21st International Symposium on
Mathematical Theory of Networks and Systems
July 7-11, 2014. Groningen, The Netherlands

Q-learning-based optimal digital feedback control with computation time delay of linear discrete-time systems

Taishi Fujita[1] and Toshimitsu Ushio[2]

Abstract— **In embedded computers, there are delays due to computation time. Unless they are considered, a controlled system may be unstable. If the system is unknown, Q-learning-based optimal control is one of the useful approaches. Applying it to a system, we can obtain the optimal feedback gain for the unknown system. In this paper, we propose Q-learning-based optimal feedback control taking the delay into consideration. First, we assume that all states can be observed and consider a state feedback controller. The input at the next time is estimated by the current state and the input. Then an optimal feedback gain is provided by sequences of pairs of the state and the input. Next, we show an output feedback controller. If the system is observable, the state can be reconstructed by the last L input and output data, where L is an integer equal to or larger than the observability index of the system. An optimal feedback gain is provided by these sequences. Finally, we apply the proposed adaptive state feedback controller to a quadrotor and show its efficiency by simulation.**

Key words. Optimal control, Reinforcement learning, Discrete time linear systems.

AMS subject classifications. 93B15, 92B52, 93C40.

I. Introduction

Reinforcement learning (RL) is a method for an agent to obtain an optimal policy adaptively by the interaction with its environment, from which the agent receives rewards for its actions. The agent does not have any mathematical model for the environment. It selects a policy to interact with its environment, gets rewards from the environment, and changes its policy so as to optimize the cumulative reward. The agent repeats such actions many times and searches the optimal policy by reinforcing that by which it gets better rewards [1]. Q-learning is one of the RL methods based on dynamic programming. It was shown that a sequence of policies selected by the agent converges to an optimal one by using Q-learning [2], [3]. Q-learning has been applied in many engineering fields such as robotics, games, and multi-agent systems, and is also connected with psychology and neuroscience [4], [5].

Originally, RL was developed by the computational intelligence community, independently of the control engineering community. Recently, however, it has been applied to the design of optimal controllers for systems with unknown dynamics [6], [7], [8], [9]. In general, many adaptive control methods developed in the control engineering community assume basic structures of the systems with unknown parameters [10], while Q-learning-based adaptive control is model-free.

In applications of Q-learning to optimal control problems for discrete-time systems with a cost function given by the sum of a function of the states and the inputs at each time, the control policy and the reward correspond to the (static) feedback control law and the cost at each time, respectively, so that the cumulative reward is the cost function [6]. Thus, we can search the optimal feedback control law which minimizes the cost function. Q-learning can be applied not only to discrete-time systems, but also to continuous-time systems [7], [11], [12], [13]. Q-learning based control can achieve the optimal feedback law without knowing the structure of the system, and can adapt to changes in the system's parameters. The control law is generally a function of the state of the system. In the case of partial observation of a state, that is, of the control using the outputs of a plant, we cannot use an observer, which is a standard approach to the estimation of the state, since the observer is a dynamic system. For applying Q-learning to the output feedback control of linear discrete-time systems, the data-based control scheme using Markov parameters of the system is useful [14], [15], [16]. Lewis and Vamvoudakis proposed a Q-learning-based output feedback control method where the data-based approach is utilized for the estimation of the state [17].

On the other hand, due to the recent development of information and network technology, embedded control systems where control data are transmitted using networks have been receiving a lot of attention [18]. In embedded control systems, there is a time delay caused by the computation of updated control inputs in the embedded computer and the transmission of the output and the input data in the network. In this paper, for simplicity, we will call such a delay a computation time delay. The delay degrades the control performances in general and may make the controlled system unstable in the worst case [19]. Mita proposed a design method of an optimal digital state feedback controller for a linear discrete-time system, where the computation time delay is taken into consideration [20]. The optimal control of non linear systems with delays in both the states and the inputs using adaptive dynamic programming has been studied in [9]. To the best of our knowledge, however, a design method of a Q-learning-based optimal controller where the computation time delay is taken into account has not been studied yet. This paper deals with both state feedback and output feedback optimal control under the existence of a computation time delay. Then, we apply the proposed state feedback controller to the hovering control of a quadrotor.

[1]T. Fujita and [2]T. Ushio are with the Graduate School of Engineering Science, Osaka University, 1-3 Machikaneyama, Toyonaka, Osaka, 560-8531, Japan

ISBN: 978-90-367-6321-9

This paper is organized as follows. Section II explains Q-learning-based state feedback control. The updated control input is computed based on the state and the current control input. The feedback gain is obtained by Q-learning. Section III deals with the output feedback control. We use the last L outputs and inputs data, where L is a sufficiently large integer for the data-based estimation of the state of the system. Thus, the control input is determined by the data. Section IV applies the proposed state feedback control to the hovering control of a quadrotor to demonstrate its efficiency. Section V concludes the paper.

II. Q-LEARNING-BASED OPTIMAL STATE FEEDBACK CONTROL WITH A DELAY

We consider a discrete-time linear system described by

$$x_{k+1} = Ax_k + Bu_k, \tag{1}$$
$$y_k = Cx_k, \tag{2}$$

where $x_k \in R^n$, $u_k \in R^m$, and $y_k \in R^p$ are a state, an input, and an output at time k, respectively, and A, B, and C are matrices of dimensions $n \times n$, $n \times m$, and $p \times n$, respectively. We assume that (A, B) is controllable and (C, A) is observable.

We consider state feedback control keeping in mind the delay in the control-loop. This delay is caused by the computation in an embedded computer and the data transmission. For simplicity, we assume that the delay is one unit time. Then, the control input u_k is not computed using x_k, but x_{k-1}. Moreover, since the input u_k is stored in the controller, we can use it if its usage improves the control performance. First, we consider the case where the matrices A, B, and C are known. Therefore, we can use these matrices to compute an input. The input is computed by the following control policy:

$$u_k = -F \begin{bmatrix} x_{k-1} \\ u_{k-1} \end{bmatrix}, \tag{3}$$

where F is an $m \times (n+m)$ feedback gain matrix. We set $\hat{x}_k = [x_k^T,\ u_k^T]^T$ as an extended state. Then, we rewrite Eqs. (1), (2) as

$$\hat{x}_{k+1} = \hat{A}\hat{x}_k + \hat{B}v_k, \tag{4}$$
$$y_k = \hat{C}\hat{x}_k, \tag{5}$$
$$v_k = -F\hat{x}_k, \tag{6}$$

where $v_k = u_{k+1}$ is an input to the extended state equation, and the matrices $\hat{A}$, $\hat{B}$, and $\hat{C}$ are defined by

$$\hat{A} = \begin{bmatrix} A & B \\ 0 & 0 \end{bmatrix}, \quad \hat{B} = \begin{bmatrix} 0 \\ I_m \end{bmatrix}, \quad \hat{C} = \begin{bmatrix} C & 0 \end{bmatrix}. \tag{7}$$

We define a cost function as follows:

$$V_F(\hat{x}_k) = \sum_{i=k}^{\infty} \left(x_i^T Q_s x_i + v_i^T R_s v_i \right), \tag{8}$$

where Q_s and R_s are the weight matrices. We assume that $Q_s = Q_s^T \geq 0$ and $R_s = R_s^T > 0$. Then, we obtain the following Bellman equation:

$$V_F(\hat{x}_k) = x_k^T Q_s x_k + v_k^T R_s v_k + V_F(\hat{x}_{k+1}). \tag{9}$$

The optimal control policy v_k^* is given by

$$v_k^* = -(R_s + B^T P_s B)^{-1} B^T P_s A \begin{bmatrix} A & B \end{bmatrix} \hat{x}_k, \tag{10}$$

where P_s is an $n \times n$ positive definite symmetric matrix which is a solution of the following Riccati equation:

$$Q_s + A^T P_s A - A^T P_s B (R_s + B^T P_s B)^{-1} B^T P_s A = P_s. \tag{11}$$

The optimal cost function is rewritten as follows:

$$V^*(\hat{x}_k) = \hat{x}_k^T \bar{P}_s \hat{x}_k, \tag{12}$$

where $\bar{P}_s$ is defined by

$$\bar{P}_s = \begin{bmatrix} Q_s + A^T P_s A & A^T P_s B \\ B^T P_s A & B^T P_s B \end{bmatrix}. \tag{13}$$

This result is the same as that in [20].

Next, we consider the case where the matrices A, B, and C are unknown. Namely, a feedback gain is learned by sequences of states and inputs. Then, we obtain the optimal gain using Q-learning. From Eq. (12), we can assume that $V_F(\hat{x}_k)$ is written as follows:

$$V_F(\hat{x}_k) = \hat{x}_k^T \hat{P} \hat{x}_k, \tag{14}$$

where $\hat{P}$ is an $(n+m)\times(n+m)$ positive definite symmetric matrix. Then, we denote the right hand side of (9) by Q_F.

$$\begin{aligned} Q_F(\hat{x}_k, v_k) &= x_k^T Q_s x_k + v_k^T R_s v_k + V_F(\hat{x}_{k+1}) \\ &= x_k^T Q_s x_k + v_k^T R_s v_k \\ &\quad + (\hat{A}\hat{x}_k + \hat{B}v_k)^T \hat{P} (\hat{A}\hat{x}_k + \hat{B}v_k) \\ &= \begin{bmatrix} \hat{x}_k \\ v_k \end{bmatrix}^T \begin{bmatrix} H_{F(\hat{x}\hat{x})} & H_{F(\hat{x}v)} \\ H_{F(v\hat{x})} & H_{F(vv)} \end{bmatrix} \begin{bmatrix} \hat{x}_k \\ v_k \end{bmatrix} \\ &= \begin{bmatrix} \hat{x}_k \\ v_k \end{bmatrix}^T H_F \begin{bmatrix} \hat{x}_k \\ v_k \end{bmatrix}, \end{aligned} \tag{15}$$

where

$$H_{F(\hat{x}\hat{x})} = \begin{bmatrix} Q_s & 0 \\ 0 & 0 \end{bmatrix} + \hat{A}^T \hat{P} \hat{A}, \tag{16}$$
$$H_{F(\hat{x}v)} = \hat{A}^T \hat{P} \hat{B}, \tag{17}$$
$$H_{F(v\hat{x})} = H_{F(\hat{x}v)}^T, \tag{18}$$
$$H_{F(vv)} = R_s + \hat{B}^T \hat{P} \hat{B}. \tag{19}$$

Using $Q_F(\hat{x}_k, v_k)$, we rewrite the Bellman equation (9) as follows:

$$Q_F(\hat{x}_k, v_k) = x_k^T Q_s x_k + v_k^T R_s v_k + Q_F(\hat{x}_{k+1}, -F\hat{x}_{k+1}). \tag{20}$$

H_F is estimated by using the least-squares method or the recursive least-squares (RLS) method for the above equation

[21], [22]. Taking the derivative of the right hand side of Eq. (15) with respect to v_k, we have the following equation:

$$0 = H^T_{F(\hat{x}v)}\hat{x}_k + H_{F(vv)}v_k. \tag{21}$$

Thus, we have the following control policy:

$$v_k = -H^{-1}_{F(vv)}H^T_{F(\hat{x}v)}\hat{x}_k. \tag{22}$$

As a result, v_k is obtained without using the state and input at time $k+1$. Repeating the H_F evaluation and policy improvement based on Eq. (22), $\hat{P}$ and v_k converge to $\bar{P}_s$ and v^*_k, respectively. Algorithm 1 shows the Q-learning for the control policy. The noise e is added to the control policy so as to excite exploration. The initial control gain F_0 is selected so as to be a stabilizing control gain. The lines from 3 to 7 evaluate H_F, and the policy is improved at line 8. Positive numbers N and M denote the number of iterations of each loop. Shown in Fig. 1 is a block diagram of the state feedback control system where the optimal gain is adapted using Q-learning. The learner performs Algorithm 1 except for the lines 4 and 5, which are done in the controller. If we design a controller proposed by [6], which depends on only the state, under the existence of the computation time delay, then the optimal feedback gain may not be obtained in general since it depends on both the state and the input as shown in Eq. (10).

Algorithm 1 Q-learning for the state feedback control with a delay in the feedback loop using RLS

1: The initial feedback gain F_0 is selected so as to stabilize the system.
2: **for** $i = 0$ to N **do**
3: **for** $k = 1$ to M **do**
4: $v_k = -F_i\hat{x}_k + e$. // e is a noise.
5: Input $u_k = v_{k-1}$ to the system, resulting in state x_{k+1}.
6: Update the estimations of the Q function parameters $\hat{H}_{F_i}$, using RLS.
7: **end for**
8: $F_{i+1} = \hat{H}^{-1}_{F_i(vv)}\hat{H}^T_{F_i(\hat{x}v)}$.
9: $\hat{H}_{F_{i+1}} = \hat{H}_{F_i}$.
10: **end for**

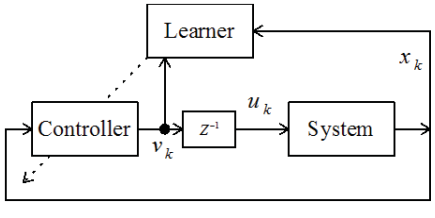

Fig. 1. The block diagram of the Q-learning-based adaptive state feedback control system taking the computation time delay into consideration, where z^{-1} corresponds to the computation time delay.

Simulation

We consider the following unstable linear system:

$$x_{k+1} = \begin{bmatrix} 1 & 2 \\ -3 & -4 \end{bmatrix} x_k + \begin{bmatrix} 1 \\ 0 \end{bmatrix} u_k, \tag{23}$$

$$y_k = \begin{bmatrix} 4 & 6 \end{bmatrix}. \tag{24}$$

Let $Q_s = \begin{bmatrix} 3 & 2 \\ 2 & 5 \end{bmatrix}$ and $R_s = 1$. The theoretical value of the optimal H_{F^*} and the optimal control gain F^* are as follows:

$$H_{F^*} = \begin{bmatrix} 2805.209 & 2958.5572 & -1169.5823 & 366.76898 \\ 2958.5572 & 3129.1554 & -1226.8812 & 382.04318 \\ -1169.5823 & -1226.8812 & 498.84291 & -160.47319 \\ 366.76898 & 382.04318 & -160.47319 & 58.325296 \end{bmatrix}, \tag{25}$$

$$F^* = \begin{bmatrix} 6.2883347 & 6.5502142 & -2.7513481 \end{bmatrix}. \tag{26}$$

Let $N = 10$ and $M = 400$. Shown in Figs. 2 and 3 are the convergence processes of $||\hat{H}_F - H_{F^*}||_2$ and $||\hat{F} - F^*||_2$, respectively.

In the simulation, when we did not set M to be sufficiently large, the feedback gain updated at the policy improvement loop sometimes made the controlled system unstable, so that we could not obtain the optimal gain. However, the larger M is, the longer the initial H_F evaluation loop is. So, we have to take into consideration such a trade-off when we determine M.

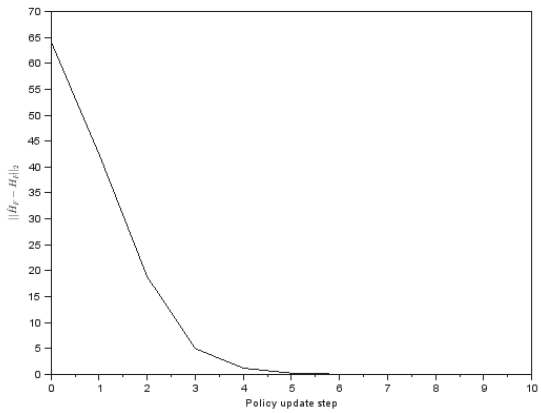

Fig. 2. Convergence process of $||\hat{H}_F - H_{F^*}||_2$ for state feedback control.

III. Q-LEARNING-BASED OPTIMAL CONTROL USING OUTPUT DATA

In this section, we consider the Q-learning-based optimal control using the output. In general, we use an observer to estimate the state. However, we cannot use it here, since A, B, and C are unknown. Therefore, we estimate the state by

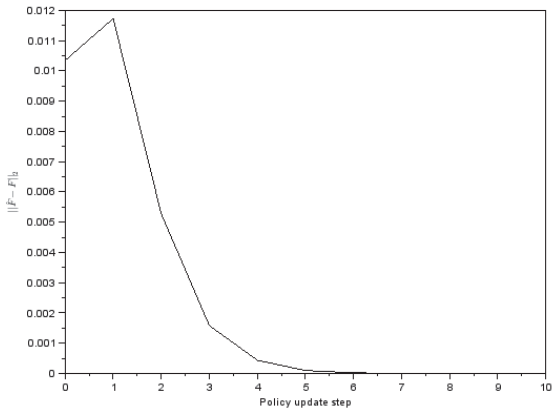

Fig. 3. Convergence process of $||\hat{F} - F^*||_2$ for state feedback control.

the data-based approach. Namely, we use the last L outputs and inputs, by which the current state is estimated if the system is observable, where L is an integer equal to or larger than the observability index of the system [14], [16], [17]. We consider the following cost function:

$$V_F(\hat{x}_k) = \sum_{i=k}^{\infty} \left(y_i^T Q_o y_i + v_i^T R_o v_i \right), \tag{27}$$

where $Q_o = Q_o^T \geq 0$ and $R_o = R_o^T > 0$. Note that the cost function has the quadratic term y_k instead of x_k.
The Bellman equation is given by

$$V_F(\hat{x}_k) = y_k^T Q_o y_k + v_k^T R_o v_k + V_F(\hat{x}_{k+1}). \tag{28}$$

The observability condition means that observations of the output y_k over a long enough time horizon can be used to reconstruct the full state x_k. Given the current time k, we consider a finite time horizon $[k-L,\ k]$. M_y is defined as follows:

$$M_y = A^L V_L^+, \tag{29}$$

where V_L^+ is the left inverse of the observability matrix V_L. M_u is defined as $U_L - M_0 T_L$, where U_L is the controllability matrix and T_L is the Toeplits matrix of Markov parameters [17]. Then x_k is obtained as

$$\begin{aligned} x_k &= \begin{bmatrix} M_u & M_y \end{bmatrix} \begin{bmatrix} \overline{u}_{k-1,k-L} \\ \overline{y}_{k-1,k-L} \end{bmatrix} \\ &= \begin{bmatrix} M_u & M_y \end{bmatrix} \overline{z}_{k-1,k-L}, \end{aligned} \tag{30}$$

where

$$\overline{u}_{k-1,k-L} = \begin{bmatrix} u_{k-1} \\ u_{k-2} \\ \vdots \\ u_{k-L} \end{bmatrix} \in R^{mL}, \tag{31}$$

$$\overline{y}_{k-1,k-L} = \begin{bmatrix} y_{k-1} \\ y_{k-2} \\ \vdots \\ y_{k-L} \end{bmatrix} \in R^{pL}. \tag{32}$$

We define $\hat{z}_k$ and $\hat{M}$ as follows:

$$\hat{z}_k = \begin{bmatrix} \overline{z}_{k-1,k-L} \\ u_k \end{bmatrix}, \tag{33}$$

$$\hat{M} = \begin{bmatrix} \begin{bmatrix} M_u & M_y \end{bmatrix} & 0 \\ 0 & I_m \end{bmatrix}. \tag{34}$$

Then, the current state $\hat{x}_k$ is obtained from the observed last L outputs and the last L inputs as follows:

$$\hat{x}_k = \hat{M} \hat{z}_k. \tag{35}$$

The optimal control policy is

$$v_k^* = -(R_o + B^T P_o B)^{-1} B^T P_o A \begin{bmatrix} A & B \end{bmatrix} \hat{M} \hat{z}_k, \tag{36}$$

where P_o is a solution of the following equation:

$$\begin{aligned} &C^T Q_o C + A^T P_o A \\ &\quad - A^T P_o B (R_o + B^T P_o B)^{-1} B^T P_o A = P_o. \end{aligned} \tag{37}$$

The optimal cost function is described by the quadratic equation of the state.

$$\begin{aligned} V^*(\hat{x}_k) &= \hat{x}_k^T \bar{P}_o \hat{x}_k \\ &= \hat{z}_k^T \hat{M}^T \bar{P}_o \hat{M} \hat{z}_k, \end{aligned} \tag{38}$$

where $\bar{P}_o$ is defined by

$$\bar{P}_o = \begin{bmatrix} C^T Q_o C + A^T P_o A & A^T P_o B \\ B^T P_o A & B^T P_o B \end{bmatrix}. \tag{39}$$

Thus, A, B, and C are used to compute the optimal input. But, since we assume that they are unknown, we propose a Q-learning-based control policy without using the system's parameters A, B, and C. Let $Q_F(\hat{x}_k, v_k)$ be the right hand side of Eq. (28). Then, we have

$$\begin{aligned} Q_F(\hat{x}_k, v_k) &= y_k^T Q_o y_k + v_k^T R_o v_k + V_F(\hat{x}_{k+1}) \\ &= \hat{z}_k^T \hat{M}^T \hat{C}^T Q_o \hat{C} \hat{M} \hat{z}_k + v_k^T R_o v_k \\ &\quad + (\hat{A}\hat{x}_k + \hat{B} v_k)^T \hat{P} (\hat{A}\hat{x}_k + \hat{B} v_k) \\ &= \hat{z}_k^T (\hat{M}^T \hat{C}^T Q_o \hat{C} \hat{M} + \hat{M}^T \hat{A}^T \hat{P} \hat{A} \hat{M}) \hat{z}_k \\ &\quad + v_k^T \hat{B}^T \hat{P} \hat{A} \hat{M} \hat{z}_k + \hat{z}_k^T \hat{M}^T \hat{A}^T \hat{P} \hat{B} v_k \\ &\quad + v_k^T (R_o + \hat{B}^T \hat{P} \hat{B}) v_k. \end{aligned} \tag{40}$$

Thus, we have

$$\begin{aligned} Q_F(\hat{x}_k, v_k) &= \begin{bmatrix} \hat{z}_k \\ v_k \end{bmatrix}^T \begin{bmatrix} H_{F(\hat{z}\hat{z})} & H_{F(\hat{z}v)} \\ H_{F(v\hat{z})} & H_{F(vv)} \end{bmatrix} \begin{bmatrix} \hat{z}_k \\ v_k \end{bmatrix} \\ &= \begin{bmatrix} \hat{z}_k \\ v_k \end{bmatrix}^T H_F \begin{bmatrix} \hat{z}_k \\ v_k \end{bmatrix}, \end{aligned} \tag{41}$$

where

$$H_{F(\hat{z}\hat{z})} = \hat{M}^T \hat{C}^T Q_o \hat{C} \hat{M} + \hat{M}^T \hat{A}^T \hat{P} \hat{A} \hat{M}, \tag{42}$$

$$H_{F(\hat{z}v)} = \hat{M}^T \hat{A}^T \hat{P} \hat{B}, \tag{43}$$

$$H_{F(v\hat{z})} = H_{F(\hat{z}v)}^T, \tag{44}$$

$$H_{F(vv)} = R_o + \hat{B}^T \hat{P} \hat{B}. \tag{45}$$

We take the derivative of the right hand side of Eq. (41) with respect to v_k, where H_F is estimated by using RLS. Then, we obtain

$$0 = H_{F(\hat{z}v)}^T \hat{z}_k + H_{F(vv)} v_k. \tag{46}$$

Thus, we have

$$\begin{aligned} v_k &= -H_{F(vv)}^{-1} H_{F(\hat{z}v)}^T \hat{z}_k \\ &= -H_{F(vv)}^{-1} H_{F(\hat{z}v)}^T \begin{bmatrix} \overline{u}_{k-1,k-L} \\ \overline{y}_{k-1,k-L} \\ u_k \end{bmatrix}. \end{aligned} \tag{47}$$

From Eq. (47), we can compute the input v_k using the last L inputs and outputs. In the same way as in the case of a state feedback controller, $\hat{P}$ and v_k converge to $\bar{P}_o$ and v_k^*, respectively. An algorithm of Q-learning-based optimal output feedback control with a delay is obtained by replacing line 4 of Algorithm 1 with $v_k = -F_i \hat{z}_k + e$, and line 8 with $F_{i+1} = \hat{H}_{F_i(vv)}^{-1} \hat{H}_{F_i(\hat{z}v)}^T$.

Simulation

We reconsider the unstable linear system (23), (24). Let $Q_o = 1$ and $R_o = 1$. The observability index K of the system is 2, and L is set to be 2. The theoretical optimal value of H_{F^*} and the optimal control policy F^* are as follows:

$$H_{F^*} = \begin{bmatrix} 1979.1997 & -817.15935 & -880.85801 \\ -817.15935 & 337.57485 & 364.35264 \\ -880.85801 & 364.35264 & 394.37623 \\ -817.15935 & 337.57485 & 364.35264 \\ -802.50388 & 330.24632 & 353.35824 \\ 236.15596 & -96.789802 & -102.60835 \end{bmatrix}$$
$$\begin{bmatrix} -817.15935 & -802.50388 & 236.15596 \\ 337.57485 & 330.24632 & -96.789802 \\ 364.35264 & 353.35824 & -102.60835 \\ 337.57485 & 330.24632 & -96.789802 \\ 330.24632 & 331.91618 & -100.62233 \\ -96.789802 & -100.62233 & 33.85551 \end{bmatrix}, \tag{48}$$

$$F^* = \begin{bmatrix} 6.9754054 & -2.8589078 & -3.030772 \\ & -2.8589078 & -2.9721103 \end{bmatrix}. \tag{49}$$

Let $N = 10$ and $M = 400$. Shown in Figs. 4 and 5 are the convergence processes of $||\hat{H}_F - H_{F^*}||_2$ and $||\hat{F} - F^*||_2$, respectively. From these figures, it is shown that both $\hat{H}_F$ and $\hat{F}$ converge to their optimal values. Under the output observation, the proposed learning controller can achieve the optimal control action, and its convergence speed to the optimal gain is the same as that of the state feedback controllers proposed in the previous section.

IV. Q-LEARNING-BASED CONTROL OF A QUADROTOR

A quadrotor has four rotors in cross configuration which generates its lift. By varying the rotor speed, we can change the lift force and create motion. Shown in Fig. 6 is the body coordinate of the quadrotor. RL has been applied to the quadrotor for accommodating nonlinear disturbances [23], and to a helicopter, in order to design controllers

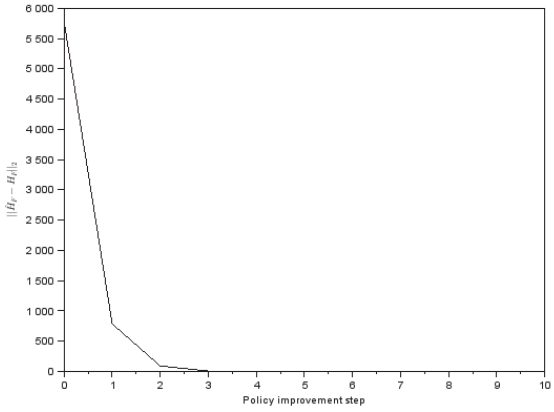

Fig. 4. Convergence process of $||\hat{H}_F - H_{F^*}||_2$ for output feedback control.

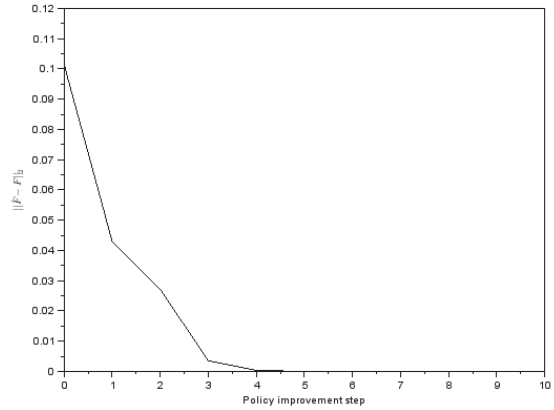

Fig. 5. Convergence process of $||\hat{F} - F^*||_2$ for output feedback control.

for low speed aerobatic maneuvers [24]. In this paper, we apply the proposed state feedback controller to quadrotor hovering. Shown in Table I is a list of the quadrotor's system parameters. x, y, and z axes are the body coordinates of the quadrotor. The continuous-time state model of the hovering quadrotor is given as follows [25]:

$$\begin{bmatrix} \dot{\Theta}_1 \\ \dot{\Theta}_2 \end{bmatrix} = \begin{bmatrix} 0_{3\times3} & I_{3\times3} \\ 0_{3\times3} & 0_{3\times3} \end{bmatrix} \begin{bmatrix} \Theta_1 \\ \Theta_2 \end{bmatrix} + \begin{bmatrix} 0_{3\times3} \\ T^{-1} \end{bmatrix} U. \tag{50}$$

Discretizing the above equation by the sampling period h, we obtain the following discrete-time model:

$$x_{k+1} = \begin{bmatrix} I_{3\times3} & hI_{3\times3} \\ 0_{3\times3} & I_{3\times3} \end{bmatrix} x_k + \begin{bmatrix} \frac{h^2}{2} T^{-1} \\ hT^{-1} \end{bmatrix} U_k, \tag{51}$$

where x_k and U_k are the discretized state and the input at the time kh, respectively.

Here, let $T_x = 0.0717$, $T_y = 0.0717$, $T_z = 0.135$, and

$h = 0.5$. The theoretical optimal gain F^* is as follows:

$$F^* = \begin{bmatrix} 0.11113 & 0 & 0 & 0.22316 & 0 \\ 0 & 0.11113 & 0 & 0 & 0.22316 \\ 0 & 0 & 0.19473 & 0 & 0 \\ 0 & 1.36248 & 0 & 0 & \\ 0 & 0 & 1.36248 & 0 & \\ 0.39473 & 0 & 0 & 1.28166 & \end{bmatrix}. \tag{52}$$

Let $N = 10$ and $M = 1800$. Shown in Fig. 7 are the time responses of each angle before learning. Shown in Fig. 8 are the responses after learning. By comparing these figures, it is clear that the responses after learning converge to the origin faster than before learning, and do not exhibit undershoot. Shown in Figs. 9 and 10 are the convergence processes of $||\hat{H}_F - H_{F^*}||_2$ and $||\hat{F} - F^*||_2$, respectively. It is noted that the larger the number of optimized parameters in the feedback gain matrix is, the more data we need to improve the policy. So, we set $M = 1800$ in this example while $M = 400$ in the examples in previous sections. On the other hand, in general, we set M to be larger, policy improvement can be more efficiently achieved. Figures 9 and 10 show that the convergence speeds of both H_F and F are fast.

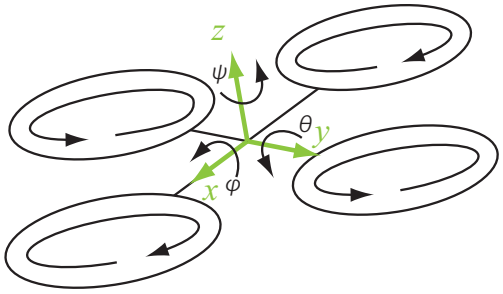

Fig. 6. The body coordinates of the quadrotor.

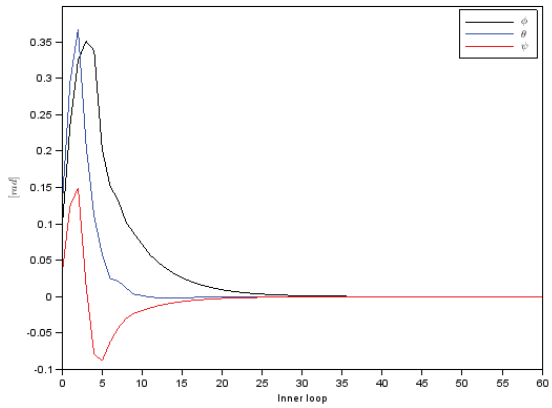

Fig. 7. Responses of ϕ, θ, and ψ before learning.

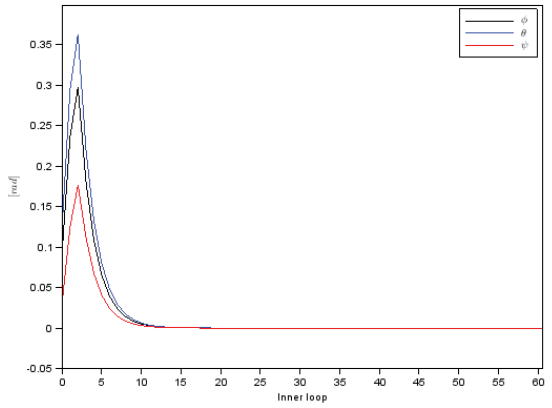

Fig. 8. Responses of ϕ, θ, and ψ after learning.

V. Conclusion

In this paper, we have proposed a Q-learning-based optimal state and output feedback control with a delay in the feedback loop. For an unknown system, an optimal state feedback gain was provided by sequences of pairs of a state and an input. As a state can be reconstructed by the last L input and output data, the optimal output feedback control was realized. The simulation for a quadrotor showed the applicability of the proposed method for multiple input systems.

Our future work is to extend the proposed methods to the design of digital controllers with more than one unit time delay in the computation of the control input.

TABLE I
QUADROTOR'S PARAMETERS.

variables and coefficients	explanation
ϕ	roll angle [rad]
θ	pitch angle [rad]
ψ	yaw angle [rad]
Θ_1	$[\phi\ \theta\ \psi]^T$
Θ_2	$\dot{\Theta}_1$
T_x	moment of inertia about x axis [kgm^2]
T_y	moment of inertia about y axis [kgm^2]
T_z	moment of inertia about z axis [kgm^2]
T	$diag(T_x, T_y, T_z)$
U_ϕ	torque of roll angle $[N \cdot m]$
U_θ	torque of pitch angle $[N \cdot m]$
U_ψ	torque of yaw angle $[N \cdot m]$
U	$[U_\phi\ U_\theta\ U_\psi]^T$

Acknowledgment

This work was supported in part by a Grant-in-Aid for Scientific Research (B) No. 24360164 from the MEXT of Japan.

References

[1] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*, Cambridge University Press, 1998.

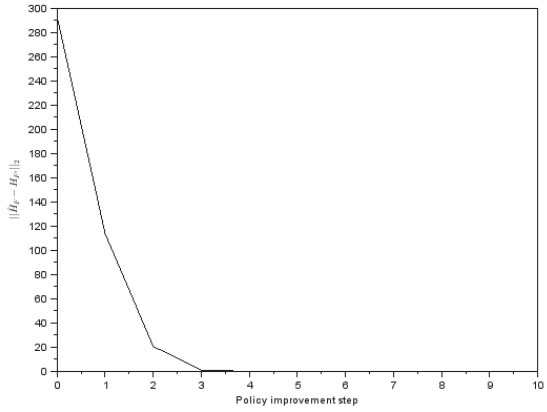

Fig. 9. Convergence process of $||\hat{H}_F - H_{F^*}||_2$ for the control of the quadrotor.

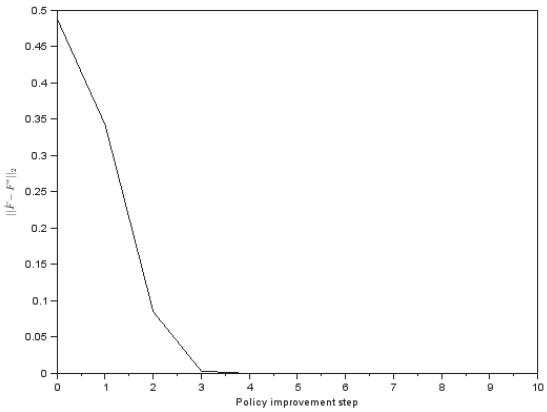

Fig. 10. Convergence process of $||\hat{F} - F^*||_2$ for the control of the quadrotor.

[2] C. Watkins, "learning from delayed rewords." Ph.D. dissertation, University of Cambridge, 1989.
[3] C. Watkins and P. Dayan, "Q-learning," *Machine learning*, vol. 8, no. 3-4, pp. 279–292, 1992.
[4] D. P. Bertsekas and J. N. Tsitsiklis, *Neuro-Dynamic Programming*, Athena Scientific, 1996.
[5] M. Wiering and M. van Otterlo(Eds.), *Reinforcement Learning*, Springer, 2012.
[6] F. L. Lewis and D. Vrabie, "Reinforcement Learning and adaptive dynamic programming for feedback control," *IEEE Circuits and Systems Magazine*, vol. 9, no. 3, pp. 32–50, 2009.
[7] D. Vrabie, K. G. Vamvoudakis, and F. L. Lewis, *Optimal Adaptive Control and Differential Games by Reinforcement Learning Principles*, The Institute of Engineering and Technology, London, UK, 2013.
[8] F. L. Lewis and D. Liu, *Reinforcement Learning and Approximate Dynamic Programming for Feedback Control*, IEEE Press, 2013.
[9] H. Zhang D. Liu, Y. Luo, and D. Wang, *Adaptive Dynamic Programming for Control*, Springer, 2013.
[10] K. J. Åström and B. Wittenmark, *Adaptive Control*, 2nd Edition, Addison-Wesley Pub., 1995.
[11] K. Doya, "Reinforcement learning in continuous time and space," *Neural Computation*, vol. 12, no. 1, pp. 219–245, 2000.
[12] K. Vamvoudakis and F. Lewis, "Online actor-critic algorithm to solve the continuous-time infinite horizon optimal control problem," *Automatica*, vol. 46, no. 5, pp. 878–888, 2010.
[13] H. Modares, F.Ł. Lewis, and M. Naghibi-Sistani, "Adaptive optimal control of unknown constrained-input systems using policy iteration and neural networks," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 24, no. 10, pp. 1513–1525, 2013.
[14] R. K. Lim, M. Q. Phan, and R. W. Longman, "State estimation with ARMarkov models," *Department of Mechanical and Aerospace Engineering Technical Report* no. 3046, Princeton University, Princeton, NJ, 1998.
[15] G. Shi and R. E. Skelton, "Markov data-based LQG control," *ASME Journal of Dynamic Systems, Measurement, and Control*, vol. 122, no. 3, pp. 551–559, 2000.
[16] W. Aangenent, D. Kostić, B. De Jager, R. van de Molengraft, and M. Steinbuch, "Data-based optimal control", in *Proceedings of 2005 American Control Conference*, pp. 1460–1465, 2005.
[17] F. L. Lewis and K. G. Vamvoudakis, "Reinforcement learning for partially observable dynamic processes: adaptive dynamic programming using measured output data," *IEEE Transactions on Systems, Man, and Cybernetics. Part B-Cybernetics*, vol. 41, no. 1, pp. 14–25, 2011.
[18] A. Bemporad, M. Heemels, and M. Johansson, *Networked Control Systems*, Springer, 2010.
[19] K. G. Shin and X. Cui, "Computing time delay and its effects on real-time control systems," *IEEE Transactions on Control Systems Technology*, vol. 3, no. 2, pp. 218–224, 1995.
[20] T. Mita, "Optimal digital feedback control systems counting computation time of control laws," *IEEE Transactions on Automatic Control*, vol. AC–30, no. 6, pp. 542–548, 1985.
[21] S. Bradtke and A. Barto, "Linear least-squares algorithms for temporal difference learning," *Machine Learning*, vol. 22, no. 1-3, pp. 33–57, 1996.
[22] G. C. Goodwin and K. S. Sin, *Adaptive Filtering Prediction and Control*. Prentice-Hall, 1984.
[23] A. Y. Ng, A. Coates, M. Diel, V. Ganapathi, J. Schulte, B. Tse, E. Berger, and E. Liang, "Autonomous inverted helicopter flight via reinforcement learning," in *Experimental Robotics IX*, pp. 363–372, 2006
[24] S. L. Waslander, G. M. Hoffmann, J. S. Jang, and C. J. Tomlin, "Multi-agent quadrotor testbed control design: Integral sliding mode vs. reinforcement learning," in *International Conference on Intelligence Robots and Systems*, pp. 3712–3717, 2005.
[25] S. Bouabdallah and R. Siegwart, "Full control of a quadrotor," in *IEEE/RSJ International Conference on Intelligent Robots and Systems, 2007*, pp. 153–158, 2007.