AIMS Mathematics, 8(5): 10249–10265.
DOI: 10.3934/math.2023519
Received: 13 December 2022
Revised: 13 February 2023
Accepted: 20 February 2023
Published: 27 February 2023

http://www.aimspress.com/journal/Math

Research article

Stochastic linear quadratic optimal tracking control for discrete-time systems with delays based on Q-learning algorithm

Xufeng Tan[1], Yuan Li[1,*] and Yang Liu[2]

[1] School of Science, Shenyang University of Technology, Shenyang 110870, China
[2] School of Electrical and Electronic Engineering, Shenyang University of Technology, Shenyang 110870, China

* **Correspondence:** Email: syliyuan@sut.edu.cn.

Abstract: In this paper, a reinforcement Q-learning method based on value iteration (VI) is proposed for a class of model-free stochastic linear quadratic (SLQ) optimal tracking problem with time delay. Compared with the traditional reinforcement learning method, Q-learning method avoids the need for accurate system model. Firstly, the delay operator is introduced to construct a novel augmented system composed of the original system and the command generator. Secondly, the SLQ optimal tracking problem is transformed into a deterministic one by system transformation and the corresponding Q function of SLQ optimal tracking control is derived. Based on this, Q-learning algorithm is proposed and its convergence is proved. Finally, a simulation example shows the effectiveness of the proposed algorithm.

Keywords: reinforcement Q-learning; value iterative; model-free; stochastic linear quadratic optimal tracking; time delay; deterministic system
Mathematics Subject Classification: 93E20, 93C05, 93C41, 93C55

1. Introduction

It is well known that the optimal tracking control (OTC) problem plays an important role in the field of optimal control and develops fast in applications [1–4]. The goal of OTC problem is to design a controller, which can make the output of the system track the reference trajectory by minimizing the cost function. Traditional OTC problem is realized by feedback linearization [5] and object inversion [6], but this usually requires complex mathematical analysis. As for the linear quadratic tracking (LQT) problem, the traditional method of LQT problem is to solve the algebraic Riccati equation (ARE) and the noncausal difference equation. However, these methods require accurate system model [7]. In practical situations, the system parameters are partially unknown or completely

unknown, so it is impossible to be realized by traditional methods.

The key to the OTC problem is to solve Hamilton-Jacobi-Bellman (HJB) equation. However, HJB equation involves solving difference or differential equations, so it is difficult to solve it. Although dynamic programming has always been an effective method to solve the HJB equation, it is not feasible in the calculation of large dimensions because of "the curse of dimensionality". To solve the solution of the HJB equation, adaptive dynamic programming (ADP) algorithms have been widely used and developed. In [8], a policy iteration (PI) scheme was adopted to approximate the optimal control for the partly unknown continuous-time systems. In [9], B. Kiumarsi solves the LQT problem online only by measuring the input, output, and reference trajectory data of the system. In [10], a Q-learning method was proposed to calculate the optimal control, only relying on system parameters and command generators.

In recent years, stochastic system control theory has become the focus of optimal control theory because of its academic difficulty and wide application, especially the model-free SLQ optimal tracking problem has attracted more and more attention [11–15]. In [14], ADP algorithm based on neural networks is proposed to solve the model-free SLQ optimal tracking control problem. In addition, the Q-learning algorithm is used to solve the model-free SLQ optimal tracking control problem in [15]. For all we know, there seem to be many research results on the model-free SLQ optimal tracking problem based on ADP algorithm, but the SLQ optimal tracking problem with delays has received little attention. Time delay [16] is an important factor that cannot be ignored. It exists in many practical systems, such as industrial processes, power grids, chemical reactions, and so on [17–20]. However, in these methods [11–15], the influence of time delay on the system is neglected. If the time delay is ignored, it will affect the control effect and even make the system divergence. The method proposed in [16] takes into account the time delay but ignores the influence of stochastic disturbance disturbances on the system. As far as we know, there is no research on the optimal tracking problem of stochastic linear systems with delays. Therefore, how to use ADP algorithm to deal with the model-free SLQ optimal tracking control problem has important practical significance. This is the motivation we study in this paper.

The main contributions of this paper include:

(1) For stochastic linear system, this paper proposes Q-learning to model-free solve SLQ optimal tracking control problem with delays for the first time, which enhances the practicability of ADP algorithm in tracking problems.

(2) By introducing the delay factor, the influence of delays on the subsequent algorithm can be effectively eliminated.

(3) In this paper, the Q-learning algorithm is used to solve the model-free SLQ optimal tracking control problem with delays. Compared with other methods which need accurate system model to obtain the optimal control, this method makes full use of the online system state information to obtain the optimal control and avoids solving augmented stochastic algebraic equation (SAE).

The structure of this paper is organized as follows. In section 2, we give the problem formulation and conversion. In section 3, we derive the Q-learning algorithm and prove its convergence. In section 4, we give the implementation steps of Q-learning algorithm. In section 5, a simulation example is given to verify the effectiveness of the algorithm. In section 6, the conclusion is given.

2. Problem formulation and transformation

2.1. Problem formulation

Consider the following linear stochastic systems with delays

$$\begin{aligned} x_{k+1} &= Ax_k + A_d x_{k-d} + Bu_k + B_d u_{k-d} + (Cx_k + C_d x_{k-d} + Du_k + D_d u_{k-d})\omega_k, \\ y_k &= Ex_k + E_d x_{k-d} \end{aligned} \tag{2.1}$$

where $x_k \in \mathcal{R}^n$ is the system state vector, $u_k \in \mathcal{R}^m$ is the control input vector, $y_k \in \mathcal{R}^q$ is the system output, while x_{k-d}, u_{k-d} and y_{k-d} are the delay variables with delay index $d \in N$. $A \in \mathcal{R}^{n\times n}$, $B \in \mathcal{R}^{n\times m}$, $C \in \mathcal{R}^{n\times n}$, $D \in \mathcal{R}^{n\times m}$, $E \in \mathcal{R}^{q\times n}$ are given constant, $A_d \in \mathcal{R}^{n\times n}$, $B_d \in \mathcal{R}^{n\times m}$, $C_d \in \mathcal{R}^{n\times n}$, $D_d \in \mathcal{R}^{n\times m}$, $E_d \in \mathcal{R}^{q\times n}$ are their corresponding delay dynamics matrices. One-dimensional stochastic disturbance sequence ω_k is defined on the given probability space $(\Omega, \mathcal{F}, \mathcal{P}, \mathcal{F}_k)$, and meets the following condition $E(\omega_k \mid \mathcal{F}_k) = 0$, $E(\omega_k^2 \mid \mathcal{F}_k) = 1$. The initial state x_0 is irrelevant with ω_k.

Assume the reference trajectory of SLQ optimal tracking control is generated by a command generator

$$r_{k+1} = Fr_k \tag{2.2}$$

where $r_k \in \mathcal{R}^q$ represents the reference system trajectory, and F is the constant matrix.

The tracking error can be expressed as

$$e_k = y_k - r_k \tag{2.3}$$

where r_k is the reference trajectory.

The goal of the SLQ optimal tracking problem with delays is to design an optimal controller, which can not only ensure that the output of the target system track the reference trajectory stably, but also minimize the cost function. The cost function is denoted as

$$J(x_k, r_k, u_k) = E\sum_{i=k}^{\infty} U_i(x_i, x_{i-d}, u_i) \tag{2.4}$$

where $U_i(x_i, x_{i-d}, u_i) = (y_i - r_i)^T O(y_i - r_i) + u_i^T \mathcal{R} u_i + u_{i-d}^T \mathcal{R}_d u_{i-d}$ is the utility function. $O = O^T \in \mathcal{R}^{q\times q} \geq 0$, $R = R^T \in \mathcal{R}^{m\times m} \geq 0$, $R_d = R_d^T \in \mathcal{R}^{m\times m} \geq 0$ are the constant matrices.

Only when F is Hurwitz can the cost function (2.4) be used, that is, the reference trajectory system is required to be asymptotically stable. If the reference trajectory does not tend to zero with time delay, then the cost function (2.4) will be unbounded. In practice, this condition is difficult to achieve. Therefore, a discount factor γ is introduced into the cost function (2.4) to relax this restriction. Based on (2.4), the cost function with discount factor is redefined as

$$\begin{aligned} J(x_k, r_k, u_k) &= E\sum_{i=k}^{\infty} \gamma^{i-k} U_i(x_i, x_{i-d}, u_i) \\ &= E\sum_{i=k}^{\infty} \gamma^{i-k} (y_i - r_i)^T O(y_i - r_i) + u_i^T \mathcal{R} u_i + u_{i-d}^T \mathcal{R}_d u_{i-d} \end{aligned} \tag{2.5}$$

where $0 < \gamma \leq 1$ is the discount factor.

Definition 1 ([21]). u_k is called mean-square stabilizing at e_0 if there exists a linear feedback form of u_k for every initial state e_0 satisfies $\lim_{k\to\infty} E(e_k^T e_k) = 0$. The system (2.3) with a mean-square stabilizing control u_k is called mean-square stabilizable.

Definition 2 ([21]). u_k is said to be admissible if u_k satisfies the following: (1) u_k is a F_k adapted and measurable stochastic process; (2) u_k is mean-square stabilizing; (3) It enables the cost function to reach the minimum value.

The goal of this paper is to seek an admissible control, which not only minimizes the cost function (2.5) but also stabilizes the system (2.3) for each initial state e_0. We denote the optimal cost function as follows

$$V(e_0) = \min_u J(e_0, u). \tag{2.6}$$

In order to achieve the above goal, this paper establishes an augmented system composed of system (2.1) and the reference trajectory system (2.2), and then transforms the optimal tracking problem into an optimal regulation problem.

The system (2.1) can be rewritten as the following equivalent form:

$$\begin{aligned} x_{k+1} &= [A \ \ A_d]\begin{bmatrix} x_k \\ x_{k-d} \end{bmatrix} + [B \ \ B_d]\begin{bmatrix} u_k \\ u_{k-d} \end{bmatrix} \\ &\quad + ([C \ \ C_d]\begin{bmatrix} x_k \\ x_{k-d} \end{bmatrix} + [D \ \ D_d]\begin{bmatrix} u_k \\ u_{k-d} \end{bmatrix})\omega_k, \\ y_k &= [E \ \ E_d]\begin{bmatrix} x_k \\ x_{k-d} \end{bmatrix}. \end{aligned} \tag{2.7}$$

According to [16, 22, 23], we define the delay operator ∇_d satisfies $\nabla_d x_k = x_{k-d}$ and $(\nabla_d x_k)^T = x_{k-d}^T$. Then, the system (2.7) can be expressed as

$$\begin{aligned} x_{k+1} &= A_\nabla x_k + B_\nabla u_k + (C_\nabla x_k + D_\nabla u_k)\omega_k, \\ y_k &= E_\nabla x_k \end{aligned} \tag{2.8}$$

where $A_\nabla = A + A_d\nabla_d$, $B_\nabla = B + B_d\nabla_d$, $C_\nabla = C + C_d\nabla_d$, $D_\nabla = D + D_d\nabla_d$, $E_\nabla = E + E_d\nabla_d$.

Based on the system (2.1) and the reference trajectory system (2.2), the augmented system can be defined as

$$\begin{aligned} G_{k+1} &= \begin{bmatrix} x_{k+1} \\ r_{k+1} \end{bmatrix} = \begin{bmatrix} A_\nabla + C_\nabla\omega_k & 0 \\ 0 & F \end{bmatrix}\begin{bmatrix} x_k \\ r_k \end{bmatrix} + \begin{bmatrix} B_\nabla + D_\nabla\omega_k \\ 0 \end{bmatrix}u_k \\ &= TG_k + B_0u_k \end{aligned} \tag{2.9}$$

where $G_k = \begin{bmatrix} x_k \\ r_k \end{bmatrix} \in \mathcal{R}^{n+q}$, $T \in \mathcal{R}^{(n+q)\times(n+q)}$, $B_0 \in \mathcal{R}^{(n+q)\times m}$.

Based on the augmented system (2.9), the cost function (2.5) can be expressed as

$$J(G_k, u_k) = E\sum_{i=k}^{\infty} \gamma^{i-k}[G_i^T O_1 G_i + u_i^T \mathcal{R}_\nabla u_i] \tag{2.10}$$

where $O_1 = \begin{bmatrix} E & -I \end{bmatrix}^T O \begin{bmatrix} E & -I \end{bmatrix} \in \mathcal{R}^{(n+q)\times(n+q)}$, $R_\nabla = R + R_d\nabla_d$.

The state feedback linear controller is defined as

$$u_k = KG_k, \quad K \in \mathcal{R}^{m\times(n+q)} \tag{2.11}$$

where K represents the control gain matrix of the system.

Substituting (2.11) into (2.10), the cost function (2.10) can be transformed into

$$J(G_k, K) = E\sum_{i=k}^{\infty} \gamma^{i-k} G_i^T [O_1 + K^T R_\nabla K] G_i. \tag{2.12}$$

Therefore, the target of SQL optimal tracking problem with delays can be further expressed as

$$V(G_0, K) = \min_K J(G_0, K). \tag{2.13}$$

Definition 3. The SLQ optimal control problem is well posed if

$$-\infty < V(G_0, K) < +\infty.$$

Before solving the SLQ control problem, we need to know whether it is well-posed. Therefore, we give the following lemma first.

Lemma 1. *If there exists an admissible control $u_k = KG_k$, then the SLQ optimal tracking control is well-posed, and the cost function can be expressed as*

$$J(G_k, K) = E(G_k^T P G_k) \tag{2.14}$$

where the matrix $P \in \mathcal{R}^{(n+q)\times(n+q)}$ satisfies the following augmented SAE

$$\begin{aligned} P &= \gamma(A_1 + B_1K)^T P(A_1 + B_1K) \\ &\quad + \gamma(C_1 + D_1K)^T P(C_1 + D_1K) + O_1 + K^T R_\nabla K \end{aligned} \tag{2.15}$$

where $A_1 = \begin{bmatrix} A_\nabla & 0 \\ 0 & F \end{bmatrix} \in \mathcal{R}^{(n+q)\times(n+q)}$, $B_1 = \begin{bmatrix} B_\nabla \\ 0 \end{bmatrix} \in \mathcal{R}^{(n+q)\times m}$, $C_1 = \begin{bmatrix} C_\nabla & 0 \\ 0 & 0 \end{bmatrix} \in \mathcal{R}^{(n+q)\times(n+q)}$, $D_1 = \begin{bmatrix} D_\nabla \\ 0 \end{bmatrix} \in \mathcal{R}^{(n+q)\times m}$.

Proof. Assuming that the control u_k is admissible and the matrix P satisfies (2.15), then

$$\begin{aligned} & E\sum_{i=k}^{\infty} [\gamma {G_{i+1}}^T P G_{i+1} - {G_i}^T P G_i] \\ &= E\sum_{i=k}^{\infty} \Big\{\gamma[(A_1 + B_1K)G_i + (C_1\omega_i + D_1K\omega_i)G_i]^T P \\ &\qquad [(A_1 + B_1K)G_i + (C_1\omega_i + D_1K\omega_i)G_i] - {G_i}^T P G_i\Big\} \\ &= E\sum_{i=k}^{\infty} \Big\{G_i^T[\gamma(A_1 + B_1K)^T P(A_1 + B_1K) \\ &\qquad + \gamma(C_1 + D_1K)^T P(C_1 + D_1K) - P]G_i\Big\}. \end{aligned}$$

Based on (2.12) and (2.15), we have

$$\begin{aligned} J(G_k, K) &= E\sum_{i=k}^{\infty}\gamma^{i-k}G_i^T[O_1 + K^T R_\nabla K]G_i] \\ &= E\sum_{i=k}^{\infty}\gamma^{i-k}G_i^T[P - \gamma(A_1 + B_1K)^T P(A_1 + B_1K) \\ &\quad - \gamma(C_1 + D_1K)^T P(C_1 + D_1K)]G_i \\ &= -E\sum_{i=k}^{\infty}\gamma^{i-k}[\gamma G_{i+1}^T PG_{i+1} - G_i^T PG_i] \\ &= E(G_k^T PG_k) - \lim_{i\to\infty}\gamma^{i-k+1}E(G_i^T PG_i) \\ &= E(G_k^T PG_k). \end{aligned}$$

□

Since the feedback control u_k is admissible, we can obtain $J(G_k, K)$= E$({G_k}^T PG_k)$, which satisfies the well-posedness of SLQ optimal tracking control problem.

To make sure the mean-square stable control, we make the following assumption.

Assumption 1. The system (2.9) is mean-square stabilizable.

2.2. Problem transformation

At present, ADP algorithm has achieved great success in the optimal tracking control of deterministic systems [24–26], which inspires us to transform stochastic problems into deterministic problems through system transformation.

Let $M_k = E(G_kG_k^T)$, then the system (2.9) can be converted to

$$\begin{aligned} M_{k+1} &= E(G_{k+1}G_{k+1}^T) \\ &= E((TG_k + B_0u_k)(TG_k + B_0u_k)^T) \\ &= (A_1 + B_1K)M_k(A_1 + B_1K)^T \\ &\quad + (C_1 + D_1K)M_k(C_1 + D_1K)^T \end{aligned} \tag{2.16}$$

where $M_k \in \mathcal{R}^{(n+q)\times(n+q)}$ is the state of a deterministic system and M_0 is the initial state.

Therefore, the cost function (2.10) can be rewritten as

$$J(M_k, K) = tr\{\sum_{i=k}^{\infty}\gamma^{i-k}[(O_1 + K_T R_\nabla K)M_k]\}. \tag{2.17}$$

Remark 1. After system transformation, the stochastic system is transformed into deterministic system. The system (2.17) completely gets rid of stochastic disturbance ω_k and will only be dependent on the initial state M_0 and control gain matrix K, which makes preparation for the derivation and application of Q-learning algorithm.

3. The Q-learning algorithm and convergence proof

In this paper, Q-learning method is used to solve the SLQ optimal tracking problem, which avoids the need for accurate system model. Thus we first give the formula of the optimal control and the corresponding augmented SAE.

Lemma 2. *Given the admissible control u_k, we can get the following optimal control*

$$u_k^* = K^* G_k = -(R_\nabla + \gamma B_1^T P B_1)^{-1} \gamma (B_1^T P A_1 + D_1^T P D_1) G_k \tag{3.1}$$

and the optimal cost function

$$V(G_k) = E(G_k^T P G_k) = tr(P M_k) \tag{3.2}$$

where the matrix P satisfies the following augmented SAE

$$\begin{cases} P = O_1 + \gamma(A_1^T P A_1 + C_1^T P C_1) - \gamma(A_1^T P B_1 + C_1^T P D_1) \\ \quad \times (R_\nabla + \gamma B_1^T P B_1 + \gamma D_1^T P D_1)^{-1} \gamma (B_1^T P A_1 + D_1^T P C_1) \\ R_\nabla + \gamma B_1^T P B_1 + D_1^T P D_1 > 0 \end{cases} . \tag{3.3}$$

Proof. Suppose u_k is an admissible control. According to Lemma 1 and (2.17), the cost function can be written as

$$\begin{aligned} J(M_k, K) &= tr\{\sum_{i=k}^{\infty} \gamma^{i-k} [(O_1 + K^T R_\nabla K) M_i]\} \\ &= tr\{(O_1 + K^T R_\nabla K) M_i\} + tr\{\sum_{i=k+1}^{\infty} \gamma^{i-k} [(O_1 + K^T R_\nabla K) M_i]\} \\ &= tr\{(O_1 + K^T R_\nabla K) M_i\} + J(M_{k+1}, K). \end{aligned} \tag{3.4}$$

According to Bellman optimality principle, the optimal cost function satisfies

$$V(M_k) = \min_K \{tr\{(O_1 + K^T R_\nabla K) M_k\} + V(M_{k+1})\}. \tag{3.5}$$

The optimal control gain matrix can be obtained as follow

$$K^*(M_k) = \arg\min_K \{tr\{(O_1 + K^T R_\nabla K) M_k\} + V(M_{k+1})\}. \tag{3.6}$$

Considering the first-order necessary condition

$$\frac{\partial [tr\{(O_1 + K^T R_\nabla K) M_k\} + V(M_{k+1})]}{\partial K} = 0, \tag{3.7}$$

we can obtain

$$(R_\nabla + \gamma B_1^T P B_1 + \gamma D_1^T P D^T) K G_k + \gamma (B_1^T P A_1 + D_1^T P C_1) G_k = 0 \tag{3.8}$$

where the matrix P satisfies augmented SAE (2.15).

Supposing $R_\nabla + \gamma B_1^T P B_1 + \gamma D_1^T P D^T > 0$, we have

$$K^* = -(R_\nabla + \gamma B_1^T P B_1)^{-1} \gamma (B_1^T P A_1 + D_1^T P D_1). \tag{3.9}$$

When taking (3.9) into the (2.15), we can obtain

$$\begin{aligned} P = {} & O_1 + \gamma(A_1^T PA_1 + C_1^T PC_1) - \gamma(A_1^T PB_1 + C_1^T PD_1) \\ & \times (R_\nabla + \gamma B_1^T PB_1 + \gamma D_1^T PD^T)^{-1} \gamma(B_1^T PA_1 + D_1^T PC_1). \end{aligned} \tag{3.10}$$

□

From Lemma 2, the SQL optimal tracking problem can be dealt with by the solution of augmented SAE (3.3). However, solving augmented SAE (3.3) requires accurate system model, so this method is not feasible when the dynamics are unknown.

3.1. Derivation of Q-learning algorithm

To solve model-free SQL optimal tracking problem with delays, we give the definition of the Q function and the corresponding matrix H.

Based on (2.10) and Bellman optimality principle, we know that the optimal cost function satisfies Hamilton Jacobi Bellman (HJB) equation

$$V(G_k) = \min_{u_k}\{E[G_k^T O_1 G_k + u_k^T R_\nabla u_k] + \gamma V(G_{k+1})\}. \tag{3.11}$$

The Q-function is defined as

$$Q(G_k,u_k) = E[{G_k}^T O_1 G_k + u_k^T R_\nabla u_k] + \gamma V(G_{k+1}). \tag{3.12}$$

According to Lemma 1, $V(G_{k+1})$ can be written as

$$\begin{aligned} & V(G_{k+1}) \\ & = E(G_{k+1}^T PG_{k+1}) \\ & = E\{(TG_k + B_0u_k)^T P(TG_k + B_0u_k)\} \\ & = E\{[(A_1G_k + C_1\omega_k G_k) + (B_1u_k + D_1\omega_k u_k)]^T \\ & \qquad P[(A_1G_k + C_1\omega_k G_k) + (B_1u_k + D_1\omega_k u_k)]\}. \end{aligned} \tag{3.13}$$

Substitute (3.13) into (3.12), we can get

$$Q(G_k,u_k) = \mathrm{E}\left\{\begin{bmatrix} G_k \\ u_k \end{bmatrix}^T \begin{bmatrix} H_{GG} & H_{Gu} \\ H_{uG} & H_{uu} \end{bmatrix} \begin{bmatrix} G_k \\ u_k \end{bmatrix}\right\} = \mathrm{E}\left\{\begin{bmatrix} G_k \\ u_k \end{bmatrix}^T H \begin{bmatrix} G_k \\ u_k \end{bmatrix}\right\} \tag{3.14}$$

where $H = H^T \in \mathcal{R}^{(n+q+m)\times(n+q+m)}$,

$$\begin{aligned} H &= \begin{bmatrix} H_{GG} & H_{Gu} \\ H_{uG} & H_{uu} \end{bmatrix} \\ &= \begin{bmatrix} O_1 + \gamma A_1^T PA_1 + \gamma C_1^T PC_1 & \gamma A_1^T PB_1 + \gamma C_1^T PD_1 \\ \gamma B_1^T PA_1 + \gamma D_1^T PC_1 & \gamma B_1^T PB_1 + \gamma D_1^T PD_1 + R_\nabla \end{bmatrix}. \end{aligned} \tag{3.15}$$

Let $\frac{\partial Q(G_k,u_k)}{\partial u_k} = 0$, then the optimal control can be obtained as follow

$$u_k^* = -H_{uu}^{-1} H_{uG} G_k. \tag{3.16}$$

From Lemma 1 and (3.15), we can know the relationship between matrix P and matrix H.

$$P = \begin{bmatrix} I & K^T \end{bmatrix} H \begin{bmatrix} I & K^T \end{bmatrix}^T. \tag{3.17}$$

As can be seen from (3.16), the optimal control only depends on the matrix H, which is completely get rid of the constraints of the system parameters. Next, we will present the Q-learning iterative algorithm for estimating the matrix H.

In this section, we propose Q-learning iterative algorithm based on the VI. This method starts with the initial value $Q_0(G_k, u_k) = 0$ and the initial admissible control $u_0(G_k)$, $Q_1(G_k, u_k)$ will be updated by the initial value and the initial control as follows

$$Q_1(G_k, u_k) = E[G_k{}^T O_1 G_k + u_0^T(G_k) R_\nabla u_0(G_k)] + \gamma Q_0(G_{k+1}, u_0(G_{k+1})). \tag{3.18}$$

The control is updated as follows

$$u_1(G_k) = \arg\min_{u(G_k)} Q_1(G_k, u_k) \tag{3.19}$$

for $i \geq 1$, Q-learning algorithm iterates between

$$Q_{i+1}(G_k,u_k) = E[G_k{}^T O_1 G_k + u_i^T(G_k) R_\nabla u_i(G_k)] + \gamma Q_i(G_{k+1},u_i(G_{k+1})) \tag{3.20}$$

and

$$u_{i+1}(G_k) = \arg\min_{u_k} \{E[G_k^T O_1 G_k + u_k^T R_\nabla u_k] + \min_{u_{k+1}} Q_i(G_{k+1}, u_{k+1})\} \tag{3.21}$$

where i is the iteration index and k is time index.

According to (3.14), the Q function can be rewritten as

$$\begin{aligned} Q_{i+1}(G_k,u_k) =& \begin{bmatrix} G_k{}^T & u_i^T(G_k) \end{bmatrix} H_{i+1} \begin{bmatrix} G_k{}^T & u_i^T(G_k) \end{bmatrix}^T \\ =& E\Bigg\{ \begin{bmatrix} G_k{}^T & u_i^T(G_k) \end{bmatrix} \begin{bmatrix} O_1 & 0 \\ 0 & R_\nabla \end{bmatrix} \begin{bmatrix} G_k{}^T & u_i^T(G_k) \end{bmatrix}^T \\ &+ \gamma \begin{bmatrix} G_{k+1}{}^T & u_i^T(G_{k+1}) \end{bmatrix} H_i \begin{bmatrix} G_{k+1}{}^T & u_i^T(G_{k+1}) \end{bmatrix}^T \Bigg\} \end{aligned} \tag{3.22}$$

and we can obtain the optimal controller

$$u_i(G_k) = -H_{uu,i}^{-1} H_{uG,i} G_k. \tag{3.23}$$

According to (3.17), we can get

$$P_i = \begin{bmatrix} I & K_i^T \end{bmatrix} H_i \begin{bmatrix} I & K_i^T \end{bmatrix}^T. \tag{3.24}$$

3.2. The convergence of Q-learning algorithm

Before proving the convergence of Q-learning algorithm, we first give the following two lemmas.

Lemma 3. *Q-learning algorithm (3.22) and (3.23) is equivalent to*

$$\begin{aligned} P_{i+1} =& O_1 + \gamma(A_1^T P_i A_1 + C_1^T P_i C_1) - \gamma(A_1^T P_i B_1 + C_1^T P_i D_1) \\ &\times (R + \gamma B_1^T P_i B_1 + \gamma D_1^T P_i D_1)^{-1} \gamma(B_1^T P_i A_1 + D_1^T P_i C_1). \end{aligned} \tag{3.25}$$

Proof. According to (2.11), the last term of (3.22) can be written as

$$\begin{aligned}
&E\left\{\left[\begin{array}{cc} {G_{k+1}}^T & u_i^T(G_{k+1}) \end{array}\right] H_i \left[\begin{array}{cc} {G_{k+1}}^T & u_i^T(G_{k+1}) \end{array}\right]^T\right\} \\
&= E\left\{{G_{k+1}}^T \left[\begin{array}{cc} I & K_i^T \end{array}\right] H_i \left[\begin{array}{cc} I & K_i^T \end{array}\right]^T G_{k+1}\right\} \\
&= E\{[(A_1G_k + C_1\omega_k G_k) + (B_1u_i(G_k) + D_1\omega_k u_i(G_k))]^T \left[\begin{array}{cc} I & K_i^T \end{array}\right] H_i \\
&\quad \left[\begin{array}{cc} I & K_i^T \end{array}\right]^T (A_1G_k + C_1\omega_k G_k) + (B_1u_i(G_k) + D_1\omega_k u_i(G_k))]\} \\
&= E\left\{\left[\begin{array}{cc} {G_k}^T & {u_i}^T(G_k) \end{array}\right]\left[\begin{array}{cc} A_1 & B_1 \end{array}\right]^T \left[\begin{array}{cc} I & {K_i}^T \end{array}\right] H_i \right. \\
&\quad \left[\begin{array}{cc} I & {K_i}^T \end{array}\right]^T \left[\begin{array}{cc} A_1 & B_1 \end{array}\right]\left[\begin{array}{cc} {G_k}^T & {u_i}^T(G_k) \end{array}\right]^T \\
&\quad + \left[\begin{array}{cc} {G_k}^T & {u_i}^T(G_k) \end{array}\right]\left[\begin{array}{cc} C_1 & D_1 \end{array}\right]^T \left[\begin{array}{cc} I & {K_i}^T \end{array}\right] H_i \\
&\quad \left.\left[\begin{array}{cc} I & {K_i}^T \end{array}\right]^T \left[\begin{array}{cc} C_1 & D_1 \end{array}\right]\left[\begin{array}{cc} {G_k}^T & {u_i}^T(G_k) \end{array}\right]^T\right\}.
\end{aligned} \tag{3.26}$$

Substitute (3.26) into (3.22), according to (3.24),we can get

$$\begin{aligned}
H_{i+1} &= \left[\begin{array}{cc} O_1 & 0 \\ 0 & R_\nabla \end{array}\right] + \left[\begin{array}{cc} \gamma A_1^T P_i A_1 & \gamma A_1^T P_i B_1 \\ \gamma B_1^T P_i A_1 & \gamma B_1^T P_i B_1 \end{array}\right] \\
&+ \left[\begin{array}{cc} \gamma C_1^T P_i C_1 & \gamma C_1^T P_i D_1 \\ \gamma D_1^T P_i C_1 & \gamma D_1^T P_i D_1 \end{array}\right].
\end{aligned} \tag{3.27}$$

Based on (3.24), we have

$$P_{i+1} = \left[\begin{array}{cc} I & K_{i+1}^T \end{array}\right] H_{i+1} \left[\begin{array}{cc} I & K_{i+1}^T \end{array}\right]^T. \tag{3.28}$$

Substitute (3.27) into (3.28), we can get

$$\begin{aligned}
P_{i+1} &= O_1 + \gamma(A_1^T P_i A_1 + C_1^T P_i C_1) - \gamma(A_1^T P_i B_1 + C_1^T P_i D_1) \\
&\times (R + \gamma B_1^T P_i B_1 + \gamma D_1^T P_i D_1)^{-1} \gamma(B_1^T P_i A_1 + D_1^T P_i C_1)
\end{aligned} \tag{3.29}$$

where $R_\nabla + \gamma B_1^T P B_1 + D_1^T P D_1 > 0$. □

Lemma 4 ([27]). *The value iteration algorithm iterates between*

$$V_{i+1}(G_k) = E({G^T}_k(O_1 + {K_i}^T R_\nabla K_i)G_k) + \gamma V_i(G_{k+1}) \tag{3.30}$$

and

$$K_{i+1} = \underset{K}{\operatorname{argmin}}\{E({G^T}_k(O_1 + {K_i}^T R_\nabla K_i)G_k) + \gamma \mathrm{V}_i(G_{k+1})\} \tag{3.31}$$

is the convergence, then

$$\lim_{i\to\infty} V_i(G_k) = V(G_k) = E({G_k}^T P G_k) = tr\{PM_k\},$$

$$\lim_{i\to\infty} K_i = K^* = -(\mathrm{R}_\nabla + \gamma \mathrm{B}_1^T \mathrm{P} \mathrm{B}_1 + \gamma \mathrm{D}_1^T \mathrm{P} \mathrm{D}_1)^{-1} \gamma(\mathrm{B}_1^T \mathrm{P} \mathrm{A}_1 + \mathrm{D}_1^T \mathrm{P} \mathrm{C}_1)$$

where the matrix P satisfies the augmented SAE (3.3).

Theorem 3.1. *Assuming that system (2.9) is mean-square stabilizable, the matrix sequence $\{H_i\}$ calculated by Q-learning algorithm (3.22) converges to matrix H and the matrix sequence $\{P_i\}$ calculated by (3.24) converges to the solution P of augmented SAE (3.3).*

Proof. According to Lemma 4, (3.30) can be rewritten as

$$\begin{aligned} V_{i+1}(G_k) &= E({G_k}^T P_{i+1} G_k) \\ &= E\left[{G_k}^T (O_1 + K_i R_\nabla K_i) G_k\right] + E(G_{k+1}^T P_i G_{k+1}) \\ &= E\{{G_k}^T (O_1 + K_i R_\nabla K_i) G_k + [(A_1 + B_1 K) G_i \\ &+ (C_1 \omega_i + D_1 K \omega_i) G_i]^T P [(A_1 + B_1 K) G_i \\ &+ (C_1 \omega_i + D_1 K \omega_i) G_i]\} \\ &= E({G_i}^T [(A_1 + B_1 K)^T P (A_1 + B_1 K) \\ &+ (C_1 + D_1 K)^T P (C_1 + D_1 K) + O_1 + {K_i}^T R_\nabla K_i] G_i). \end{aligned} \tag{3.32}$$

We can update the control gain matrix by (3.31) as follows

$$K_i = -(R_\nabla + \gamma B_1^T P_i B_1 + \gamma D_1^T P_i D_1)^{-1} \gamma (B_1^T P_i A_1 + D_1^T P_i C_1). \tag{3.33}$$

Substituting (3.33) into (3.32), we can get

$$\begin{aligned} P_{i+1} = O_1 &+ \gamma (A_1^T P_i A_1 + C_1^T P_i C_1) - \gamma (A_1^T P_i B_1 + C_1^T P_i D_1) \\ &\times (R + \gamma B_1^T P_i B_1 + \gamma D_1^T P_i D_1)^{-1} \gamma (B_1^T P_i A_1 + D_1^T P_i C_1). \end{aligned} \tag{3.34}$$

According to Lemmas 3 and 4, we can conclude $\lim_{i\to\infty} P_i = P$. when $i \to \infty$, the matrix P satisfies

$$\begin{aligned} P = O_1 &+ \gamma (A_1^T P A_1 + C_1^T P C_1) - \gamma (A_1^T P B_1 + C_1^T P D_1) \\ &\times (R + \gamma B_1^T P B_1 + \gamma D_1^T P D_1)^{-1} \gamma (B_1^T P A_1 + D_1^T P C_1). \end{aligned} \tag{3.35}$$

Based on (3.27), we can know H satisfies $\lim_{i\to\infty} H_i = H$, where

$$H = \begin{bmatrix} \gamma A_1^T P A_1 + \gamma C_1^T P C_1 + Q_1 & \gamma A_1^T P B_1 + \gamma C_1^T P D_1 \\ B_1^T P A_1 + \gamma D_1^T P C_1 & \gamma B_1^T P B_1 + \gamma D_1^T P D_1 + R_\nabla \end{bmatrix}. \tag{3.36}$$

So the Q-learning algorithm converges. □

4. Implementation of the Q-learning algorithm

Due to the existence of stochastic disturbance, the output trajectory of the system is uncertain, and the cost function has expectations, the online algorithm cannot achieve the function. Therefore, it is necessary to transform the stochastic Q-learning algorithm into a deterministic Q-learning algorithm. In this section, we will give the implementation steps of deterministic Q-learning algorithm. The flow chart of Q learning algorithm is shown in Figure 1.

According to Eq (2.11), the left side of (3.22) can be simplified to

$$\begin{aligned} & E\left\{\begin{bmatrix} {G_k}^T & u_i^T(G_k) \end{bmatrix} H_{i+1} \begin{bmatrix} {G_k}^T & u_i^T(G_k) \end{bmatrix}\right\} \\ &= E\left\{{G_k}^T \begin{bmatrix} I & K_i^T \end{bmatrix} H_{i+1} \begin{bmatrix} I & K_i^T \end{bmatrix}^T G_k\right\} \\ &= tr\left\{\begin{bmatrix} I & K_i^T \end{bmatrix} H_{i+1} \begin{bmatrix} I & K_i^T \end{bmatrix}^T M_k\right\}. \end{aligned} \tag{4.1}$$

The right side of (3.22) can be simplified as

$$\begin{aligned} & E\left\{G_k^{\ T}\begin{bmatrix} I & K_i^T \end{bmatrix}\begin{bmatrix} O_1 & 0 \\ 0 & R_\nabla \end{bmatrix}\begin{bmatrix} I & K_i^T \end{bmatrix}^T G_k \right. \\ & \quad \left. + G_{k+1}^{\ T}\begin{bmatrix} I & K_i^T \end{bmatrix} H_i \begin{bmatrix} I & K_i^T \end{bmatrix}^T G_{k+1}\right\} \\ & \quad = tr\left\{\begin{bmatrix} I & K_i^T \end{bmatrix}\begin{bmatrix} O_1 & 0 \\ 0 & R_\nabla \end{bmatrix}\begin{bmatrix} I & K_i^T \end{bmatrix}^T M_k \right. \\ & \quad \left. + \begin{bmatrix} I & K_i^T \end{bmatrix} H_i \begin{bmatrix} I & K_i^T \end{bmatrix}^T M_{k+1}\right\}. \end{aligned} \tag{4.2}$$

For simplicity, let

$$L_i(H_i) = \begin{bmatrix} I & K_i^T \end{bmatrix} H_i \begin{bmatrix} I & K_i^T \end{bmatrix}^T, \quad i = 1, 2, 3, \cdots. \tag{4.3}$$

Then (3.22) can be simplified as

$$tr\{L_i(H_{i+1})M_k\} = tr\left\{L_i(\begin{bmatrix} O_1 & 0 \\ 0 & R_\nabla \end{bmatrix})M_k + L_i(H_i)M_{k+1}\right\}. \tag{4.4}$$

The Q-learning iterative algorithm consisting of (4.4) and (3.23) only relies on determining the state M_k of the system (2.16) and iteratively controlling the gain matrix K_i, avoiding the constraints of system parameters and stochastic disturbance.

Remark 2. The Q-learning algorithm based on VI is performed online and solves (4.4) using least squares (LS) without knowing augmented system. In fact, (4.4) is a scalar equation and H is a symmetric $(n+q+m)\times(n+q+m)$ matrix with $(n+q+m)\times(n+q+m+1)/2$ independent elements. Therefore, at least $(n+q+m+1)\times(n+q+m+1)/2$ data tuples are required before (4.4) can be solved using LS.

Remark 3. Q-learning algorithm based on VI requires a persistent excitation (PE) condition [28] to ensure the sufficient exploration of the state space.

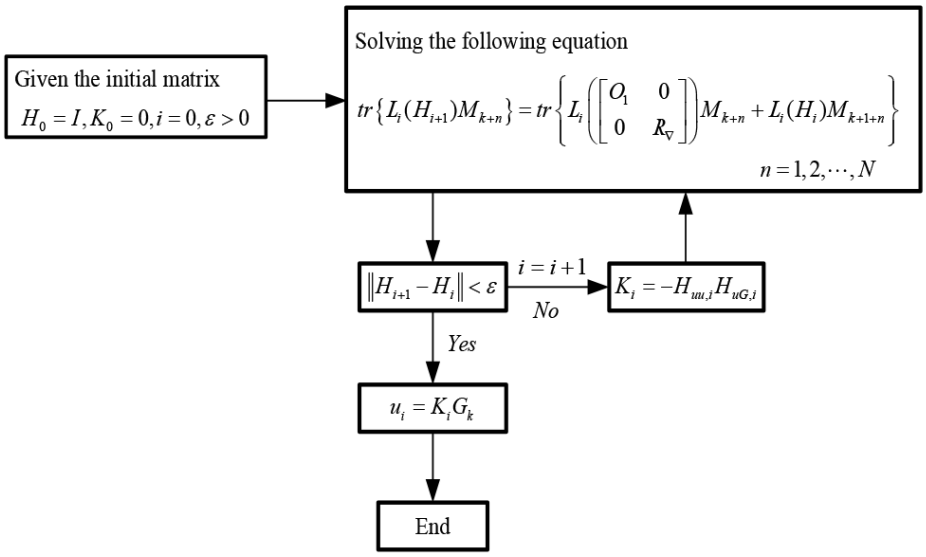

Figure 1. Flowchart of Q-learning.

5. Simulation

In this section, a simulation example is given to illustrate the effectiveness of Q-learning algorithm. Consider the following stochastic linear system with delays

$$\begin{aligned} x_{k+1} &= Ax_k + A_d x_{k-d} + Bu_k + B_d u_{k-d} \\ &\quad + (Cx_k + C_d x_{k-d} + Du_k + D_d u_{k-d})\omega_k, \\ y_k &= Ex_k + E_d x_{k-d} \end{aligned}$$

in which $A = \begin{pmatrix} 0.2 & -0.8 \\ 0.5 & -0.7 \end{pmatrix}$, $A_d = \begin{pmatrix} 0.2 & -0.2 \\ 0.1 & 0.15 \end{pmatrix}$, $B = \begin{pmatrix} 0.03 \\ -0.5 \end{pmatrix}$, $B_d = \begin{pmatrix} 0.3 \\ -0.2 \end{pmatrix}$, $C = \begin{pmatrix} -0.04 & 0.4 \\ -0.3 & 0.13 \end{pmatrix}$, $C_d = \begin{pmatrix} 0.2 & -0.1 \\ 0.2 & 0.11 \end{pmatrix}$, $D = \begin{pmatrix} 0.05 \\ -0.3 \end{pmatrix}$, $D_d = \begin{pmatrix} 0.1 \\ 0.1 \end{pmatrix}$, $E = \begin{pmatrix} 3 & 3 \end{pmatrix}$, $E_d = \begin{pmatrix} 0.1 & 0.12 \end{pmatrix}$.

Suppose the reference trajectory is as follows

$$r_{k+1} = -r_k$$

where $r_0 = 1$.

The cost function is considered as (2.5) with $R = 1$, $R_d = 1$, $O = 10$ and delay index $d = 1$. The initial state for augmented system (2.9) is chosen as $G_0 = \begin{bmatrix} 10 & -10 & 1 \end{bmatrix}^T$. The initial control gain matrix is selected as $K = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}$. In each iteration of the algorithm, 21 samples are collected to update the control gain matrix K.

In order to verify the effectiveness of the iterative Q-learning algorithm, we compared K with optimal solution K^* solved by SAE (3.1). Figure 2 shows the control gain matrix K converges to the optimal control gain matrix K^* as the number of iterations increases. Figure 3 shows the convergence process of H to its optimal values H^*, which can be calculated by (3.15). The goal of the optimal tracking problem is to trace the reference signal trajectory. In Figure 4, the expectation of system output $E(y)$ can track the reference trajectory r_k. This further proves the effectiveness of the proposed Q-learning algorithm.

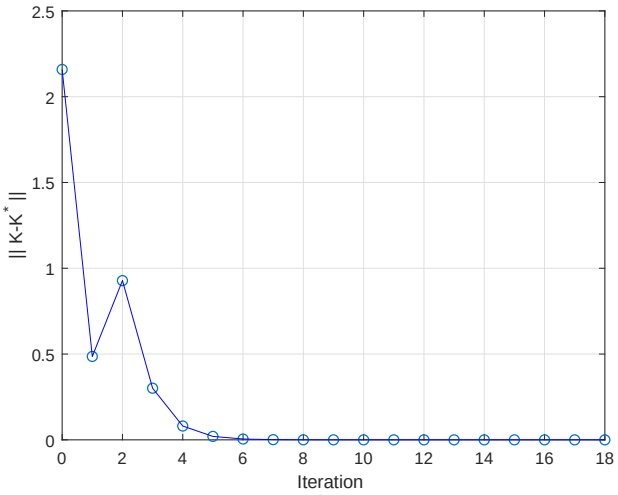

Figure 2. Convergence trajectory of control gain matrix K to K^*.

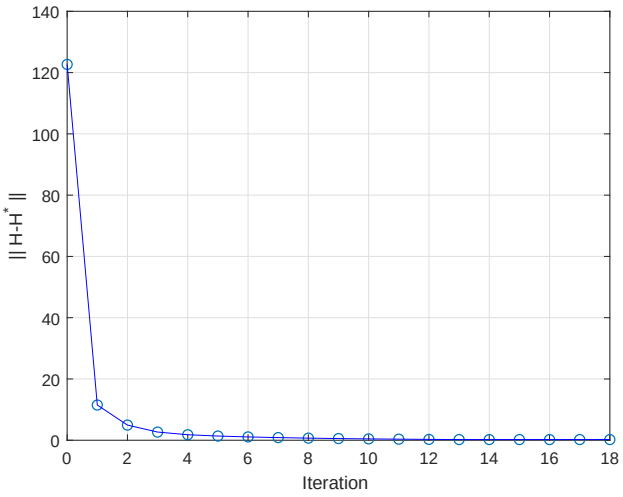

Figure 3. Convergence trajectory of matrix H to its optimal values H^*.

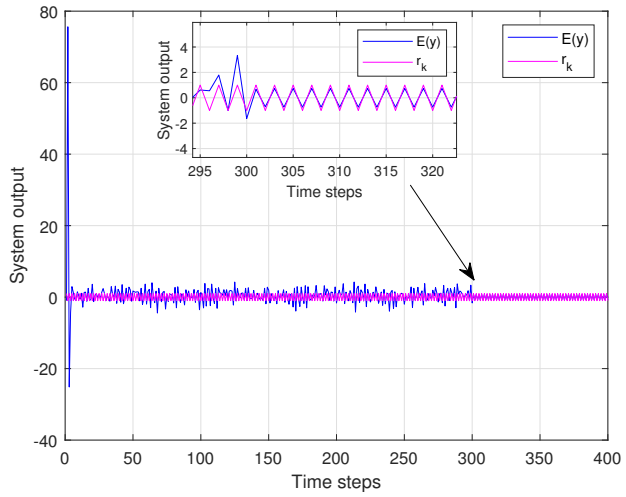

Figure 4. Curves of expectation of output $E(y)$ and reference signal r_k.

6. Conclusions

For the model-free SLQ optimal tracking problem with delays, Q-learning algorithm based on VI is proposed in this paper. This method makes full use of the system information to approximate the optimal control online, and never needs the system parameter information. In the iterative process of the algorithm, the H matrix sequence and the control gain matrix K sequence are guaranteed to approximate the optimal value. Finally, the simulation results show that the system output can track the reference trajectory effectively.

Conflict of interest

The authors declare that they have no conflicts of interest.

References

1. H. Modares, F. L. Lewis, Optimal tracking control of nonlinear partially-unknown constrained-input systems using integral reinforcement learning, *Automatica,* **50** (2014), 1780–1792. https://doi.org/10.1016/j.automatica.2014.05.011
2. B. Zhao, Y. Li, Model-free adaptive dynamic programming based near-optimal decentralized tracking control of reconfigurable manipulators, *Int. J. Control, Autom. Syst.*, **16** (2018), 478–490. https://doi.org/10.1007/s12555-016-0711-5
3. T. Huang, D. Liu, A self-learning scheme for residential energy system control and management, *Neural Comput. Appl.*, **22** (2013), 259–269. https://doi.org/10.1007/s00521-011-0711-6
4. M. Gluzman, J. G. Scott, A. Vladimirsky, Optimizing adaptive cancer therapy: dynamic programming and evolutionary game theory, *Proc. Royal Soc. B: Biol. Sci.*, **287** (2020), 20192454. https://doi.org/10.1098/rspb.2019.2454
5. I. Ha, E. Gilbert, Robust tracking in nonlinear systems, *IEEE Trans. Automat. Control*, **32** (1987), 763–771. https://doi.org/10.1109/TAC.1987.1104710
6. M. A. Rami, X. Y. Zhou, Linear matrix inequalities, Riccati equations and indefinite stochastic linear quadratic controls, *IEEE Trans. Automat. Control*, **45** (2000), 1131–1143. https://doi.org/10.1109/9.863597
7. R. Byers, Solving the algebraic Riccati equation with the matrix sign function, *Linear Algebra Appl.*, **89** (1987), 267–279. https://doi.org/10.1016/0024-3795(87)90222-9
8. D. Vrabie, O. Pastravanu, M. Abu-Khalaf, F. L. Lewis, Adaptive optimal control for continuous-time linear systems based on policy iteration, *Automatica*, **45** (2009), 477–484. https://doi.org/10.1016/j.automatica.2008.08.017
9. B. Kiumarsi, F. L. Lewis, M. B. Naghibi-Sistani, A. Karimpour, Optimal tracking control of unknown discrete-time linear systems using input-output measured data, *IEEE Trans. Cybern.*, **45** (2015), 2770–2779. https://doi.org/10.1109/TCYB.2014.2384016
10. B. Kiumarsi, F. L. Lewis, H. Modares, A. Karimpour, M. B. Naghibi-Sistani, Reinforcement Q-learning for optimal tracking control of linear discrete-time systems with unknown dynamics, *Automatica*, **50** (2014), 1167–1175. https://doi.org/10.1016/j.automatica.2014.02.015
11. G. Wang, H. Zhang, Model-free value iteration algorithm for continuous-time stochastic linear quadratic optimal control problems, *arXiv*, 2022. https://doi.org/10.48550/arXiv.2203.06547
12. H. Zhang, Adaptive dynamic programming-based algorithm for infinite-horizon linear quadratic stochastic optimal control problems, *arXiv*, 2022. https://doi.org/10.48550/arXiv.2210.04486

13. R. Liu, Y. Li, X. Liu, Linear-quadratic optimal control for unknown mean-field stochastic discrete-time system via adaptive dynamic programming approach, *Neurocomputing*, **282** (2018), 16–24. https://doi.org/10.1016/j.neucom.2017.12.007

14. X. Chen, F. Wang, Neural-network-based stochastic linear quadratic optimal tracking control scheme for unknown discrete-time systems using adaptive dynamic programming, *Control Theory Technol.*, **19** (2021), 315–327. https://doi.org/10.1007/s11768-021-00046-y

15. Z. Zhang, X. Zhao, Stochastic linear quadratic optimal tracking control for stochastic discrete time systems based on Q-learning, *J. Nanjing Univ. Inf. Sci. Technol. (Nat. Sci.)*, **13** (2021), 548–555.

16. Y. Liu, H. Zhang, Y. Luo, J. Han, ADP based optimal tracking control for a class of linear discrete-time system with multiple delays, *J. Franklin Inst.*, **353** (2016), 2117–2136. https://doi.org/10.1016/j.jfranklin.2016.03.012

17. B. L. Zhang, Q. L. Han, X. M. Zhang, X. Yu, Sliding mode control with mixed current and delayed states for offshore steel jacket platforms, *IEEE Trans. Control Syst. Technol.*, **22** (2014), 1769–1783. https://doi.org/10.1109/TCST.2013.2293401

18. M. J. Park, O. M. Kwon, J. H. Ryu, Advanced stability criteria for linear systems with time-varying delays, *J. Franklin Inst.*, **355** (2018), 520–5433. https://doi.org/10.1016/j.jfranklin.2017.11.029

19. H. Zhang, Y. Luo, D. Liu, Neural-network-based near-optimal control for a class of discrete-time affine nonlinear systems with control constraints, *IEEE Trans. Neural Networks*, **20** (2009), 1490–1503. https://doi.org/10.1109/TNN.2009.2027233

20. H. Zhang, Z. Wang, D. Liu, Global asymptotic stability of recurrent neural networks with multiple time-varying delays, *IEEE Trans. Neural Networks*, **19** (2008), 855–873. https://doi.org/10.1109/TNN.2007.912319

21. T. Wang, H. Zhang, Y. Luo, Infinite-time stochastic linear quadratic optimal control for unknown discrete-time systems using adaptive dynamic programming approach, *Neurocomputing*, **171** (2016), 379–386. https://doi.org/10.1016/j.neucom.2015.06.053

22. A. Garate-Garcia, L. A. Marquez-Martinez, C. H. Moog, Equivalence of linear time-delay systems, *IEEE Trans. Automat. Control*, **56** (2011), 666–670. https://doi.org/10.1109/TAC.2010.2095550

23. Y. Liu, R. Yu, Model-free optimal tracking control for discrete-time system with delays using reinforcement Q-learning, *Electron. Lett.*, **54** (2018), 750–752. https://doi.org/10.1049/el.2017.3238

24. H. Zhang, Q. Wei, 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*, **38** (2008), 937–942. https://doi.org/10.1109/TSMCB.2008.920269

25. J. Shi, D. Yue, X. Xie, Adaptive optimal tracking control for nonlinear continuous-time systems with time delay using value iteration algorithm, *Neurocomputing*, **396** (2020), 172–178. https://doi.org/10.1016/j.neucom.2018.07.098

26. Q. Wei, D. Liu, Adaptive dynamic programming for optimal tracking control of unknown nonlinear systems with application to coal gasification, *IEEE Trans. Automat. Sci. Eng.*, **11** (2014), 1020–1036. https://doi.org/10.1109/TASE.2013.2284545

27. T. Wang, H. Zhang, Y. Luo, Stochastic linear quadratic optimal control for model-free discrete-time systems based on Q-learning algorithm, *Neurocomputing*, **312** (2018), 1–8. https://doi.org/10.1016/j.neucom.2018.04.018

28. F. L. Lewis, D. Vrabie, Reinforcement learning and adaptive dynamic programming for feedback control, *IEEE Circuits Syst. Mag.*, **9** (2009), 32–50. https://doi.org/10.1109/MCAS.2009.933854

AIMS Press