FINITE-HORIZON OPTIMAL CONTROL OF DISCRETE-TIME LINEAR SYSTEMS WITH COMPLETELY UNKNOWN DYNAMICS USING Q-LEARNING

JINGANG ZHAO∗
School of Automation
Beijing Institute of Technology
Beijing, 100081, China

Chi Zhang
School of Automation
Beijing Institute of Technology
Beijing, 100081, China

(Communicated by Biao Luo)

Abstract. This paper investigates finite-horizon optimal control problem of completely unknown discrete-time linear systems. The completely unknown here refers to that the system dynamics are unknown. Compared with infinite-horizon optimal control, the Riccati equation (RE) of finite-horizon optimal control is time-dependent and must meet certain terminal boundary constraints, which brings the greater challenges. Meanwhile, the completely unknown system dynamics have also caused additional challenges. The main innovation of this paper is the developed cyclic fixed-finite-horizon-based Q-learning algorithm to approximate the optimal control input without requiring the system dynamics. The developed algorithm main consists of two phases: the data collection phase over a fixed-finite-horizon and the parameters update phase. A least-squares method is used to correlate the two phases to obtain the optimal parameters by cyclic. Finally, simulation results are given to verify the effectiveness of the proposed cyclic fixed-finite-horizon-based Q-learning algorithm.

1. Introduction. Considerable research efforts have been devoted to optimal control due to its importance from both theoretical and practical perspectives. It has been widely used in many fields such as industrial processes, investment, aerospace, robotics, vehicles and networked control systems, and is also an important part of control theory [16, 4, 5, 37]. Optimal control can maximize benefits and minimize costs, resource consumption. From mathematical perspective, finding the optimal controller is equivalent to solving the Hamilton-Jacobi-Berman (HJB) equation for nonlinear systems and solving the RE equation for linear systems. Dynamic programming (DP) method is an effective tool for solving optimal control problems.

2010 Mathematics Subject Classification. Primary: 58F15, 58F17; Secondary: 53C35.

Key words and phrases. Finite-horizon, Optimal control, Discrete-time linear systems, Completely unknown dynamics, Q-learning.

The first author is supported by International Graduate Exchange Program of Beijing Institute of Technology; This paper is supported by the National Natural Science Foundation of China grant 61673065.

∗Corresponding author: Jingang Zhao.
However, the computational complexity of this method increases sharply as the number of system state and input dimension increases. This is the so-called "curse of dimensionality" problem [29].

During the last decades, many researchers have made much efforts to overcome the problem of "curse of dimensionality". Based on dynamic programming method, a large number of reinforcement learning (RL) methods such as approximate dynamic programming (ADP), Neuro-dynamic programming (NDP) and adaptive dynamic programming (ADP) have thus been proposed to deal well with the problem of "curse of dimensionality" problem [36, 11, 8, 30, 41, 45, 40, 23, 38, 2, 3]. In the process of various RL methods emerged, optimal control also realizes the transition from model-based reinforcement learning to model-free reinforcement learning. For model-based reinforcement learning, interested readers can view [39, 20, 13] and the references therein. For model-free reinforcement learning, interested readers can view [7, 1, 44, 23, 22, 10, 28, 14, 31, 32] and the references therein. Q-learning is also one of the most popular and powerful reinforcement learning method, which has achieved lots of research results in theory and application [19, 35]. In [27], an off-policy actor-critic neural network structure Q-learning method is developed to tackle the optimal output regulation problem of discrete-time systems. The authors in [24, 23] develop a critic-only Q-learning method. In [26], the authors develop a multistep Q-learning method to solve the optimal output regulation problem for 2-degree-of-freedom helicopter. In [25], a based-Q function policy gradient adaptive dynamic programming algorithm is proposed to tackle the optimal control problem of general discrete-time nonlinear systems. In [18], the authors present a novel off-policy actor-critic neural network structure Q-learning method is developed to tackle the optimal output regulation problem of discrete-time systems. The authors in [24, 23] develop a critic-only Q-learning method. In [26], the authors develop a multistep Q-learning method to solve the optimal output regulation problem for 2-degree-of-freedom helicopter. In [25], a based-Q function policy gradient adaptive dynamic programming algorithm is proposed to tackle the optimal control problem of general discrete-time nonlinear systems. In [18], the authors present a novel off-policy actor-critic neural network structure Q-learning method is developed to tackle the optimal output regulation problem of discrete-time systems. The authors in [24, 23] develop a critic-only Q-learning method. In [26], the authors develop a multistep Q-learning method to solve the optimal output regulation problem for 2-degree-of-freedom helicopter. In [25], a based-Q function policy gradient adaptive dynamic programming algorithm is proposed to tackle the optimal control problem of general discrete-time nonlinear systems. In [18], the authors present a novel off-policy actor-critic neural network structure Q-learning method is developed to tackle the optimal output regulation problem of discrete-time systems. The authors in [24, 23] develop a critic-only Q-learning method. In [26], the authors develop a multistep Q-learning method to solve the optimal output regulation problem for 2-degree-of-freedom helicopter. In [25], a based-Q function policy gradient adaptive dynamic programming algorithm is proposed to tackle the optimal control problem of general discrete-time nonlinear systems. In [18], the authors present a novel off-policy actor-critic neural network structure Q-learning method is developed to tackle the optimal output regulation problem of discrete-time systems. The authors in [24, 23] develop a critic-only Q-learning method. In [26], the authors develop a multistep Q-learning method to solve the optimal output regulation problem for 2-degree-of-freedom helicopter. In [25], a based-Q function policy gradient adaptive dynamic programming algorithm is proposed to tackle the optimal control problem of general discrete-time nonlinear systems. In [18], the authors present a novel off-policy actor-critic neural network structure Q-learning method is developed to tackle the optimal output regulation problem of discrete-time systems. The authors in [24, 23] develop a critic-only Q-learning method. In [26], the authors develop a multistep Q-learning method to solve the optimal output regulation problem for 2-degree-of-freedom helicopter. In [25], a based-Q function policy gradient adaptive dynamic programming algorithm is proposed to tackle the optimal control problem of general discrete-time nonlinear systems. In [18], the authors present a novel off-policy actor-critic neural network structure Q-learning method is developed to tackle the optimal output regulation problem of discrete-time systems. The authors in [24, 23] develop a critic-only Q-learning method. In [26], the authors develop a multistep Q-learning method to solve the optimal output regulation problem for 2-degree-of-freedom helicopter. In [25], a based-Q function policy gradient adaptive dynamic programming algorithm is proposed to tackle the optimal control problem of general discrete-time nonlinear systems. In [18], the authors present a novel off-policy actor-critic neural network structure Q-learning method is developed to tackle the optimal output regulation problem of discrete-time systems. The authors in [24, 23] develop a critic-only Q-learning method. In [26], the authors develop a multistep Q-learning method to solve the optimal output regulation problem for 2-degree-of-freedom helicopter. In [25], a based-Q function policy gradient adaptive dynamic programming algorithm is proposed to tackle the optimal control problem of general discrete-time nonlinear systems. In [18], the authors present a novel off-policy actor-critic neural network structure Q-learning method is developed to tackle the optimal output regulation problem of discrete-time systems. The authors in [24, 23] develop a critic-only Q-learning method. In [26], the authors develop a multistep Q-learning method to solve the optimal output regulation problem for 2-degree-of-freedom helicopter. In [25], a based-Q function policy gradient adaptive dynamic programming algorithm is proposed to tackle the optimal control problem of general discrete-time nonlinear systems. In [18], the authors present a novel off-policy actor-critic neural network structure Q-learning method is developed to tackle the optimal output regulation problem of discrete-time systems. The authors in [24, 23] develop a critic-only Q-learning method. In [26], the authors develop a multistep Q-learning method to solve the optimal output regulation problem for 2-degree-of-freedom helicopter. In [25], a based-Q function policy gradient adaptive dynamic programming algorithm is proposed to tackle the optimal control problem of general discrete-time nonlinear systems. In [18], the authors present a novel off-policy actor-critic neural network structure Q-learning method is developed to tackle the optimal output regulation problem of discrete-time systems. The authors in [24, 23] develop a critic-only Q-learning method. In [26], the authors develop a multistep Q-learning method to solve the optimal output regulation problem for 2-degree-of-freedom helicopter. In [25], a based-Q function policy gradient adaptive dynamic programming algorithm is proposed to tackle the optimal control problem of general discrete-time nonlinear systems. In [18], the authors present a novel off-policy actor-critic neural network structure Q-learning method is developed to tackle the optimal output regulation problem of discrete-time systems. The authors in [24, 23] develop a critic-only Q-learning method. In [26], the authors develop a multistep Q-learning method to solve the optimal output regulation problem for 2-degree-of-freedom helicopter. In [25], a based-Q function policy gradient adaptive dynamic programming algorithm is proposed to tackle the optimal control problem of general discrete-time nonlinear systems. In [18], the authors present a novel off-policy actor-critic neural network structure Q-learning method is developed to tackle the optimal output regulation problem of discrete-time systems. The authors in [24, 23] develop a critic-only Q-learning method. In [26], the authors develop a multistep Q-learning method to solve the optimal output regulation problem for 2-degree-of-freedom helicopter. In [25], a based-Q function policy gradient adaptive dynamic programming algorithm is proposed to tackle the optimal control problem of general discrete-time nonlinear systems. In [18], the authors present a novel off-policy actor-critic neural network structure Q-learning method is developed to tackle the optimal output regulation problem of discrete-time systems. The authors in [24, 23] develop a critic-only Q-learning method. In [26], the authors develop a multistep Q-learning method to solve the optimal output regulation problem for 2-degree-of-freedom helicopter. In [25], a based-Q function policy gradient adaptive dynamic programming algorithm is proposed to tackle the optimal control problem of general discrete-time nonlinear systems. In [18], the authors present a novel off-policy actor-critic neural network structure Q-learning method is developed to tackle the optimal output regulation problem of discrete-time systems. The authors in [24, 23] develop a critic-only Q-learning method. In [26], the authors develop a multistep Q-learning method to solve the optimal output regulation problem for 2-degree-of-freedom helicopter. In [25], a based-Q function policy gradient adaptive dynamic programming algorithm is proposed to tackle the optimal control problem of general discrete-time nonlinear systems. In [18], the authors present a novel off-policy actor-critic neural network structure Q-learning method is developed to tackle the optimal output regulation problem of discrete-time systems. The authors in [24, 23] develop a critic-only Q-learning method. In [26], the authors develop a multistep Q-learning method to solve the optimal output regulation problem for 2-degree-of-freedom helicopter. In [25], a based-Q function policy gradient adaptive dynamic programming algorithm is proposed to tackle the optimal control problem of general discrete-time nonlinear systems. In [18], the authors present a novel off-policy actor-critic neural network structure Q-learning method is developed to tackle the optimal output regulation problem of discrete-time systems.
method. There are few studies for finite-horizon optimal control for completely unknown discrete-time linear systems. And, the application of reinforcement learning in finite-horizon optimal control is still in its infancy. All of these are the motivations of this paper.

The main innovations of this paper are summarized in the following two aspects. (1). A novel cyclic fixed-finite-horizon-based Q-learning algorithm is proposed. Terminal boundary condition is incorporated in the proposed algorithm. The collected learning data traverses the overall finite-horizon step $[0, N]$, and the robustness of the algorithm is thus improved. (2). The key of finite-time optimal control problem for discrete-time linear systems is to solve the time-varying RE equation, which is inherently challenging. In addition, completely unknown system dynamics bring additional challenges. However, these challenges have been well handled in this paper.

The structure of this paper is described as follows. In Section 2, we formulate the finite-horizon optimal control problem of discrete-time linear systems with completely unknown dynamics. In Section 3, the Q-learning formulation of finite-horizon optimal control for discrete-time linear systems is given. In Section 4, we introduce the cyclic fixed-finite-horizon-based Q-learning algorithm. In Section 5, two simulation examples are given to verify the effectiveness of our proposed cyclic fixed-finite-horizon-based Q-learning algorithm. Section 6 concludes this paper and gives the direction of future research.

2. Finite-horizon optimal control problem of discrete-time linear systems. In this section, the finite-horizon optimal control problem of discrete-time linear systems is formulated.

This paper considers the following time-invariant discrete-time linear systems

$$x(k + 1) = Ax(k) + Bu(k)$$  \hspace{1cm} (1)

where $x(k) \in \Omega$ denotes the system state and $u(k) \in \Omega_u$ denotes the system control input at time step $k$. $A \in \mathbb{R}^{n \times n}$ and $B \in \mathbb{R}^{n \times m}$ are constant matrices of suitable dimensions. $A$ and $B$ are assumed to be unknown in this study. For simplicity, $x(k)$ and $u(k)$ can be abbreviated as $x_k$ and $u_k$ respectively.

For the finite-horizon optimal control problem, the goal is to find an optimal control input sequence $u_k$, $k \in [0, 1, 2, \cdots, N - 1]$ which minimizes the following performance index function

$$J(k) = x_N^T Q_N x_N + \sum_{k=0}^{N-1} (x_k^T Q x_k + u_k^T R u_k)$$  \hspace{1cm} (2)

where $Q = Q^T \geq 0 \in \mathbb{R}^{n \times n}$, $R = R^T > 0 \in \mathbb{R}^{m \times m}$ and $Q_N = Q_N^T \geq 0 \in \mathbb{R}^{n \times n}$ are known constant matrices. $x_N^T Q_N x_N$ is the terminal boundary constraints.

**Definition 2.1.** (Finite-horizon admissible control sequence) For the discrete-time linear systems (1), $x(k) \in \Omega$, a control input $u_k \in \Phi(\Omega) \subseteq \Omega_u$, $k \in [0, 1, 2, \cdots, N - 1]$ is defined to be finite-horizon admissible with respect to performance index function (2) if $u$ is continuous on $\Omega$ and $J(u_k, x_k)$ is finite, $\forall x_k \in \Omega$. [25]

**Assumption 1.** [31] $(A, B)$ is controllable and $(\sqrt{Q}, A)$ is observable.

**Assumption 2.** All the state variables of discrete-time linear systems (1) are assumed to be available.
As we all know, according to the traditional optimal control theory [16], the optimal control input $u^*$ can be found as following

$$ u_k^* = -K_k x_k = -(B^T P_{k+1} B + R)^{-1} B^T P_{k+1} A x_k $$ (3)

where $P_{k+1}$ is the solution to the following Riccati equation

$$ P_k = A^T [P_{k+1} - P_k B (B^T P_{k+1} B + R)^{-1} B^T P_{k+1}] A + Q $$ (4)

where $P_N = Q_N$.

As we all know, equation (4) is difficult to solve directly, and it is required that $A$ and $B$ are known. In order to deal with this dilemma, a Q-learning-based algorithm will be introduced in the next.

3. Q-learning for finite-horizon optimal control of discrete-time linear systems. This section will main present the Q-learning formulation for finite-horizon optimal control of discrete-time linear systems.

Based on the defined performance index function (2), we define the value function or cost function as

$$ V(x_{k,N-k}) = x_N^T Q_N x_N + \sum_{i=k}^{N-1} (x_i^T Q x_i + u_i^T R u_i) $$ (5)

In line with [16], the cost function or value function can also be represented as follow

$$ V(x_{k,N-k}) = x_k^T P_k x_k $$ (6)

According to (3), (4) and (5), we can obtain the following Bellman equation

$$ V^*(x_{k,N-k}) = x_k^T Q x_k + u_k^T R u_k + V^*(x_{k+1,N-(k+1)}$$ (7)

where $V^*(x_{k,N-k})$ and $V^*(x_{k+1,N-(k+1)})$ can be respectively abbreviated as $V_k^*$ and $V_{k+1}^*$.

Based on equation (6) and Bellman equation (7), we can define the Q-function associated with $u_k$ as [15]

$$ Q(x_k, u_k, N-k) = x_k^T Q x_k + u_k^T R u_k + x_{k+1}^T P_{k+1} x_{k+1} $$ (8)

Substituting system (1) into (8), we can obtain

$$ Q(x_k, u_k) = \begin{bmatrix} x_k^T \\ u_k^T \end{bmatrix} \begin{bmatrix} Q + A^T P_{k+1} A & A^T P_{k+1} B \\ B^T P_{k+1} A & R + B^T P_{k+1} B \end{bmatrix} \begin{bmatrix} x_k \\ u_k \end{bmatrix} $$

$$ = \zeta \begin{bmatrix} \hat{Q}_{kxx} & \hat{Q}_{kxu} \\ \hat{Q}_{kux} & \hat{Q}_{kuu} \end{bmatrix} \zeta^T = \zeta \hat{Q}_k \zeta^T $$ (9)

where $\zeta = [x_k; u_k]^T \in \mathbb{R}^{1\times(n+m)}$, $\hat{Q}_{kxx} = Q + A^T P_{k+1} A \in \mathbb{R}^{n\times n}$, $\hat{Q}_{kxu} = A^T P_{k+1} B \in \mathbb{R}^{n\times m}$, $\hat{Q}_{kux} = B^T P_{k+1} A \in \mathbb{R}^{m\times n}$, $\hat{Q}_{kuu} = R + B^T P_{k+1} B \in \mathbb{R}^{m\times m}$ and $\hat{Q}_k = \begin{bmatrix} \hat{Q}_{kxx} & \hat{Q}_{kxu} \\ \hat{Q}_{kux} & \hat{Q}_{kuu} \end{bmatrix} \in \mathbb{R}^{(n+m)\times(n+m)}$ is a time-dependent matrix.

**Lemma 3.1.** If $u_k$ in Q-function (9) given by (3), the Q-function (9) have the same value as the Bellman equation (7), that is $V^*(x_k) = Q^*(x_k, u_k^*) = \min_{u_k} Q(x_k, u_k)$.

A detailed proof of Lemma 1 can be found in references [31] and [15].

Then we can obtain the optimal control input sequence $u_k^*$ by the following method without using the system dynamics $A$ and $B$

$$ u_k^* = \arg\min_{u_k} Q(x_k, u_k) = -\hat{Q}_{kux} \hat{Q}_{kux} x_k $$ (10)
and the following equation is satisfied
\[ Q^*(x_k, u_k^*) = x_k^T Q x_k + u_k^T R u_k + Q^*(x_{k+1}, u_{k+1}^*) \] (11)

4. Cyclic fixed-finite-horizon-based Q-learning algorithm. This section will main develop the cyclic fixed-finite-horizon-based Q-learning algorithm for finite-horizon optimal control of discrete-time linear systems.

For the convenience of representation, we introduce Kronecker product, the Q-function (9) can be rewritten as
\[ Q^*(x_k, u_k^*) = \zeta Q_k \zeta^T := \text{vec}(\hat{Q}_k)^T (\zeta^T \otimes \zeta^T) \]

where let
\[ \text{vec}(\hat{Q}_k) = [\hat{Q}_{k11}, 2\hat{Q}_{k12}, \ldots, 2\hat{Q}_{k1(n+m)}, \hat{Q}_{k22}, 2\hat{Q}_{k23}, \ldots, 2\hat{Q}_{k2(n+m)}, \ldots, \hat{Q}_{k(n+m)(n+m)}] \in \mathbb{R}^{(m+n)(m+n+1) \times 1} \]

then we can represent \((\zeta^T \otimes \zeta^T)\) in a compact form as follow
\[
\text{Compact}(\zeta^T \otimes \zeta^T) = [x_1^T, x_1 x_2, \ldots, x_1 x_n, x_1 u_1, \ldots, x_1 u_m, \\
x_2^T, x_2 x_3, \ldots, x_2 x_n, x_2 u_1, \ldots, x_2 u_m, \ldots, x_n^T, x_n u_1, \ldots, x_n u_m, \\
u_1^T, u_1 u_2, \ldots, u_1 u_m, \ldots, u_2^T, u_2 u_3, \ldots, u_2 u_m, \ldots, u_m^T] \in \mathbb{R}^{(m+n)(m+n+1) \times 1}
\]

where \(\hat{Q}_{kij}(i, j = 1, \ldots, m + n)\) represents the \(i\)-th row and the \(j\)-th column element of the matrix \(\hat{Q}_k\) and define \(Q_k = \text{vec}^{-1}[\text{vec}(\hat{Q}_k)]\).

Then (12) can be represented as
\[ Q^*(x_k, u_k^*) = \text{vec}(\hat{Q}_k)^T \text{Compact}[\zeta^T \otimes \zeta^T] \]

(13)

Since \(\hat{Q}_k\) is a time-dependent matrix, \(\text{vec}(\hat{Q}_k)^T\) is thus also time-varying. Then, we can rewrite \(\text{vec}(\hat{Q}_k)^T\) as follow
\[ \text{vec}(\hat{Q}_k)^T = W^T \varphi(N, k) \]

(14)

where \(W = [w_1, w_2, \ldots, w_{(n+m)(n+m+1)/2}]^T \in \mathbb{R}^{(n+m)(n+m+1)/2 \times 1}\) is the ideal parameters. The ideal \(W\) is unknown, we use \(\hat{W}\) as its estimated value. \(\varphi(N, k) \in \mathbb{R}^{(n+m)(n+m+1)/2 \times 1}\) is time-dependent function matrix. Then (13) can be estimated as follow
\[ \hat{Q}^*(x_k, u_k^*) = \hat{W}^T \varphi(N, k) \text{Compact}[\zeta^T \otimes \zeta^T] \]

(15)

According to (15), (11) can be rewritten as follow
\[ e = \hat{W}^T \varphi(N, k + 1) \text{Compact} [\zeta^T \otimes \zeta^T]_{k+1} \]
\[
-\hat{W}^T \varphi(N, k) \text{Compact} [\zeta^T \otimes \zeta^T]_k \\
+ x_k^T Q x_k + u_k^T R u_k
\]

(16)

where \(e\) is the Bellman estimation error.

Assumption 3. It is assumed that \(W^T \varphi(N, N) = G_N \in \mathbb{R}^{1 \times (n+m)(n+m+1)/2}\) equivalent to \(Q_N \in \mathbb{R}^{n \times n}\) in the terminal boundary constraints is given in advance.
In finite-horizon optimal control, based on equation (16), in the finite-time step 
\[0 \rightarrow N\], we have

\[e_1 = \hat{W}^T \varphi(N, 1) \text{Compact}[(\zeta^T \otimes \zeta^T)_1] \]
\[- \hat{W}^T \varphi(N, 0) \text{Compact}[(\zeta^T \otimes \zeta^T)_0] + x_0^T Q x_0 + u_0^T R u_0 \]  
\[e_2 = \hat{W}^T \varphi(N, 2) \text{Compact}[(\zeta^T \otimes \zeta^T)_2] \]
\[- \hat{W}^T \varphi(N, 1) \text{Compact}[(\zeta^T \otimes \zeta^T)_1] + x_1^T Q x_1 + u_1^T R u_1 \]  
\[e_3 = \hat{W}^T \varphi(N, 3) \text{Compact}[(\zeta^T \otimes \zeta^T)_3] \]
\[- \hat{W}^T \varphi(N, 2) \text{Compact}[(\zeta^T \otimes \zeta^T)_2] + x_2^T Q x_2 + u_2^T R u_2 \]  
\[\vdots \]
\[e_N = \hat{W}^T \varphi(N, N) \text{Compact}[(\zeta^T \otimes \zeta^T)_N] \]
\[- \hat{W}^T \varphi(N, N - 1) \text{Compact}[(\zeta^T \otimes \zeta^T)_{N-1}] + x_{N-1}^T Q x_{N-1} + u_{N-1}^T R u_{N-1} \]  
\[e_{\text{terminal}} = \hat{W}^T \varphi(N, N) - G_N \]  

where \(e_1, e_2, \ldots, e_N\) are the Bellman approximation errors and \(e_{\text{terminal}}\) is the boundary error. In order to minimize the Bellman approximation errors \(e_1, e_2, \ldots, e_N\) and the boundary error \(e_{\text{terminal}}\), that is to guarantee that \(e_1, e_2, \ldots, e_N, e_{\text{terminal}} \rightarrow 0\), \(\hat{W} \rightarrow W\), a least-squares method is thus adopted. To this end, we define the following matrices

\[
\Phi = \begin{bmatrix}
\varphi(N, 1) & 0 \\
\varphi(N, 2) & 0 \\
\vdots & \vdots \\
\varphi(N, N) & 0 \\
\varphi(N, N) & \varphi(N, N)
\end{bmatrix}
\]
\[
E = \begin{bmatrix}
e_1 \\
e_2 \\
\vdots \\
e_N \\
e_{\text{terminal}}
\end{bmatrix}^T
\]
\[
\Theta = \begin{bmatrix}
x_0^T Q x_0 + u_0^T R u_0 \\
x_1^T Q x_1 + u_1^T R u_1 \\
\vdots \\
x_{N-1}^T Q x_{N-1} + u_{N-1}^T R u_{N-1} \\
-G_N
\end{bmatrix}
\]

and

\[
\varphi(N, l) \text{Compact}[(\zeta^T \otimes \zeta^T)_l] \]
\[- \varphi(N, l-1) \text{Compact}[(\zeta^T \otimes \zeta^T)_{l-1}] \]

where \(l = 1, 2, \ldots, N\).

The least-squares solution to equation (16) is then equal to

\[
\hat{W} = -(\Phi \Phi^T)^{-1} \Phi \Theta
\]

According equations (17), (18), (19), (20), (21), the cyclic fixed-finite-horizon-based Q-learning algorithm is then given as follows:
Algorithm 1. (Cyclic fixed-finite-horizon-based Q-learning algorithm)

Phase 1: (Data collection at finite time step \([0 \ N]\))

Step 1. (Initialization): Set initial system state \(x_0, W_0, N, G_N, j_{max}, h(0)\), then \(Q_j = vec^{-1}[\phi^T(N,0)W_j]\), initial admissible control \(u_k^j = -Q_{juu}Q_{jux}x_k + e_h(L)\).

Where \(e_h(L)\) is excitation signal associated with \(h(N)\), \(j = 0\).

Step 2. (Data collection): Apply \(u_k^j\) to the system \((1)\) and collect data \(L\) times at finite time step \([0 \ N]\) based on equations \((17), (18), (19), (20)\). And compute \(\Phi_j = [\Phi^1, \Phi^2, \cdots, \Phi^L]^T, \Theta_j = [\Theta^1, \Theta^2, \cdots, \Theta^L]^T\). \(\Phi^c, \Theta^c, \kappa = 1, 2, \cdots, L\) denote the data \((\Phi\) and \(\Theta\)) collected at the \(\kappa\)–th time.

Phase 2: (parameters update)

Step 3. (\(\hat{W}\) update):

\[
\hat{W}_{j+1} = -(\Phi_j\Phi_j^T)^{-1}\Phi_j\Theta_j
\]  

(22)

Step 4. Stop if \(\|\hat{W}_{j+1} - \hat{W}_j\| \leq \varepsilon\) is satisfied or \(j = j_{max}\), else \(j = j + 1\) and go to Step 1. Where \(\varepsilon\) is a small positive number.

Step 5. Use \(u = -\hat{Q}_{j+1}Q_{j+1}x_k\) as the approximated optimal control input.

The flow chart of Algorithm 1 is depicted in Fig.1.

Remark 1. It is easy to see that Algorithm 1 does not require known system dynamics \(A\) and \(B\).

Remark 2. In initial admissible control step of Phase 1, initial system state \(x_0\) can be randomly taken from a compact set \(\Omega\) in each iteration. When each iteration randomly selects the initial state for learning, it is necessary to ensure that enough points in the compact set are traversed.

Remark 3. The computational complexity of Algorithm 1 will increase with the increase of \(N\).

Remark 4. The cyclic means that data are re-collected at time interval \([0 \ N]\) during each iteration.

Theorem 4.1. To guarantee the convergence of Algorithm 1, for each data collection in Step 2, the following condition must be satisfied \([10]\)

\[
\text{rank}(\Phi_j) = \frac{(n + m)(n + m + 1)}{2}
\]  

(23)

and \(L\) satisfies

\[
L \geq \frac{(n + m)(n + m + 1)}{2(N + 1)}
\]  

(24)

Proof. Since the number of independent elements in \(\hat{W}\) is \(\frac{(n + m)(n + m + 1)}{2}\), in order to ensure that the solution to equation \((21)\) is unique, according to the Lemma 6 in \([10]\), it is easy to obtain that equation \((23)\) should be satisfied. According to data collection step 2, the data are collected at \(N + 1\) points over the finite-time step \([0 \ N]\) for \(L\) times, based on equation \((23)\), there must be has

\[
L(N + 1) \geq \frac{(n + m)(n + m + 1)}{2}
\]

According to the above inequality, it is easy to obtain

\[
L \geq \frac{(n + m)(n + m + 1)}{2(N + 1)}
\]
Figure 1. The flow chart of Algorithm 1.

The proof is thus complete.

**Remark 5.** In applications, the condition (23) is easily satisfied by properly adjusting $L$.

5. **Simulation results.** In this section, we present two simulation examples to verify the effectiveness of our proposed cyclic fixed-finite-horizon-based Q-learning algorithm.

**Example 1:** In this example, we consider the following discrete-time linear systems from reference [42]

$$x(k + 1) = \begin{bmatrix} 0.8 & 1 \\ 0 & 0.6 \end{bmatrix} x(k) + \begin{bmatrix} 1 \\ 0.5 \end{bmatrix} u(k)$$

Let $Q = \begin{bmatrix} 10 & 0 \\ 0 & 10 \end{bmatrix}$, $R = 1$, $N = 15$. The terminal boundary constraints $G_N = [1.64, 2.88, -1.6, -0.0002, -4.88, 3.6]$. In this example, we use Remark 2 to train Algorithm 1, which means that initial system state $x_0$ is randomly selected from a
compact set $\Omega := \{-1 \leq x_1, x_2 \leq 1\}$ in each iteration. The time-dependent function matrix $\varphi(N, k)$ is selected as follows

$$\varphi(N, k) = 10000. \begin{bmatrix} \tanh(N - k), \tanh(N - k)^2, \ldots, \tanh(N-k)^{(m+n)(m+n+1)/2}; \\
\tanh(N - k)^2, \ldots, \tanh(N-k)^{(m+n)(m+n+1)/2}, \tanh(N - k); \\
\vdots & \vdots & \vdots & \vdots \\
\tanh(N-k)^{(m+n)(m+n+1)/2}, \ldots, \tanh(N-k)^{(m+n)(m+n+1)/2-1} \end{bmatrix}$$

Then, the simulation results are presented in the following. Initial system state $x_0$ randomly selected from $\Omega := \{-1 \leq x_1, x_2 \leq 1\}$ are presented in Fig.2. Fig.3 shows the convergence process of $\hat{W}$. After the second iteration, $\hat{W}$ has converged and $\hat{W} = [-0.5567, 2.6900, -2.7724, -0.7135, 1.5957, -0.2423]^T$. The system states trajectories are displayed in Fig.4. The optimal control input is depicted in Fig.5.

**Figure 2.** Initial system state $x_0$ are randomly selected from a compact set $\Omega := \{-1 \leq x_1, x_2 \leq 1\}$.

**Figure 3.** The convergence process of $\hat{W}$.

**Example 2:** In this example, we consider the following discretized F-16 aircraft plant systems from reference [43]

$$x(k+1) = \begin{bmatrix} 0.9065 & 0.0816 & -0.0009 \\
0.0741 & 0.9012 & -0.0159 \\
0 & 0 & 0.9048 \end{bmatrix} x(k) + \begin{bmatrix} 0 \\
-0.008 \\
0.0952 \end{bmatrix} u(k)$$
Let $Q = \begin{bmatrix} 10 & 0 & 0 \\ 0 & 10 & 0 \\ 0 & 0 & 10 \end{bmatrix}, R = 10, N = 30$. The terminal boundary constraints $G_N = [1.8272, 0.2816, -0.002, -0.0022, 1.8188, -0.0128, -0.0174, 1.0176, 0.2303, 1.7524]^T$

The time-dependent function matrix $\varphi(N, k)$ is the same as Example 1. In this example, we use the fixed initial state $x_0 = [1 \ -1 \ 0.5]^T$ to train Algorithm 1. The simulation results are presented in the following. Fig.6 shows the convergence process of $\hat{W}$. After the second iteration, $\hat{W}$ has converged and $\hat{W} = [-6.8483, 9.3449, 2.5021, 0.1835, 2.8256, 5.6899, 4.6236, -0.6624, -13.6104, -3.9848]^T$

The system states trajectories are displayed in Fig.7. The optimal control input is depicted in Fig.8. Therefore, the simulation results verify the effectiveness of our designed cyclic fixed-finite-horizon-based Q-learning algorithm.
6. **Conclusion.** This paper has investigated the finite-horizon optimal control problem of discrete-time linear systems with completely unknown dynamics. Dealing with this problem is equivalent to solving a time-dependent Riccati equation. For relaxation dependence on the system dynamics, Q-learning technique is introduced in this study. A cyclic fixed-finite-horizon-based Q-learning algorithm is thus developed to cope with the optimal control problem. Finally, the effectiveness of the developed algorithm is verified by two simulation examples. In future work, we will extend the results of this study to finite-horizon optimal tracking control problems.

**REFERENCES**

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

[2] D. P. Bertsekas and J. N. Tsitsiklis, *Neuro-Dynamic Programming*, 1st edition, Athena Scientific, Belmont, 1996.

[3] D. P. Bertsekas, Value and policy iterations in optimal control and adaptive dynamic programming, *IEEE Transactions on Neural Networks and Learning Systems*, 28 (2017), 500–509.

[4] X. S. Chen, X. Li and F. H. Yi, Optimal stopping investment with non-smooth utility over an infinite time horizon, *Journal of Industrial & Management Optimization*, 15 (2019), 81–96.

[5] Y. F. Chen and Y. G. Zhu, Indefinite LQ optimal control with process state inequality constraints for discrete-time uncertain systems, *Journal of Industrial & Management Optimization*, 14 (2018), 913–930.

[6] T. Cheng, F. L. Lewis and M. Abu-Khalaf, A neural network solution for fixed-final time optimal control of nonlinear systems, *Automatica*, 43 (2007), 482–490.

[7] M. G. Gan, J. G. Zhao and C. Zhang, Extended adaptive optimal control of linear systems with unknown dynamics using adaptive dynamic programming, *Asian J Control*, (2019), 1–10.

[8] W. N. Gao, Y. Jiang, Z. P. Jiang and Y. T. Chai, Output-feedback adaptive optimal control of interconnected systems based on robust adaptive dynamic programming, *Automatica*, 72 (2016), 37–45.
[9] A. Heydari and S. N. Balakrishnan, Finite-horizon control-constrained nonlinear optimal control using single network adaptive critics, IEEE Transactions on Neural Networks and Learning Systems, 24 (2013), 145–157.

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

[11] Y. Jiang and Z. P. Jiang, Global adaptive dynamic programming for continuous-time nonlinear systems, IEEE Trans on Automatic Control, 60 (2015), 2917–2929.

[12] W. K. Jong, J. P. Byung, Y. Haeun, H. L. Jay and M. L. Jong, Deep reinforcement learning based finite-horizon optimal tracking control for nonlinear systems, in International Federation Automatic Control, (2018), 257–262.

[13] R. Kamalapurkar, P. Walters and W. E. Dixon, Model-based reinforcement learning for approximate optimal control, in Reinforcement Learning for Optimal Feedback Control, Communications and Control Engineering. Springer, Cham, (2018), 99–148.

[14] J. Y. Lee, B. P. Jin and Y. H. Chio, Integral Q-learning and explorized policy iteration for adaptive optimal control of continuous-time linear systems, Automatica, 48 (2012), 2850–2859.

[15] F. L. Lewis and D. R. Liu, Reinforcement learning and approximate dynamic programming for feedback control, IEEE Circuits and Systems Magazine, 9 (2015), 32–50.

[16] F. L. Levis, D. L. Vrabie and V. L. Syrmos, Optimal Control, 3rd edition, John Wiley and Sons, Hoboken, 2015.

[17] J. N. Li, T. Y. Chai, F. L. Lewis, Z. T. Ding and Y. Jiang, Off-policy interleaved Q-learning: optimal control for affine nonlinear discrete-time systems, IEEE Transactions on Neural Networks and Learning Systems, 30 (2019), 1308–1320.

[18] J. N. Li, T. Y. Chai, F. L. Lewis, J. L. Fan, Z. T. Ding and J. L. Ding, Off-policy Q-learning: Set-point design for optimizing dual-rate rougher flotation operational processes, IEEE Transactions on Industrial Electronics, 65 (2018), 4092–4102.

[19] X. X. Li, Z. H. Peng, L. Liang and W. Z. Zha, Policy iteration based Q-learning for linear nonzero-sum quadratic differential games, Science China(Information Sciences), 62 (2019), 195–213.

[20] Q. Lin, Q. L. Wei and D. R. Liu, A novel optimal tracking control scheme for a class of discrete-time nonlinear systems using generalised policy iteration adaptive dynamic programming algorithm, International Journal of Systems Science, 48 (2017), 525–534.

[21] H. L. Liu and Q. X. Zhu, New forms of Riccati equations and the further results of the optimal control for linear discrete-time systems, International Journal of Control, Automation, and Systems, 12 (2014), 1160–1166.

[22] B. Luo, D. R. Liu, T. W. Huang and D. Wang, Data-based approximate policy iteration for nonlinear continuous-time optimal control design, Automatica, 50 (2014), 3281–3290.

[23] B. Luo, D. R. Liu, T. W. Huang and D. Wang, Model-free optimal tracking control via critic-only Q-learning, IEEE Transactions on Neural Networks and Learning Systems, 27 (2016), 2134–2144.

[24] B. Luo, D. R. Liu and H. N. Wu, Adaptive constrained optimal control design for data-based nonlinear discrete-time systems with critic-only structure, IEEE Transactions on Neural Networks and Learning Systems, 29 (2018), 2099–2111.

[25] B. Luo, D. R. Liu, H. N. Wu, D. Wang and F. L. Lewis, Policy gradient adaptive dynamic programming for data-based optimal control, IEEE Transactions on Cybernetics, 47 (2017), 3341–3354.

[26] B. Luo, H. N. Wu and T. W. Huang, Optimal output regulation for model-free quanser helicopter with multistep Q-learning, IEEE Transactions on Industrial Electronics, 65 (2018), 4953–4961.

[27] B. Luo, Y. Yang and D. R. Liu, Adaptive Q-Learning for data-based optimal output regulation with experience replay, IEEE Transactions on Cybernetics, 48 (2018), 3337–3348.

[28] Y. F. Lv, N. Jing, Q. M. Yang, X. Wu and Y. Guo, Online adaptive optimal control for continuous-time nonlinear systems with completely unknown dynamics, International Journal of Control, 89 (2016), 99–112.

[29] W. B. Powell, Approximate Dynamic Programming: Solving the Curses of Dimensionality, 2nd edition, Wiley Princeton, New Jersey, 2011.

[30] A. Sahoo, H. Xu and S. Jagannathan, Approximate optimal control of affine nonlinear continuous-time systems using event-sampled neurodynamic programming, IEEE Transactions on Neural Networks and Learning Systems, 28 (2017), 639–652.
[31] K. G. Vamvoudakis, Q-learning for continuous-time linear systems: A model-free infinite horizon optimal control approach, Systems and Control Letters, 100 (2017), 14–20.

[32] K. G. Vamvoudakis and F. L. Lewis, Online actor-critic algorithm to solve the continuous-time infinite horizon optimal control problem, Automatica, 46 (2010), 878–888.

[33] D. Wang, D. R. Liu and Q. L. Wei, Finite-horizon neuro-optimal tracking control for a class of discrete-time nonlinear systems using adaptive dynamic programming approach, Neurocomputing, 78 (2012), 14–22.

[34] F. Y. Wang, N. Jin, D. R. Liu and Q. L. Wei, Adaptive dynamic programming for finite-horizon optimal control of discrete-time nonlinear systems with ε–error bound, IEEE Transactions on Neural Networks, 22 (2011), 24–36.

[35] Q. L. Wei and D. R. Liu, A novel policy iteration based deterministic Q-learning for discrete-time nonlinear systems, Science China (Information Sciences), 58 (2015), 1–15.

[36] Q. L. Wei, D. R. Liu and H. Q. Lin, Value iteration adaptive dynamic programming for optimal control of discrete-time nonlinear systems, IEEE Trans Cybern, 46 (2016), 840–853.

[37] Y. Wu, Z. H. Yuan and Y. P. Wu, Optimal tracking control for networked control systems with random time delays and packet dropouts, Journal of Industrial & Management Optimization, 11 (2015), 1343–1354.

[38] H. G. Zhang, J. He, Y. H. Luo and G. Y. Xiao, Data-driven optimal consensus control for discrete-time multi-agent systems with unknown dynamics using reinforcement learning method, IEEE Transactions on Industrial Electronics, 64 (2017), 4091–4100.

[39] H. G. Zhang, Q. L. Wei and Y. H. Luo, A novel infinite-time optimal tracking control scheme for a class of discrete-time nonlinear systems via the greedy HDP iteration algorithm, IEEE Transactions on Systems Man and Cybernetics Part B, 38 (2008), 937–942.

[40] Q. C. Zhang, D. B. Zhao and D. Wang, Event-based robust control for uncertain nonlinear systems using adaptive dynamic programming, IEEE Transactions on Neural Networks and Learning Systems, 29 (2018), 37–50.

[41] J. G. Zhao, M. G. Gan and C. Zhang, Event-triggered $H_\infty$ optimal control for continuous-time nonlinear systems using neurodynamic programming, Neurocomputing, 360 (2019), 14–24.

[42] Q. M. Zhao, Finite-horizon Optimal Control of Linear and a Class of Nonlinear Systems, Ph.D thesis, Missouri University of Science and Technology, 2013.

[43] Q. M. Zhao, X. Hao and J. Sarangapani, Finite-horizon near optimal adaptive control of uncertain linear discrete-time systems, Optimal Control Applications and Methods, 36 (2016), 853–872.

[44] X. N. Zhong, H. B. He, D. Wang and Z. Ni, Model-free adaptive optimal control for unknown nonlinear zero-sum differential game, IEEE Transactions on Cybernetics, 48 (2018), 1633–1646.

[45] Q. X. Zhu and G. M. Xie, Finite-horizon optimal control of discrete-time switched linear systems, Mathematical Problems in Engineering, 2012 (2012), 1–12.

Received March 2019; revised September 2019.

E-mail address: zhaojingang521@126.com
E-mail address: zhangchibit@foxmail.com