(i)}(x(k + 1))).
\]

Then, \( \lim_{i\to\infty} Q^{(i)}(x, a) = Q^*(x, a) \).

**Proof:** First, we prove \( Q^{(i)}(x, a) \leq Q^{(i+1)}(x, a) \) with mathematical induction. Note that

\[
Q^{(i)}(x(k), a) = C(x(k), a) + Q^{(0)}(x(k + 1), \hat{v}^{(0)})
\]

\[
\geq C(x(k), a) \geq 0 = Q^{(0)}(x(k), a)
\]

which means that \( Q^{(i)}(x, a) \leq Q^{(i+1)}(x, a) \) holds for \( i = 0 \).

Assume that \( Q^{(i-1)}(x, a) \leq Q^{(i)}(x, a) \) holds. Then

\[
Q^{(i+1)}(x(k), a)
\]

\[
= C(x(k), a) + Q^{(i)}(x(k + 1), \hat{v}^{(i)}(x(k + 1)))
\]

\[
\geq C(x(k), a) + Q^{(i-1)}(x(k + 1), \hat{v}^{(i-1)}(x(k + 1)))
\]

\[
\geq C(x(k), a) + \min_a Q^{(i-1)}(x(k + 1), a)
\]

\[
= C(x(k), a) + Q^{(i-1)}(x(k + 1), \hat{v}^{(i-1)}(x(k + 1)))
\]

\[
= Q^{(i)}(x(k), a).
\]

Based on (22) and (23), we have that \( Q^{(i)}(x, a) \leq Q^{(i+1)}(x, a) \) holds for all \( i \).

Next, we will prove \( Q^{(0)}(x, a) \leq Q^*(x, a) \) by mathematical induction. It is observed that \( Q^{(0)}(x, a) = 0 \leq Q^*(x, a) \).
Assume that $Q_i^{(i)}(x, a) \leq Q^*(x, a)$. Then

$$Q_i^{(i+1)}(x(k), a) = C(x(k), a) + Q_i^{(i)}(x(k + 1), \pi_i^{(i)}(x(k + 1))) \leq C(x(k), a) + \min_a Q_i^{(i)}(x(k + 1), a)$$

$$\leq C(x(k), a) + Q_i^{(i)}(x(k + 1), b^i(x(k + 1))) = C(x(k), a) + C(x(k + 1), b^i(x(k + 1))) + \sum_{i=1}^{\infty} Q_i^{(i)}(x(k + 2), \pi_i^{(i)}(x(k + 2))) = C(x(k), a) + V^*(x(k + 1)) = Q^*(x, a).$$

Therefore, the inequality $Q_i^{(i)}(x, a) \leq Q_i^{(i+1)}(x, a)$ holds for all $i$. That is to say, $\{Q_i^{(i)}(x, a)\}$ is nondecreasing sequence bounded by $Q^*(x, a)$. Considering that a bounded monotone sequence always has a limit, denote $Q^{(\infty)}(x, a) \equiv \lim_{i \to \infty} Q_i^{(i)}(x, a)$. By using $Q^{(\infty)}(x, a)$, define

$$u^{(\infty)}(x) = \arg\min_a Q^{(\infty)}(x, a).$$

Let $V^{(\infty)}(x(k)) \equiv \sum_{i=1}^{\infty} C(x(k), u^{(\infty)}(x(k))))$ be the state value function of control policy $u^{(\infty)}(x)$. From (19)

$$Q^{(\infty)}(x(k), a) = C(x(k), a) + V^{(\infty)}(x(k + 1))$$

$$= C(x(k), a) + \arg\min_a Q^{(\infty)}(x, a)$$

It is observed that (25) and (26) are essentially the same as (16) and (17). Thus, $\lim_{i \to \infty} Q_i^{(i)}(x, a) = Q^*(x, a)$. □

**Theorem 2:** For $\forall (x, a) \in \mathcal{D}$, let $\{Q_i^{(i)}(x, a)\}$ and $\{\pi_i^{(i)}(x)\}$ be the sequences generated by Algorithm 1 with an initial function $Q^{(0)}(x, a) \geq Q^*(x, a)$, that is

$$\pi_i^{(i)}(x) = \arg\min_a Q_i^{(i)}(x, a)$$

$$Q_i^{(i+1)}(x(k), a) = C(x(k), a) + Q_i^{(i)}(x(k + 1), \pi_i^{(i)}(x(k + 1))).$$

Then, $\lim_{i \to \infty} Q_i^{(i)}(x, a) = Q^*(x, a)$.

**Proof:** First, the mathematical induction is used to prove $Q_i^{(i)}(x, a) \geq Q^*(x, a)$. From the condition, $Q_i^{(i)}(x, a) \geq Q^*(x, a)$, which means that $Q_i^{(i)}(x, a) \geq Q^*(x, a)$ holds for $i = 0$. Assume that $Q_i^{(i)}(x, a) \geq Q^*(x, a)$. Then

$$Q_i^{(i+1)}(x(k), a) = C(x(k), a) + Q_i^{(i)}(x(k + 1), \pi_i^{(i)}(x(k + 1))) \geq C(x(k), a) + Q^*(x(k + 1), \pi_i^{(i)}(x(k + 1))) \geq C(x(k), a) + \min_a Q^*(x(k + 1), a)$$

$$= C(x(k), a) + V^*(x(k + 1)) = Q^*(x, a).$$

This means that $Q_i^{(i)}(x, a) \geq Q^*(x, a)$ holds for all $i$. Thus, we have

$$\lim_{i \to \infty} Q_i^{(i)}(x, a) \geq Q^*(x, a).$$

Let $\mu(x)$ be a stabilizing control policy. Then, with the control $\mu(x)$, the system state $x(k) = 0$ as time $k \to \infty$. Then, we have

$$\lim_{k \to \infty} Q_i^{(i)}(x(k), \pi_i^{(i)}(x(k))) = 0.$$ (31)

According to (27) and (28), we have

$$Q_i^{(i)}(x(k), a) \geq C(x(k), a) + \arg\min_a Q_i^{(i)}(x(k + 1), \mu(x(k + 1))) \geq C(x(k), a) + \min_a Q_i^{(i)}(x(k + 1), a)$$

$$\leq C(x(k), a) + Q_i^{(i)}(x(k + 1), \mu(x(k + 1))) = C(x(k), a) + Q_i^{(i)}(x(k + 1), \mu(x(k + 1))) + Q_i^{(i-1)}(x(k + 2), \pi_i^{(i-1)}(x(k + 2)))$$

$$\leq C(x(k), a) + \sum_{i=1}^{\infty} C(x(k + l), \mu(x(k + l))) = C(x(k), a) + \sum_{i=1}^{\infty} C(x(k + l), \mu(x(k + l))) = Q_{\mu}(x, a).$$

Based on (30) and (33), we have

$$Q^*(x, a) \leq \lim_{i \to \infty} Q_i^{(i)}(x, a) \leq Q_{\mu}(x, a).$$

Taking the minimum with respect to control $\mu$ on (34) yields $Q^*(x, a) = \lim_{i \to \infty} Q_i^{(i)}(x, a) \leq Q^*(x, a)$, which means that $\lim_{i \to \infty} Q_i^{(i)}(x, a) = Q^*(x, a)$. □

**Theorem 3:** For $\forall (x, a) \in \mathcal{D}$, let $\{Q_i^{(i)}(x, a)\}$ and $\{\pi_i^{(i)}(x)\}$ be the sequences generated by Algorithm 1 with an initial function $Q_0^{(0)}(x, a) \geq 0$. Then, $\lim_{i \to \infty} Q_i^{(i)}(x, a) = Q^*(x, a)$.

**Proof:** Let the sequences $\{Q_i^{(i)}(x, a)\}$ and $\{\pi_i^{(i)}(x)\}$ be defined in Theorem 1 with the initial function given by

$$Q_0^{(0)}(x, a) = 0.$$ (35)

The sequences $Q_i^{(i)}(x, a)$ and $\pi_i^{(i)}(x)$ are defined in Theorem 2 with the initial function given by

$$Q_0^{(0)}(x, a) = \max\{Q_0^{(0)}(x, a), Q^*(x, a)\}.\quad (36)$$

Note that $Q_0^{(0)}(x, a) \geq Q^*(x, a)$.

We use mathematical induction to prove the result $Q_i^{(i)}(x, a) \leq Q_i^{(i)}(x, a) \leq Q_0^{(0)}(x, a)$ holds for all $i$. From (35) and (36), we have

$$Q_0^{(0)}(x, a) \leq Q_i^{(i)}(x, a) \leq Q_0^{(0)}(x, a)$$.
which means that the result \( \overline{Q}^{(i)}(x, a) \leq Q^{(i)}(x, a) \leq \underline{Q}^{(i)}(x, a) \) holds for \( i = 0 \).

Assume that \( \overline{Q}^{(i-1)}(x, a) \leq Q^{(i-1)}(x, a) \leq \underline{Q}^{(i-1)}(x, a) \) holds. On the one hand

\[
\overline{Q}^{(i)}(x(k), a) = C(x(k), a) + \overline{Q}^{(i-1)}(x(k + 1), b^{(i-1)}(x(k + 1)))
\]

\[
\leq C(x(k), a) + \min_a \overline{Q}^{(i-1)}(x(k + 1), a)
\]

\[
\leq C(x(k), a) + Q^{(i-1)}(x(k + 1), b^{(i-1)}(x(k + 1)))
\]

\[
\leq C(x(k), a) + Q^{(i-1)}(x(k + 1), b^{(i-1)}(x(k + 1)))
\]

\[
Q^{(i)}(x(k), a).
\]

On the other hand

\[
\begin{aligned}
Q^{(i)}(x(k), a) &= C(x(k), a) + Q^{(i-1)}(x(k + 1), b^{(i-1)}(x(k + 1)))
\leq C(x(k), a) + \min_a Q^{(i-1)}(x(k + 1), a)
\leq C(x(k), a) + Q^{(i-1)}(x(k + 1), b^{(i-1)}(x(k + 1)))
\leq C(x(k), a) + Q^{(i-1)}(x(k + 1), b^{(i-1)}(x(k + 1)))
\end{aligned}
\]

\[
Q^{(i)}(x(k), a).
\]

According to (38) and (39), we have

\[
Q^{(i)}(x, a) \leq \overline{Q}^{(i)}(x, a) \leq \underline{Q}^{(i)}(x, a).
\]

Based on (37) and (40), we have that the result \( \overline{Q}^{(i)}(x, a) \leq Q^{(i)}(x, a) \leq \underline{Q}^{(i)}(x, a) \) holds for all \( i \). According to Theorems 1 and 2, \( \lim_{i \to \infty} \underline{Q}^{(i)}(x, a) = Q^*(x, a) \) and \( \lim_{i \to \infty} \overline{Q}^{(i)}(x, a) = Q^*(x, a) \). Therefore, \( Q^*(x, a) \leq \lim_{i \to \infty} Q^{(i)}(x, a) \), i.e., \( \lim_{i \to \infty} Q^{(i)}(x, a) = Q^*(x, a) \).

In Theorem 3, we prove the convergence of Algorithm 1 with the initial condition \( Q^{(0)}(x, a) \geq 0 \), which is easy to realize in practice. One simple way is to let \( Q^{(0)}(x, a) \equiv 0 \) or \( Q^{(0)}(x, a) = \begin{bmatrix} x^T & D & x^T \end{bmatrix} \), where \( D \) is an arbitrary positive semidefinite matrix.

IV. ADAPTIVE CONSTRAINED OPTIMAL CONTROL

WITH CRITIC-ONLY STRUCTURE

Based on Algorithm 1, the CoVQL is developed in this section. In the CoVQL, the critic-only structure is introduced by using only one NN to approximate the Q-function. With the Q-function obtained from the CoVQL, the adaptive constrained optimal controller is designed based on the gradient descent scheme.

A. Implementation of VIQL With Critic-Only Structure

In order to solve (19) for the unknown Q-function \( Q^{(i)} \), the critic NN is employed to approximate the unknown Q-function \( Q^{(i)}(x, a) \) on \( D \). \( Q^{(i)}(x, a) \) can be represented by

\[
Q^{(i)}(x, a) = \sum_{j=1}^{L} \theta_j^{(i)} \psi_j(x, a) + \epsilon^{(i)}(x, a)
\]

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

\[
\hat{Q}^{(i)}(x, a) = \sum_{j=1}^{L} \hat{\theta}_j^{(i)} \psi_j(x, a) = \Psi_L^T(x, a) \hat{\theta}^{(i)}
\]

where \( \hat{\theta}^{(i)} = [\hat{\theta}_1^{(i)}, \ldots, \hat{\theta}_L^{(i)}]^T \) is the estimation of the ideal constant critic NN weight vector \( \theta^{(i)} \). According to (18), we have

\[
\hat{\theta}^{(i)}(x) = \arg \min_a \hat{Q}^{(i)}(x, a). \tag{43}
\]

For \( \forall x \in \mathcal{X} \), the equation is an optimization problem with respect to \( a \). Thus, the gradient descent method is employed as follows:

\[
\hat{\theta}^{(i)}(x) = \hat{\theta}^{(i-1)}(x) - a \frac{\partial \hat{Q}^{(i)}(x, a)}{\partial a}_{a = \hat{\theta}^{(i-1)}(x)}
\]

\[
= \hat{\theta}^{(i-1)}(x) - a \frac{\partial \Psi_L^T(x, a) \hat{\theta}^{(i)}}{\partial a}_{a = \hat{\theta}^{(i-1)}(x)} \tag{44}
\]

with \( \hat{\theta}^{(i,0)}(x) = \hat{\theta}^{(0)}(x) \), where \( a > 0 \). After several iterative steps on index \( j \), \( \hat{\theta}^{(i)}(x) \) takes the final \( \hat{\theta}^{(i)}(x) \).

To compute \( \hat{\theta}^{(i)} \) for \( \hat{Q}^{(i)}(x, a) \), a least-squares scheme is developed using real system data. For notation simplicity, denote \( (x, a, x', C(x, a)) \) as a data measured from the real system (1), where \( a = \theta^T \tanh^{-1}(\theta^{-1} u) \), and \( x' \) represents the next state under the control action \( u \) at state \( x \), i.e., \( x' = f(x, u) \). For real implementation of the CoVQL, \( x' \) is measured with sensors from real system without requiring the mathematical system model \( f \).

By using \( \hat{Q}^{(i)}(x, a) \) and \( \hat{\theta}^{(i)}(x) \), (19) is written as

\[
\epsilon^{(i+1)}(x, a) = \hat{Q}^{(i+1)}(x, a) - \hat{Q}^{(i)}(x', \hat{\theta}^{(i)}(x')) - C(x, a)
\]

\[
= \Psi_L^T(x, a) \hat{\theta}^{(i+1)} - \Psi_L^T(x', \hat{\theta}^{(i)}(x')) \hat{\theta}^{(i)} - C(x, a) \tag{45}
\]

where \( \epsilon^{(i+1)}(x, a) \) is the residual error due to the critic NN approximation errors \( e^{(i)} \) on \( \hat{Q}^{(i)}(x, a) \) and \( e^{(i+1)} \) on \( \hat{Q}^{(i+1)}(x, a) \). In (45), \( \hat{\theta}^{(i+1)} \) is the unknown vector requires to be computed with real system data. Let \( S_M \triangleq \{ (x_l^1, a_l^1, x_l^1, \hat{C}_l^1) | (x_l^1, a_l^1) \in D, l = 1, 2, \ldots, M \} \) denote the data set sampled from real system, where \( M \) is its size. Before starting the CoVQL, the data set \( S_M \) should be collected from the measurements of sensors during the operations of real system. For each data \( (x_l^1, a_l^1, x_l^1, \hat{C}_l^1) \) in \( S_M \), the residual error (45) is given by

\[
\epsilon_l^{(i+1)} = \Psi_L^T(x_l^1, a_l^1) \hat{\theta}^{(i+1)} - \Psi_L^T(x_l^1, \hat{\theta}^{(i)}(x_l^1)) \hat{\theta}^{(i)} - \hat{C}_l^1 \tag{46}
\]

where \( \epsilon_l^{(i)} \triangleq \epsilon_l^{(i)}(x_l^1, a_l^1) \) and \( \hat{C}_l^1 \triangleq \hat{C}(x_l^1, a_l^1) \). The critic NN weight vector \( \hat{\theta}^{(i+1)} \) can be computed by minimizing the sum of residual errors, that is

\[
\min_{i=1}^{M} (\epsilon_l^{(i+1)})^2. \tag{47}
\]
Then, the least-squares scheme is derived as follows:

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

(48)

where $\eta(i) \triangleq [\eta_1(i) \cdots \eta_M(i)]_a^T$ and $Z \triangleq [z_1 \cdots z_M]^T$, with $z_i \triangleq \Psi_L(x_i^a, a_i^a)$ and $\eta_i(i) \triangleq \Psi_L^T(x_i^a, \hat{\theta}(i))(y_i^a) - C_i$.

Based on the above-mentioned discussion, the critic-only NN structure for the implementation of VIQL is summarized in Fig. 1. By using the derived least-squares scheme (48), the following CoVIQL implementation procedure is developed to learn the optimal Q-function.

**Algorithm 2 CoVIQL**

- **Step 1:** Let $Q(0)(x, a) \geq 0$ and $i = 0$;
- **Step 2:** Compute critic NN weight vector $\hat{\theta}(i+1)$ with (48).
- **Step 3:** If $\|\hat{\theta}(i+1) - \hat{\theta}(i)\| \leq \epsilon$ ($\epsilon > 0$ is a small parameter), stop the iteration; else, $i = i + 1$, go back to **Step 2** and continue.

**B. Adaptive Constrained Optimal Control**

Algorithm 2 is to learn the optimal Q-function. After the convergence of Algorithm 2, the obtained Q-function is used to design the adaptive constrained optimal controller. Denote the converged critic NN weight vector as $\theta_c$. According to (42), the converged Q-function is given by $Q_c(x, a) = \Psi_L^T(x, a)\theta_c$. According to (14), the control policy based on the Q-function $Q_c(x, a)$ is given by

$$v(k) = \arg \min_a Q_c(y(k), a).$$

(49)

Note that (49) is an optimization problem with respect to $a$. To solve this problem, the gradient descent method can be employed at each time instant $k$, that is

$$v^{j+1}(x(k)) = v^j(x(k)) - \alpha \frac{\partial Q_c(x(k), a)}{\partial a} \bigg|_{a=v^j(x(k))}$$

$$= v^j(x(k)) - \alpha \frac{\partial \Psi_L^T(x, a)}{\partial a} \bigg|_{a=v^j(x(k))} \theta_c$$

(50)

with $v^0(x(k)) = v(x(k-1))$. The index $j$ represents the iteration of the gradient descent method. $v(x(k))$ is the converged value of $v^j(x(k))$ after several iterations. Therefore, it follows from (5) that the adaptive constrained controller is given by:

$$u(x(k)) = \mathcal{U} \tanh(\mathcal{U}^{-1} v((x(k)))).$$  

(51)

It is worth mentioning that the developed CoVIQL method also suits for solving the unconstrained control problem by using very large bound $\mathcal{U}$ on control input $u$.

**Remark 3:** The CoVIQL method is a data-based method, which does not require the mathematical model of the real systems. If a model is available, it may be helpful for data-based control design approaches. For the developed CoVIQL method, the model could be useful to give a better initial Q-function $Q^{(0)}$, which is more close to the optimal Q-function than the arbitrary given $Q^{(0)} \geq 0$ and the number of iteration needed may be reduced. Combining model-based approaches and data-based reinforcement learning for controller design is an interesting and promising issue, which remains an open topic. Its theoretical analysis requires our further investigation that is beyond the consideration of this paper.

**Remark 4:** The developed CoVIQL method aims to overcome the three problems mentioned in Section III-A. Compared with the existing representative constrained control approaches [17], [57]–[60], [62], there are some differences in the developed CoVIQL method.

1) In [17], [57]–[60], and [62], the methods were proposed for linear or affine nonlinear systems, while the CoVIQL method is developed for nonaffine nonlinear systems.

2) In [17], [57]–[60], and [62], the methods require a certain performance index, which is unreasonable in practice as analyzed in Section III-A. For the developed CoVIQL method, there does not exist restrictions on the form of the performance index and the general performance index can be used.

3) The reinforcement learning methods used in [58]–[60], and [62] are model-based and partly model-based, or prior model identification is required before control design. The developed CoVIQL method is completely data-based, where the mathematical expressions of the system model and utility function in the performance index are not required.

4) The developed CoVIQL method is a kind of Q-learning, where the action-state value function, i.e., Q-function $Q(x, a)$, is employed and its convergence is analyzed. In [17], [57]–[60], and [62], the reinforcement learning methods were proposed based on state value function $V(x)$.

**V. Simulation Studies**

In this section, the effectiveness of the developed adaptive constrained optimal control method is tested through simulation studies. Considering that the optimal Q-function of nonlinear systems is often unavailable, we first use an unconstrained simple linear system to demonstrate that the CoVIQL method will learn the optimal Q-function. Then, control constraints are added to linear system to show the changes in control performance. Subsequently, the adaptive constrained optimal control approach is applied to a nonlinear system.

**A. Example 1: Unconstrained Linear System**

Consider the following linear system:

$$x(k + 1) = Ax(k) + Bu(k)$$

(52)
with
\[ A = \begin{bmatrix} 0.8336 & -0.1844 \\ -0.6447 & 0.4335 \end{bmatrix}, \quad B = \begin{bmatrix} 0.0065 \\ -0.5830 \end{bmatrix} \]
and \[ x(0) = \begin{bmatrix} -1 \\ 3 \end{bmatrix}. \]

For the linear system (52), use the performance index
\[ J(x(0), u) = \sum_{i=0}^{\infty} x^T(I)Sx(I) + u^T(I)Ra(I) \]
with \( S = \text{diag}(0.5, 0.5) \) and \( R = 1 \). By using a large control bound on \( u \), this becomes an unconstrained optimal control problem. To show the effectiveness of the developed CoVIQL, we first derive the optimal Q-function of the unconstrained linear optimal control problem for comparison purpose.

From the optimal control theories [19], [45], for this linear quadratic regulation problem of system (52), its optimal value function is \( V^*(x) = x^T P x \), where \( P \geq 0 \). It follows from (14) that:
\[ u^*(x(k)) = -R^{-1}B^T P x(k + 1) \]
\[ = -R^{-1}B^T P [A x(k) + Bu^*(x(k))]. \]

Through simplification, we have
\[ u^*(x) = -(R + B^T P B)^{-1} B^T P A x. \] (53)

According to (13) and (53), we have that \( P \) satisfies the following algebraic Riccati equation:
\[ A^T P A - P - A^T P B (B^T P B + R)^{-1} B^T P A + S = 0. \] (54)

By using \( V^*(x) = x^T P x \) for \( \forall x \in \mathcal{X} \), it follows from (16) that:
\[ Q^*(x(k), a) \]
\[ = x^T(k) S x(k) + a^T(k) Ra + x^T(k + 1) P x(k + 1) \]
\[ = x^T(k) S x(k) + a^T(k) Ra + [A x(k) + Ba]^T P [A x(k) + Ba] \]
\[ = \begin{bmatrix} x(k) \\ a \end{bmatrix}^T G \begin{bmatrix} x(k) \\ a \end{bmatrix} \] (55)
where \( G \) is a block matrix that is denoted as
\[ G \triangleq \begin{bmatrix} G_{11} & G_{12} \\ G_{21} & G_{22} \end{bmatrix} \] (56)
with \( G_{11} = S + A^T P A, G_{12} = A^T P B, \) and \( G_{22} = R + B^T P B \). By using MATLAB command DARE to solve the algebraic Riccati equation (54), we have
\[ P = \begin{bmatrix} 5.0362 & -1.3874 \\ -1.3874 & 0.9523 \end{bmatrix}. \]

According to (56), the matrix \( G \) is obtained as follows:
\[ G = \begin{bmatrix} 5.8867 & -1.7066 & 1.0653 \\ -1.7066 & 1.0720 & -0.3998 \\ 1.0653 & -0.3998 & 1.3344 \end{bmatrix}. \] (57)

To use the CoVIQL (i.e., Algorithm 2), chose \( a = 0.02 \) and the critic NN activation function \( \Psi_{NN}(x, a) = [x_1^7, x_1 x_2, x_3 a, x_2 a, a^2]^T \). According to (55)–(57), the ideal optimal critic NN weight vector is given by
\[ \hat{\theta} = [5.8867, -3.4132, 2.1306, 1.0720, -0.7995, 1.3344]^T. \] (58)

Without loss of generality, select the initial \( \hat{\theta}^{(0)}(x, a) = 0 \), i.e., \( \hat{\theta}^{(0)} = 0 \). For the adaptive control (50), ten iterative steps are employed on index \( j \). With the above-mentioned parameters, use the CoVIQL (i.e., Algorithm 2) to learn the optimal Q-function with real system data, and the algorithm achieves convergence at \( i = 52 \) iteration. Figs. 2–6 show the simulation results. Figs. 2 and 3 give the critic NN weight parameters at each iteration, where the dotted lines represent the ideal optimal critic NN weight parameters given by (58). It is observed that the converged critic NN weight vector \( \hat{\theta} \) approaches to the ideal optimal vector in (58). With \( \hat{\theta} \), the adaptive controller (51) is applied for closed-loop simulation. Figs. 4 and 5 give the trajectories of system states and control, respectively. To compute the real cost generated with the adaptive control method, define the cost with respect to
Fig. 5. For Example 1, the trajectory of the adaptive unconstrained controller $u(k)$.

Fig. 6. For Example 1, the trajectory of the real cost $J(k)$.

Fig. 7. For Example 2, the critic NN weight parameters $\hat{\theta}_1^{(i)} \sim \hat{\theta}_3^{(i)}$ at each iteration.

Fig. 8. For Example 2, the critic NN weight parameters $\hat{\theta}_4^{(i)} \sim \hat{\theta}_6^{(i)}$ at each iteration.

Fig. 9. For Example 2, the trajectories of the system state $x(k)$.

Fig. 10. For Example 2, the trajectory of the adaptive constrained controller $u(k)$.

$J(k) \triangleq \sum_{i=0}^{k} R(x(k), u(k))$. \hspace{1cm} (59)

Fig. 6 shows the trajectory of $J(k)$, which converges to 24.4297.

B. Example 2: Constrained Linear System

Consider the linear system (52) with control constraint $\|u_i\| \leq 0.8$, i.e., $|u_i| \leq 0.8$. It is observed from Fig. 5 that the unconstrained control method in Example 1 violates the constraint. By using the same parameters as for Example 1, the CoVIQL (i.e., Algorithm 2) is applied to solve the constrained optimal control problem with real system data. Figs. 7 and 8 give the critic NN weight parameters at each iteration. It is observed that the critic NN weight vector converges to

$\theta_c = [4.9844, -2.8949, 1.5118, 0.9948, -0.3718, 0.5375]^T$ at $i = 41$ iteration. By using the converged critic NN weight vector $\theta_c$, the adaptive constrained controller (51) is applied for closed-loop simulation. The trajectories of system states and control are shown in Figs. 9 and 10, respectively. From Fig. 10, it is found that the control constraint $|u_i| \leq 0.8$ is satisfied. The trajectory of the real cost is demonstrated in Fig. 11, where the $J(k)$ converges to 32.0085.

C. Example 3: Constrained Nonlinear System

This example is a torsional pendulum system used in [54], which is given as follows:

$$\begin{align*}
\frac{d\theta}{dt} &= \omega \\
\frac{d\omega}{dt} &= u - Mgl \sin(\theta) - f_d \omega
\end{align*}$$

\hspace{1cm} (60)
The following discrete-time system is obtained using the Euler and trapezoidal methods for discretization to the system (60). The system states $\alpha$ to solve this constrained optimal control problem, select $a$.

The performance index is $J(x(0), u) = \sum_{t=0}^{\infty} 0.5\dot{\vartheta}^2(l) + 5\omega^2(l) + 0.05a^2(l)$. To use the CoVIQL (i.e., Algorithm 2) to solve this constrained optimal control problem, select $a = 5$ and the critic NN activation function $\Psi_L((\dot{\vartheta}, \omega), a) = [\dot{\vartheta}^2, \dot{\vartheta}\dot{\omega}, \dot{\vartheta}a, \omega^2, \omega a, a^2, \dot{\vartheta}^3, \dot{\vartheta}^2\dot{\omega}, \dot{\vartheta}^2 a, \omega^3, \omega^2\dot{\vartheta}, \omega^2 a]^T$. Without loss of generality, select the initial $\hat{Q}(0)((\dot{\vartheta}, \omega), a) = 0$, i.e., $\hat{Q}(0) = 0$. For the adaptive control (50), 20 iterative steps are employed on index $j$.

With the above-mentioned parameters, the CoVIQL (i.e., Algorithm 2) is employed to learn the optimal Q-function of system (61). Figs. 12–15 show the critic NN weight parameters at each iteration.

Fig. 12. For Example 3, the critic NN weight parameters $\hat{\theta}_4(i) \sim \hat{\theta}_6(i)$ at each iteration.

Fig. 13. For Example 3, the critic NN weight parameters $\hat{\theta}_4(i) \sim \hat{\theta}_6(i)$ at each iteration.

Fig. 14. For Example 3, the critic NN weight parameters $\hat{\theta}_5(i) \sim \hat{\theta}_9(i)$ at each iteration.

Fig. 15. For Example 3, the critic NN weight parameters $\hat{\theta}_5(i) \sim \hat{\theta}_9(i)$ at each iteration.

Fig. 16. For Example 3, the trajectories of the current angle $\dot{\vartheta}(k)$ and angular velocity $\omega(k)$.

the converged critic NN weight vector $\hat{\theta}_c$, the adaptive constrained controller (51) is applied for closed-loop simulation. The trajectories of system states and control
are shown in Figs. 16 and 17, respectively. It is observed from Fig. 17 that the control constraint $|u_1| \leq 1$ is satisfied. Fig. 18 gives the trajectory of the real cost $J(k)$ that converges to 13.0352.

VI. Conclusion

The data-based constrained optimal control problem of nonaffine nonlinear discrete-time systems was investigated in this paper. The CoVIQL method was developed to learn the optimal Q-function by using the real system model rather than the mathematical system model. Through a system transformation, the constrained control problem was converted to an unconstrained control problem. Then, the VIQL algorithm was proposed, and its convergence results were established. By using only one critic NN to estimate the Q-function, the CoVIQL implementation procedure was developed, and the adaptive constrained optimal controller was designed. It is worth mentioning that the developed CoVIQL method suits for general performance index rather than being restricted to the predetermined form. Through the numerical simulation studies on three examples, the results demonstrated the effectiveness of the developed adaptive constrained optimal controller.

There are some related topics for us and interested readers to conduct further studies on the CoVIQL method. For example, the approximation errors were considered in [63]–[66] for control design. The studies on the CoVIQL method incorporating the effects of approximation errors are an interesting topic. The application of the CoVIQL method to complicated practical systems requires further investigation, e.g., aircrafts, ecosystems, transportation systems, and so on.

REFERENCES


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. In 2013, he was a Research Assistant with City University of Hong Kong, Hong Kong. In 2013 (for three months), 2014 (for two months), and 2015 (for two months), he was a Research Assistant/Associate with Texas A&M University at Qatar, Doha, Qatar. He is currently an Associate Professor with the Institute of Automation, Chinese Academy of Sciences, Beijing. His current research interests include distributed parameter systems, optimization control, data-based control, reinforcement learning, deep 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 is the Secretariat of the Adaptive Dynamic Programming and Reinforcement Learning Technical Committee, Chinese Association of Automation. He serves as an Associate Editor of the IEEE TRANSACTIONS ON EMERGING TOPICS IN COMPUTATIONAL INTELLIGENCE and the Artificial Intelligence Review.

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 became a Full Professor of electrical and computer engineering and of computer science with the University of Illinois at Chicago, Chicago, IL, USA, in 2006. He served as the Associate Director of the State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing, China, from 2010 to 2015. He was selected for the 100 Talents Program by the Chinese Academy of Sciences in 2008. He has authored 18 books.

Dr. Liu is a fellow of the International Neural Network Society and the International Association of Pattern Recognition. He was the Editor-in-Chief of the IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS from 2010 to 2015. He is the Editor-in-Chief of Artificial Intelligence Review (Springer).

Huai-Ning Wu was born in Anhui, China, in 1972. He received the B.E. degree in automation from the Shandong Institute of Building Materials Industry, Jinan, China, in 1992, and the Ph.D. degree in control theory and control engineering from Xi’an Jiaotong University, Xi’an, China, in 1997. From 1997 to 1999, he was a Post-Doctoral Research Fellow with the Department of Electronic Engineering, Beijing Institute of Technology, Beijing, China. Since 1999, he has been with the School of Automation Science and Electrical Engineering, Beihang University, Beijing. From 2005 to 2006, he was a Senior Research Associate with the City University of Hong Kong (CityU), Hong Kong. From 2006 to 2008, from 2010 to 2011, and in 2013, he was a Research Fellow with CityU. He is currently a Professor with Beihang University and also a Distinguished Professor of the Yangtze River Scholar with the Ministry of Education of China, Beijing. His current research interests include robust control, fault-tolerant control, distributed parameter systems, and fuzzy/neural modeling and control.

Dr. Wu was a recipient of the China National Funds for Distinguished Young Scientists. He serves as an Associate Editor of the IEEE TRANSACTIONS ON FUZZY SYSTEMS and the IEEE TRANSACTIONS ON SYSTEMS, MAN, & CYBERNETICS: SYSTEMS.