Data-Driven Optimal Tracking Control for Linear Systems Based on Output Feedback Approach

Shikang Chen, Zhenfei Xiao, and Jinna Li(✉)

School of Information and Control Engineering, Liaoning Shihua University, Fushun 113001, Liaoning, China
lijinna_721@126.com

Abstract. In this paper, an off-policy Q-learning method is proposed to solve the linear quadratic tracking problem of discrete-time system based on the output feedback of the system when the system model parameters are unknown. First, a linear discrete-time system with unknown parameters in the system matrix is given. Then, based on the Q-learning method and dynamic programming, an off-policy Q-learning algorithm without knowing system model parameters is proposed, such that the optimal controller is designed to obtain the control strategy which uses the system output data to learn the output feedback data driven optimal tracking control for linear discrete time systems with output feedback. Finally, the simulation results verify the effectiveness of the method.

Keywords: Off-policy Q-learning · LQT problem · Output feedback · Optimal control

1 Introduction

Linear quadratic tracking (LQT) of discrete time systems (DT) is a very important problem in the field of control. The basic idea is that the performance index is a quadratic function defined by the accumulation of deviation between the reference signal and the system output and the accumulation of control input. By designing an optimal controller, the performance index is minimized so that the output of the system can follow the track of the reference signal by an optimal approach. The traditional method to solve LQT problem is to solve a Riccati algebraic equation [1–3]. The traditional controller design methods all require the system model information to be known.

Reinforcement learning (RL) is a one kind of machine learning methods, which was born in 1950s and 1960s [4–6]. During the dynamic interaction with unknown dynamic environment, performance is evaluated and action is updated, such that the optimal performance together with the optimal action can be learned [7–11]. Reinforcement learning has many advantages and strong adaptability. At present, it has become an important learning method for solving optimization problems. It is widely used in robot, artificial intelligence, intelligent systems and other fields, and it is one of the research hot spots in recent years [12–15].

In the existing researches on the optimal tracking control results which do not depend on the system model by reinforcement learning method, most of them use the system state data to learn the state feedback control strategy and track the reference input of the system in the optimal or nearly optimal way, such as the optimal tracking control [16–22], etc. In [19], a Q-learning method based on off policy iteration was proposed to solve the optimal tracking control problem of networked control system. In [20], when the system parameter model is unknown, an optimal control method of linear network control was proposed. According to the state feedback information of the system, a novel off-policy Q-learning algorithm was proposed in [21], which solved the problem of linear quadratic tracking in discrete time under the condition of unknown system parameters. In [22], an optimal tracking control scheme was proposed.

In this paper, a Q-learning algorithm is developed to design the output feedback optimal tracking control strategy, such that the optimal quadratic tracking problem can be solved without the knowledge of system dynamics.

The innovation of this paper lies in (a): different from the traditional research which needs the traditional model information [1–3], the research of this paper is to learn the optimal tracking control strategy when the system model parameters are unknown. (b): compared with other model-free research on the state feedback of the system [20–22], this paper adopts a fully data-driven off policy Q-learning method to solve the linear quadratic tracking control problem of the discrete-time system based on the output feedback of the system and independent of the system model parameters. Finally, simulation experiments and practical application examples are given to verify the effectiveness of the algorithm.

2 On the Optimal Control Problem

This section will introduce the optimal control of linear quadratic tracking problem for discrete-time systems. The state equations of the following linear discrete systems are considered below:

$$\begin{cases} x_{k+1} = Ax_k + Bu_k \\ y_k = Cx_k \end{cases} \tag{1}$$

where x_k is the state of the controlled object, u_k is the input of the controlled object, and y_k is the output of the controlled object. A, B, and C are matrices of appropriate dimensions, respectively. The reference signal of our interest is as follows:

$$r_{k+1} = Fr_k \tag{2}$$

where r_k is the input of the reference object, F is also a matrix with appropriate dimensions. For the linear quadratic tracking problem of discrete-time system, we need to control the output signal y_k in the system (1), and gradually track and catch up with our reference input signal r_k as time goes on. According to the actual problems, we design and select the output feedback controller as follows:

$$u_k = -K_1 y_k - K_2 r_k \tag{3}$$

The purpose of our controller is to optimize performance index J. Our performance indicator is:

$$J = \min_{u_k} \sum_{k=0}^{\infty} [\beta^{\mathrm{k}}(y_k - r_k)^T Q(y_k - r_k) + u_k^T R u_k] \tag{4}$$

where $Q \geq 0$ and $R > 0$ are symmetric matrices, and β is a discount factor with $0 < \beta < 1$. The constraints are as follows:

$$\begin{cases} x_{k+1} = Ax_k + Bu_k \\ y_k = Cx_k \\ r_{k+1} = Fr_k \end{cases} \tag{5}$$

According to the performance index, we can define the optimal value function V^* as:

$$\begin{aligned} V^*(x_k, r_k) &= \min_{u_k} \sum_{i=k}^{\infty} [\beta^{\mathrm{k}}(y_k - r_k)^T Q(y_k - r_k) + u_k^T R u_k] \\ &= \min_{u_k} \sum_{k=0}^{\infty} [\beta^{\mathrm{k}}(Cx_k - r_k)^T Q(Cx_k - r_k) + u_k^T R u_k] \end{aligned} \tag{6}$$

Then, the Q function can be expressed as:

$$Q(x_k,\, r_k,\, u_k) = (y_k - r_k)^T Q(y_k - r_k) + u_k^T R u_k + \sum_{i=k+1}^{\infty} [(y_i - r_i)^T Q(y_i - r_i) + u_i^T R u_i] \tag{7}$$

The optimal function Q^* can be expressed as:

$$Q^*(x_k,\, r_k,\, u_k) = (y_k - r_k)^T Q(y_k - r_k) + u_k^T R u_k + V^*(x_{k+1},\, r_{k+1}) \tag{8}$$

For the convenience of calculation and understanding, (8) can be rewritten as:

$$Q^*(x_k,\, r_k,\, u_k) = \begin{bmatrix} y_k \\ r_k \\ u_k \end{bmatrix}^T \overline{Q} \begin{bmatrix} y_k \\ r_k \\ u_k \end{bmatrix} + V^*(x_{k+1},\, r_{k+1}) \tag{9}$$

where $\overline{Q}$ can be written as:

$$\overline{Q} = \begin{bmatrix} Q & -Q & 0 \\ -Q & Q & 0 \\ 0 & 0 & R \end{bmatrix} \tag{10}$$

From the above formula, we can know that the relationship between the optimal value function V^* and the optimal function Q is:

$$V^*(x_k,\, r_k) \;=\; Q^*(x_k,\, r_k,\, u_k^*) \tag{11}$$

Since the input u_k of the control object is controllable, the optimal value function V^* can be expressed as [15]:

$$V^*(x_k,\, r_k) \;=\; \begin{bmatrix} x_k \\ r_k \end{bmatrix}^T P \begin{bmatrix} x_k \\ r_k \end{bmatrix} \tag{12}$$

The optimal Q function can be expressed as:

$$Q^*(x_k,\, r_k,\, u_k) \;=\; \begin{bmatrix} x_k \\ r_k \\ u_k \end{bmatrix}^T H \begin{bmatrix} x_k \\ r_k \\ u_k \end{bmatrix} \tag{13}$$

The matrix H can be expressed as follows:

$$H = \begin{bmatrix} H_{xx} & H_{xr} & H_{xu} \\ H_{rx} & H_{rr} & H_{ru} \\ H_{ux} & H_{ur} & H_{uu} \end{bmatrix} = \begin{bmatrix} A^T PA + C^T QC & A^T PF - C^T Q & A^T PB \\ F^T PA - QC & F^T PF + Q & F^T PB \\ B^T PA & B^T PF & B^T PB + R \end{bmatrix} \tag{14}$$

According to the necessary conditions to achieve optimal performance, implementing $\frac{\partial Q^*(x_k,\, r_k,\, u_k)}{\partial u_k} = 0$ yields the following forms.

$$\begin{cases} K_1 C \;=\; H_{uu}^{-1} H_{xu}^T \\ K_2 \;=\; H_{uu}^{-1} H_{ru}^T \end{cases} \tag{15}$$

From (15), we find that we cannot get K_1 if the matrix C is unknown. The output equation can be treated as follows.

$$(C^T C)^{-1} C^T y_k \;=\; x_k \tag{16}$$

By substituting (16) into (9) above, we can obtain a new optimal Q function equation:

$$\begin{aligned} Q^*(x_k,\, r_k,\, u_k) \;&=\; (y_k - r_k)^T Q (y_k - r_k) + u_k^T R u_k + \begin{bmatrix} x_{k+1} \\ r_{k+1} \end{bmatrix}^T P \begin{bmatrix} x_{k+1} \\ r_{k+1} \end{bmatrix} \\ &= \begin{bmatrix} y_k \\ r_k \\ u_k \end{bmatrix}^T \begin{bmatrix} Q & -Q & 0 \\ -Q & Q & 0 \\ 0 & 0 & R \end{bmatrix} \begin{bmatrix} y_k \\ r_k \\ u_k \end{bmatrix} + \begin{bmatrix} y_k \\ r_k \\ u_k \end{bmatrix}^{\mathrm{T}} \begin{bmatrix} (C^T C)^{-1} C^T & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & I \end{bmatrix}^{\mathrm{T}} \begin{bmatrix} A & 0 & B \\ 0 & F & 0 \end{bmatrix}^T P \\ &\times \begin{bmatrix} A & 0 & B \\ 0 & F & 0 \end{bmatrix} \begin{bmatrix} (C^T C)^{-1} C^T & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & I \end{bmatrix} \begin{bmatrix} y_k \\ r_k \\ u_k \end{bmatrix} = \begin{bmatrix} y_k \\ r_k \\ u_k \end{bmatrix}^T \begin{bmatrix} \overline{H}_{yy} & \overline{H}_{yr} & \overline{H}_{yu} \\ \overline{H}_{ry} & \overline{H}_{rr} & \overline{H}_{ru} \\ \overline{H}_{uy} & \overline{H}_{ur} & \overline{H}_{uu} \end{bmatrix} \begin{bmatrix} y_k \\ r_k \\ u_k \end{bmatrix} \\ &= Z_k^T \overline{H} Z_k \end{aligned} \tag{17}$$

where

$$\overline{H} = \begin{bmatrix} \overline{H}_{yy} & \overline{H}_{yr} & \overline{H}_{yu} \\ \overline{H}_{ry} & \overline{H}_{rr} & \overline{H}_{ru} \\ \overline{H}_{uy} & \overline{H}_{ur} & \overline{H}_{uu} \end{bmatrix}$$
$$= \begin{bmatrix} [C^T(CC^T)^{-1}]^T(A^TPA)[C^T(CC^T)^{-1}] + Q & (C^TC)^{-1}C^TA^TPF - Q & (C^TC)^{-1}C^TA^TPB \\ (F^TPA)[C^T(CC^T)^{-1}] - Q & F^TPF + Q & F^TPB \\ B^TPA[C^T(CC^T)^{-1}] & B^TPF & B^TPB + R \end{bmatrix} \tag{18}$$

Implementing $\frac{\partial Q^*(x_k, r_k, u_k)}{\partial u_k} = 0$ yields the following forms

$$\begin{cases} K_1 = \overline{H}_{uu}^{-1}\overline{H}_{yu}^T \\ K_2 = \overline{H}_{uu}^{-1}\overline{H}_{ru}^T \end{cases} \tag{19}$$

According to the above relation, the Riccati equation for the optimal Q function is as follows:

$$Z_k^T\overline{H}Z_k = Z_k^T\overline{Q}Z_k + Z_{k+1}^T\overline{H}Z_{k+1} \tag{20}$$

3 Data Driven Q-Learning Algorithm

This section will give off-policy Q-learning algorithm for designing the output feedback optimal tracking control strategy, under which the system output can track the reference signal in an approximate optimal way.

First, the on-policy Q-learning algorithm is given, and then based on it the off-policy Q-learning algorithm is derived.

Algorithm 1: On-policy Q-learning algorithm.

1. Give a stablizing controller gain K_1^j and K_2^j, let the initial j value be 0, j represents the number of iterations. The control object input is defined as.

$$u_k^j = -K_1^j y_k - K_2^j r_k \tag{21}$$

2. Evaluate the control policy by solving the optimal Q function and matrix $\overline{H}$.

$$Z_k^T\overline{H}^{j+1}Z_k = (y_k - r_k)^T Q (y_k - r_k) + u_k^j R u_k^j + Z_{k+1}^T\overline{H}^{j+1}Z_{k+1} \tag{22}$$

3. Update the control policy.

$$u_k^{j+1} = -K_1^{j+1} y_k - K_2^{j+1} r_k$$
$$\begin{cases} K_1^{j+1} = (\bar{H}_{uu}^{j+1})^{-1} (\bar{H}_{yu}^{j+1})^T \\ K_2^{j+1} = (\bar{H}_{uu}^{j+1})^{-1} (H_{ru}^{j+1})^T \end{cases} \tag{23}$$

4. If $\left\|\bar{H}^{j+1} - \bar{H}^j\right\| \le \varepsilon$($\varepsilon$ is a small positive number) stops the iteration of the strategy. Otherwise,$j = j+1$, and return to Step 2.

In view of the advantages of off policy Q-learning algorithm, we will propose an off-policy algorithm based on Q-function, and adopt data-driven algorithm without model to solve the linear quadratic tracking problem of discrete-time system. We introduce the target control strategy into the system dynamics and get the following equation, where u_k is the behavior control strategy and u_k^j is the target control strategy.

$$y_{k+1} = Cx_{k+1} = CAx_k + CBu_k = CAC^T(CC^T)^{-1} y_k + CBu_k \tag{24}$$

$$\begin{bmatrix} y_{k+1} \\ r_{\mathrm{k}+1} \end{bmatrix} = \begin{bmatrix} CAC^T(CC^T)^{-1} & 0 \\ 0 & F \end{bmatrix} \times \begin{bmatrix} y_k \\ r_k \end{bmatrix} + \begin{bmatrix} CB \\ 0 \end{bmatrix} \times u_k^j + \begin{bmatrix} CB \\ 0 \end{bmatrix} \times (u_k - u_k^j) \tag{25}$$

From Eq. (22), one has

$$\begin{aligned}
&\begin{bmatrix} y_k \\ r_k \end{bmatrix}^T P^{j+1} \begin{bmatrix} y_k \\ r_k \end{bmatrix} - \begin{bmatrix} y_{k+1} \\ r_{\mathrm{k}+1} \end{bmatrix}^T P^{j+1} \begin{bmatrix} y_{k+1} \\ r_{\mathrm{k}+1} \end{bmatrix} + 2 \begin{bmatrix} y_{k+1} \\ r_{\mathrm{k}+1} \end{bmatrix}^T P^{j+1} \begin{bmatrix} CB \\ 0 \end{bmatrix} \\
&\times \begin{bmatrix} y_k \\ r_k \end{bmatrix}^T \times \begin{bmatrix} \mathrm{I} & 0 \\ 0 & \mathrm{I} \\ -K_1^j & -K_2^j \end{bmatrix}^T \overline{H}^{\mathrm{j}+1} \begin{bmatrix} \mathrm{I} & 0 \\ 0 & \mathrm{I} \\ -K_1^j & -K_2^j \end{bmatrix} \times \begin{bmatrix} y_k \\ r_k \end{bmatrix} \\
&- \left(\begin{bmatrix} y_{k+1} \\ r_{\mathrm{k}+1} \end{bmatrix} - \begin{bmatrix} CB \\ 0 \end{bmatrix} \times (u_k - u_k^j) \right)^T \times \begin{bmatrix} \mathrm{I} & 0 \\ 0 & \mathrm{I} \\ -K_1^j & -K_2^j \end{bmatrix}^T \overline{H}^{\mathrm{j}+1} \\
&\times \begin{bmatrix} \mathrm{I} & 0 \\ 0 & \mathrm{I} \\ -K_1^j & -K_2^j \end{bmatrix} \left(\begin{bmatrix} y_{k+1} \\ r_{\mathrm{k}+1} \end{bmatrix} - \begin{bmatrix} CB \\ 0 \end{bmatrix} \times (u_k - u_k^j) \right) \\
&= (y_k - r_k)^T Q (y_k - r_k) + (u_k^j)^T R u_{\mathrm{k}}^j
\end{aligned} \tag{26}$$

where

$$P^{j+1} = \begin{bmatrix} I & 0 \\ 0 & I \\ -K_1^j & -K_2^j \end{bmatrix}^T \overline{H}^{\mathrm{j}+1} \begin{bmatrix} I & 0 \\ 0 & I \\ -K_1^j & -K_2^j \end{bmatrix} \tag{27}$$

(26) can be rewritten as:

$$\begin{bmatrix} y_k \\ r_k \end{bmatrix}^T P^{j+1} \begin{bmatrix} y_k \\ r_k \end{bmatrix} - \begin{bmatrix} y_{k+1} \\ r_{\mathrm{k}+1} \end{bmatrix}^T P^{j+1} \begin{bmatrix} y_{k+1} \\ r_{\mathrm{k}+1} \end{bmatrix} + 2 \begin{bmatrix} y_{k+1} \\ r_{\mathrm{k}+1} \end{bmatrix}^T P^{j+1} \begin{bmatrix} CB \\ 0 \end{bmatrix} \times (u_k - u_k^j)$$
$$- (u_k - u_k^j)^T (CB)^T P^{j+1} (CB)(u_k - u_k^j) = (y_k - r_k)^T Q (y_k - r_k) + (u_k^j)^T R u_{\mathrm{k}}^j \tag{28}$$

where

$$2\begin{bmatrix} y_{k+1} \\ r_{\mathrm{k}+1} \end{bmatrix}^T P^{j+1} \begin{bmatrix} CB \\ 0 \end{bmatrix} \times (u_k - u_k^j) = 2\begin{bmatrix} CAC^T(CC^T)^{-1} y_k + CBu_k \\ Fr_k \end{bmatrix}^T P^{j+1}$$
$$\times \begin{bmatrix} CB \\ 0 \end{bmatrix} (u_k - u_k^j) = 2y_k^T [C^T(CC^T)^{-1}]^T (CA)^T P^{j+1} CB(u_k - u_k^j)$$
$$+ 2u_k^T (CB)^T P^{j+1} CB(u_k - u_k^j) + 2r_k^T F^T P^{j+1} CB(u_k - u_k^j) \tag{29}$$

Further, one has

$$\begin{bmatrix} y_k \\ r_k \end{bmatrix}^T P^{j+1} \begin{bmatrix} y_k \\ r_k \end{bmatrix} - \begin{bmatrix} y_{k+1} \\ r_{\mathrm{k}+1} \end{bmatrix}^T P^{j+1} \begin{bmatrix} y_{k+1} \\ r_{\mathrm{k}+1} \end{bmatrix}$$
$$+ 2y_k^T [C^T(CC^T)^{-1}]^T (CA)^T P^{j+1} CB(u_k - u_k^j) + 2u_k^T (CB)^T P^{j+1} CB(u_k - u_k^j)$$
$$+ 2r_k^T F^T P^{j+1} CB(u_k - u_k^j) - (u_k - u_k^j)^T (CB)^T P^{j+1} (CB)(u_k - u_k^j)$$
$$= (y_k - r_k)^T Q (y_k - r_k) + (u_k^j)^T R u_{\mathrm{k}}^j \tag{30}$$

Rewriting (30) yields below

$$\theta_k^j L^{j+1} = \rho_k^j \tag{31}$$

where

$$\rho_k^j = (y_k - r_k)^T Q (y_k - r_k) + (u_k)^T R u_k$$

$$L^{j+1} = \Big[(vec(L_1^{j+1}))^T \ (vec(L_2^{j+1}))^T \quad (vec(L_3^{j+1}))^T$$
$$(vec(L_4^{j+1}))^T \quad (vec(L_5^{j+1}))^T \quad (vec(L_6^{j+1}))^T \Big]^{\mathrm{T}}$$

$L_1^{j+1} = \overline{H}_{yy}^{j+1}$, $L_2^{j+1} = \overline{H}_{yr}^{j+1}$, $L_3^{j+1} = \overline{H}_{yu}^{j+1}$, $L_4^{j+1} = \overline{H}_{rr}^{j+1}$, $L_5^{j+1} = \overline{H}_{ru}^{j+1}$, $L_6^{j+1} = \overline{H}_{uu}^{j+1}$,

$$\theta_k^j = \begin{bmatrix} \theta_1^j & \theta_2^j & \theta_3^j & \theta_4^j & \theta_5^j & \theta_6^j \end{bmatrix}$$

$$\theta_1^{\mathrm{j}} = y_k^T \otimes y_k^T - y_{k+1}^T \otimes y_{k+1}^T$$

$$\theta_2^j = 2y_k^T \otimes y_k^T - 2y_{k+1}^T \otimes r_{k+1}^T$$

$$\theta_3^j = 2y_k^T \otimes \mathrm{u}_k^T - 2y_{k+1}^T \otimes (u_{k+1}^j)^T$$

$$\theta_4^j = r_k^T \otimes r_k^T - r_{k+1}^T \otimes r_{k+1}^T$$

$$\theta_5^j = 2\mathrm{r}_k^T \otimes u_k^{\mathrm{T}} - 2\mathrm{r}_{k+1}^T \otimes (u_{k+1}^j)^T$$

$$\theta_6^j = \mathrm{u}_k^T \otimes \mathrm{u}_k^T - (u_{k+1}^j)^T \otimes (u_{k+1}^j)^T$$

4 Simulation Experiment

In this section, we use the proposed algorithm to simulate the experiment, and use the experimental results to verify whether the algorithm is effective.

Example 1: Consider the following system:

$$x_{k+1} = \begin{bmatrix} -1 & 2 \\ 2.2 & 1.7 \end{bmatrix} x_k + \begin{bmatrix} 2 \\ 1.6 \end{bmatrix} u_k \tag{32}$$

$$y_k = \begin{bmatrix} 1 & -2 \\ -1 & 4 \end{bmatrix} x_k \tag{33}$$

The reference signal generator is:

$$r_{k+1} = \begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix} r_k \tag{34}$$

Choose $Q = \begin{bmatrix} 1000 & 0 \\ 0 & 10 \end{bmatrix}$ and $R = 1$. The optimal matrix H and $\overline{H}$ can be obtained from (13) and (17), respectively, and the control gain K_1 and K_2 of the optimal tracking control can be obtained from (19).

$$H = \begin{bmatrix} 55014.1382 & -47201.9872 & -10241.4410 & 193.0046 & -48231.9264 \\ -47201.9872 & 124780.7167 & 9344.6976 & -162.08466 & 124282.7373 \\ -10241.4410 & 9344.6976 & 2067.8905 & 51.2434 & 9527.2908 \\ 193.0046 & -162.08466 & 51.2434 & 49.1044 & -166.5050 \\ -48231.9264 & 124282.7373 & 9527.2908 & -166.5050 & 123826.6900 \end{bmatrix} \tag{35}$$

$$\overline{H} = \begin{bmatrix} 157847.7575 & 70420.4747 & -16810.5331 & 304.5860 & -34322.4842 \\ 70420.4747 & 39017.3301 & -5569.0922 & 101.5813 & 13909.4422 \\ -16810.5331 & -5569.0922 & 3067.8905 & 51.2434 & 9527.2908 \\ 304.5860 & 101.5813 & 51.2434 & 59.1044 & -166.5050 \\ -34322.4842 & 13909.4422 & 9527.2908 & -166.5050 & 123826.6900 \end{bmatrix} \tag{36}$$

$$\begin{cases} K_1 = [0.2772 \quad -0.1123] \\ K_2 = [0.0769 \quad 0.0013] \end{cases} \tag{37}$$

After 8 iterations, we find that the algorithm converges and the matrix $\overline{H}^8$ and the control gain K_1^8 and K_2^8 of the optimal tracking control are the following data.

$$\overline{H}^8 = \begin{bmatrix} 157847.7575 & 70420.4747 & -16810.5331 & 304.5860 & -34322.4842 \\ 70420.4747 & 39017.3301 & -5569.0922 & 101.5813 & 13909.4422 \\ -16810.5331 & -5569.0922 & 3067.8905 & 51.2434 & 9527.2908 \\ 304.5860 & 101.5813 & 51.2434 & 59.1044 & -166.5050 \\ -34322.4842 & 13909.4422 & 9527.2908 & -166.5050 & 123826.6900 \end{bmatrix} \tag{38}$$

$$\begin{cases} K_1^8 = [0.2772 \quad -0.1123] \\ K_2^8 = [0.0769 \quad 0.0013] \end{cases} \tag{39}$$

In the learning process, the optimal tracking controller gain is convergent, and the following figure shows its convergence process in the learning process (Fig. 1).

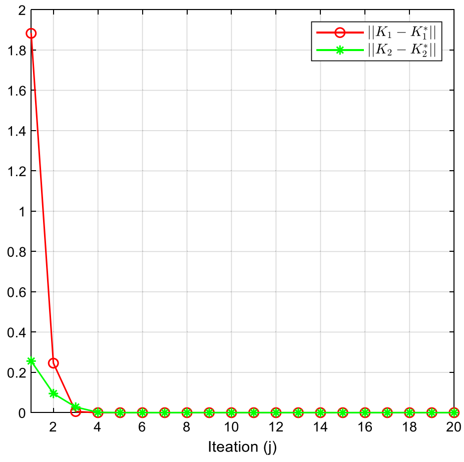

Fig. 1. Optimal control gain convergence process of tracking controller

In order to obtain the exact solution of Riccati Eq. (20) of the optimal Q-function under sufficient excitation conditions, it is necessary to add detection noise (Figs. 2 and 3).

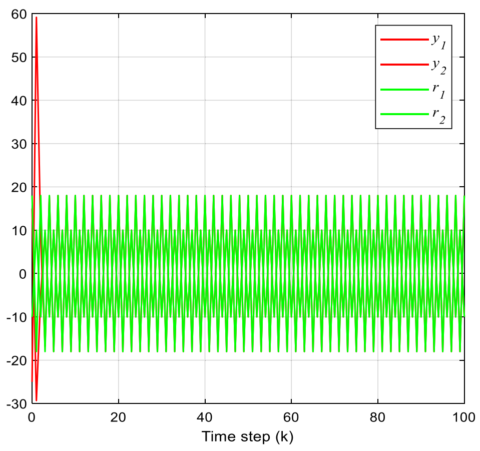

Fig. 2. System output and reference signal

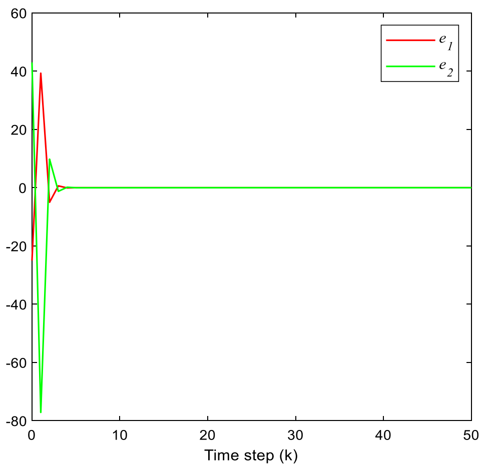

Fig. 3. Tracking errors using the learned controller

Example 2:
We select the water tank system made by ingenieurburo gurski Schramm company in Germany, which is a three tank water tank of TTS20 type, as our simulation object, and the water tank is shown in Fig. 4. This water tank system consists of a nonlinear multi input multi output system with two actuators and a digital controller, which meets our requirements for the system. The main structure and overall industrial process of TTS20 three tank water tank are shown in Fig. 5.

Fig. 4. Three tank of TTS20

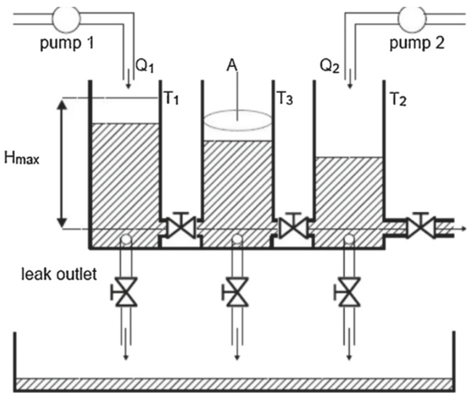

Fig. 5. Structure and industrial process of TTS20

TTS20 three tank device is composed of three plexiglass cylinders T1, T2 and T3 with Section A, which are connected in series with each other through cylindrical pipes with section Sn. There is a one-way valve in T2 glass pipe, and the outflow liquid will be collected in a reservoir to provide water for pump 1 and pump 2. $H_{\max}$ is the highest liquid level. If the level of T1 or T2 exceeds this value, the corresponding pumps 1 and 2 will automatically shut down. Q1 and Q2 represent the flow of pump 1 and pump 2. In addition, to simulate leakage, each tank has a circular opening with a manually adjustable ball valve on the cross section. The drain valve and leakage flow can describe the failure information of the water tank. The liquid extracted from the pool is injected into T1 and T2 by pump 1 (P1) and pump 2 (P2), respectively. Then their bottom valve and T3 drain valve discharge water into the reservoir for P1 and P2 recycling, forming a circuit. Among them, T1, T2 and T3 are measured by three pressure level sensors as the measuring elements of the system, and the flow of Q1 and Q2 is regulated by the digital controller.

For TTS20, we can design its model as follows:

$$\begin{cases} \begin{bmatrix} \overset{\bullet}{h}_1 \\ \overset{\bullet}{h}_3 \end{bmatrix} = \frac{1}{s}\begin{bmatrix} -Q_{13} \\ -Q_{13} - Q_{out} \end{bmatrix} Q_{in} \\ y = h_1 \end{cases} \tag{40}$$

In the model, h_1 and h_3 are the control variables, representing the water level height of water tanks T1 and T3, Q_n is selected as the control variable of the system as the flow of Q1, and the flow from T1 to T3 is $Q_{13} = az_1 S_n \mathrm{sgn}(h_1 - h_3)\sqrt{2g|h_1 - h_3|}$, the water flow from the bottom of T3 is $Q_{out} = az_2 S_1 \sqrt{2gh_2}$, $S_1 = S_n = 5 \times 10^{-5}\,\mathrm{m}^2$, $S = 0.154\,\mathrm{m}^2$,$H_{\max} = 0.6\,\mathrm{m}$, Flow coefficient $az_1 = 0.48$, $az_2 = 0.58$, sgn($\bullet$) is a symbolic function. We set the initial value of h_1 and h_3 is 0, and the relationship between state variable and input variable is $\begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix} = \begin{bmatrix} h_1(k) \\ h_3(k) \end{bmatrix}$, $u(k) = Q_{in}(k)$. The state space model of TTS20 is as follows:

$$x_{k+1} = \begin{bmatrix} 0.9850 & 0.0107 \\ 0.0078 & 0.9784 \end{bmatrix} x_k + \begin{bmatrix} 64.4453 \\ 0.2559 \end{bmatrix} u_k \tag{41}$$

$$y_k = [1 \quad 0] x_k \tag{42}$$

The reference signal generator is:

$$r_{k+1} = r_k \tag{43}$$

Select the value of reference signal water level as 0.5 m. Choose $Q = 10$ and $R = 5$, The optimal Q- function matrix H and $\overline{H}$ are obtained, The control gain K_1 and K_2 of the optimal tracking control are as follows:

$$H = \begin{bmatrix} 17.7618 & -17.6205 & 507.6889 \\ -17.6205 & -131.2013 & -514.5708 \\ 507.6889 & -514.5708 & 33224.5750 \end{bmatrix} \tag{44}$$

$$\overline{H} = \begin{bmatrix} 17.7627 & -17.8809 & 507.8883 \\ -17.8809 & 18.0010 & -515.6235 \\ 507.8883 & -515.6235 & 33234.4549 \end{bmatrix} \tag{45}$$

$$\begin{cases} K_1 = [-0.0153] \\ K_2 = [0.0155] \end{cases} \tag{46}$$

After 10 iterations, we find that the algorithm converges and the optimal Q-function matrix $\overline{H}^{10}$ and the gain K_1^{10} and K_2^{10} of the optimal tracking control are the following data.

$$\overline{H}^{10} = \begin{bmatrix} 17.7627 & -17.8809 & 507.8883 \\ -17.8809 & 18.0010 & -515.6235 \\ 507.8883 & -515.6235 & 33234.4549 \end{bmatrix} \tag{47}$$

$$\begin{cases} K_1^{10} = [-0.0153] \\ K_2^{10} = [0.0155] \end{cases} \tag{48}$$

We find that in the learning process, the optimal tracking controller gain is convergent, and the following figure will show its convergence process in the learning process (Figs. 6, 7 8 and 9).

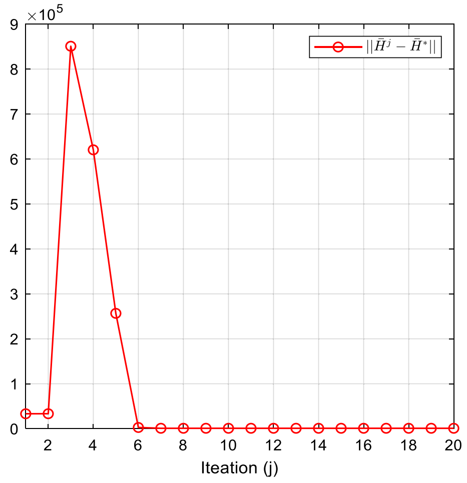

Fig. 6. Optimal $\overline{H}$ matrix convergence process

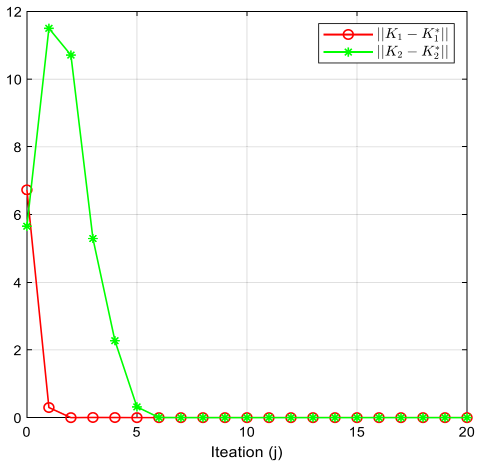

Fig. 7. Optimal control gain convergence process of tracking controller

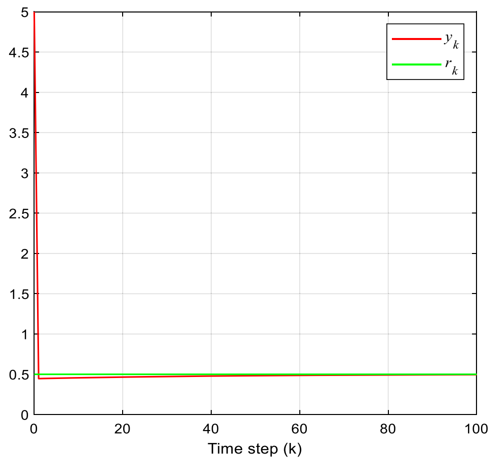

Fig. 8. Output trajectories of system

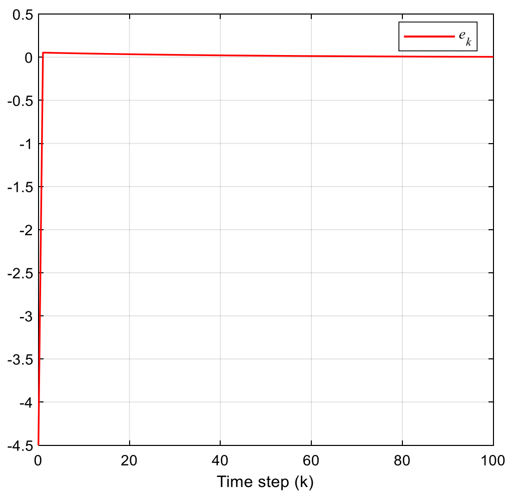

Fig. 9. Tracking errors using the learned controller

5 Conclusion

In this paper, a data-driven off policy Q-learning method is proposed to solve the linear quadratic tracking problem of discrete-time system based on the output feedback of the system. This paper introduces and compares the on policy Q-learning method and the off policy Q-learning method for the linear quadratic tracking problem of the discrete-time system, combines the dynamic programming with the Q-learning method, and uses the off policy Q-learning method to learn the optimal controller gain when the system environment is unknown. Finally, the simulation results show that the method is effective.

Acknowledgments. This work was supported by the National Natural Science Foundation of China under Grants 61673280, the Open Project of Key Field Alliance of Liaoning Province under Grant 2019-KF-03-06 and the Project of Liaoning Shihua. University under Grant 2018XJJ-005.

References

1. Lewis, F.L., Vrabie, D.L., Syrmos, V.L.: Optimal Control for Polynomial Systems. pp. 287–296, Wiley (2012)
2. Hengster-Movric, K., You, K., Lewis, F.L., Xie, L.: Synchronization of discrete-time multi-agent systems on graphs using riccati design. Automatica **49**(2), 414–423 (2013)
3. Stoorvogel, A.A., Weeren, A.J.T.M.: The discrete-time riccati equation related to the H∞ control problem. IEEE Trans. Autom. Control **39**(3), 686–691 (1994)
4. Kaelbling, L.P., Littman, M.L., Moore, A.W.: Reinforcement learning: a survey. J. Artif. Intell. Res. **4**(1), 237–285 (1996)

5. Sutton, R.S., Barto, A.G.: Reinforcement Learning: An Introduction. MIT Press, Cambridge (1998)
6. Li, J., Ding, J., Chai, T., Lewis, F.L.: Nonzero-sum game reinforcement learning for performance optimization in large-scale industrial processes. IEEE Trans. Cybern. **50**(9), 4132–4145 (2020)
7. Sutton, R.: Learning to predict by the methods of temporal difference. Mach. Learn. **3**(1), 9–44 (1988)
8. Bertsekas, D.P., Tsitsiklis, J.N., Volgenant, A.: Neuro-dynamic programming. Encycl. Optim. **27**(6), 1687–1692 (2011)
9. Santamaria, J.C., Sutton, R., Ram, A.: Experiments with reinforcement learning in problems with continuous state and action spaces. Adap. Behav. **6**, 163–217 (1997)
10. Watkins, C., Dayan, P.: Q-learning. Mach. Learn. **8**, 279–292 (1992)
11. Wang, D., Liu, D.: Learning and guaranteed cost control with event-based adaptive critic implementation. IEEE Trans. Neural Netw. Learn. Syst. **29**(12), 6004–6014 (2018)
12. Smart, W.D., Kaelbling, L.P.: Effective reinforcement learning for mobile robots. In: Proceedings of the IEEE International Conference on Robotics and Autoinforcement learning for performance optimization, pp. 3404–3410 (2002)
13. Beom, H.R., Cho, H.S.: A sensor-based navigation for a mobile robot using fuzzy logic and reinforcement learning. IEEE Trans. Syst. Man Cybern. **25**, 464–477 (1995)
14. Kondo, T., Ito, K.: A reinforcement learning with evolutionary state recruitment strategy for autonomous mobile robots control. Robot. Auton. Syst. **46**(2), 111–124 (2004)
15. Li, J.N., Ding, J.L., Chai, T.Y, Li, C., Lewis, F.L.: Nonzero-sum game reinforcement learning for performance optimization in large-scale industrial processes. IEEE Trans. Cybern. (2019). https://doi.org/10.1109/tcyb.2019.2950262
16. Kiumarsi, B., Lewis, F.L., Jiang, Z.P.: H∞ control of linear discrete-time systems: off-policy reinforcement learning. Automatic **37**(1), 144–152 (2017)
17. Kim, J.H., Lewis, F.L.: Model-free H∞ control design for unknown linear discrete-time systems via Q-learning with LMI. Automatica **46**(8), 1320–1326 (2010)
18. Al-Tamimi, A., Lewis, F.L., Abu-Khalaf, M.: Model-free Q-learning designs for linear discrete-time zero-sum games with application to H-infinity control. Automatica **43**(3), 473–481 (2007)
19. Li, J.N., Yin, Z.X.: Optimal tracking control of networked control systems based on off policy Q-learning. Control Decis. **34**(11), 2343–2349 (2019)
20. Xu, H., Sahoo, A., Jagannathan, S.: Stochastic adaptive event-triggered control and network scheduling protocol co-design for distributed networked systems. Control Theory Appl. IET. **8**(18), 2253–2265 (2014)
21. Li, J.N., Chai, T.Y., Lewis, F.L., Ding, Z.T., Jiang, Y.: Off-policy interleaved Q-learning: optimal control for affine nonlinear discrete-time systems. IEEE Trans. Neural Netw. Learn. Syst. **30**(5), 1308–1320 (2019)
22. Li, X.F., Xue, L., Sun, C.Y.: Linear quadratic tracking control of unknown discrete-time systems using value iteration algorithm. Neurocomputing **314**(7), 86–93 (2018)