MANCHESTER 1824
The University of Manchester

The University of Manchester Research

Networked controller and observer design of discrete-time systems with inaccurate model parameters

DOI:
10.1016/j.isatra.2019.08.029

Document Version
Accepted author manuscript

Link to publication record in Manchester Research Explorer

Citation for published version (APA):
Li, J., Xiao, Z., Li, P., & Ding, Z. (2019). Networked controller and observer design of discrete-time systems with inaccurate model parameters. *ISA Transactions*. https://doi.org/10.1016/j.isatra.2019.08.029

Published in:
ISA Transactions

Citing this paper
Please note that where the full-text provided on Manchester Research Explorer is the Author Accepted Manuscript or Proof version this may differ from the final Published version. If citing, it is advised that you check and use the publisher's definitive version.

General rights
Copyright and moral rights for the publications made accessible in the Research Explorer are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

Takedown policy
If you believe that this document breaches copyright please refer to the University of Manchester's Takedown Procedures [http://man.ac.uk/04Y6Bo] or contact uml.scholarlycommunications@manchester.ac.uk providing relevant details, so we can investigate your claim.

OPEN ACCESS

Download date:09. Jun. 2022

To appear in ISA Transactions

Networked controller and observer design of discrete-time systems with inaccurate model parameters

Jinna Li[1,2*], Zhenfei Xiao[1], Ping Li[1], and Zhengtao Ding[3]

[1] School of Information and Control Engineering, Liaoning Shihua University, Liaoning 113001, P.R. China;
[2]State Key Laboratory of Synthetical Automation for Process Industries, Northeastern University, Shenyang 110819, P.R. China;
[3]School of Electrical & Electronic Engineering, the University of Manchester, Manchester M13 9PL, UK

Abstract
This paper develops a novel off-policy Q-learning method to find the optimal observer gain and the optimal controller for achieving optimality of network-communication based linear discrete-time systems using only measured data. The primary advantage of this off-policy Q-learning method is that it can work for the linear discrete-time systems with inaccurate system model, unmeasurable system states and network-induced delays. To this end, an optimization problem for networked control systems composed of a plant, a state observer and a Smith predictor is formulated first. The Smith predictor is employed to not only compensate network-induced delays, but also make the separation principle hold, thus the observer and controller can be designed separately. Then, the off-policy Q-learning is implemented for learning the optimal observer gain and the optimal controller combined with the Smith predictor, such that a novel off-policy Q-learning algorithm is derived using only input, output and delayed estimated state of systems, not the inaccurate system matrices. The convergences of the iterative observer gain and the iterative controller gain are rigorously proven. Finally, simulation results are given to verify the effectiveness of the proposed method.
Key words: Data-driven optimal control, Networked control, Q-learning, State observation

1. Introduction

It is well known that networked control systems have attracted much attention by researchers and have been applied into wide variety of practical applications, such as industries, unmanned vehicles, medical treatment, etc., over the past couples of decades [1-4]. This is because the basic characteristics that information and control signals are transmitted via wire or wireless networks among sensors, controllers and actuators brings the advantages to control systems, such as low cost, reduced weight, easy installation and maintenance. While, the negative impact brought by network-induced delays and packet losses on performance of control systems inherently exist in networked control systems [1-6].

For alleviating the above-mentioned negative impact and optimizing the control performance of systems, rich achievements have been reported in the fields of control and communication, such as handling network-induced delay, packet losses, bandwidth, coding and decoding [1-9], etc. It can be noticed that all the above-mentioned approaches have something in common, that is they all require the system dynamics to be accurately known a prior.

Consider the following networked control system

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

and the networked controller

$$u_k = Kx_{k-d_k} \tag{2}$$

where $x_k = x(k) \in R^n$, $u_k = u(k) \in R^m$ and $y_k = y(k) \in R^p$ are state, control input and output, respectively. k $(k = 0,1,2\cdots)$ is the sampling time instant. A, B and C are matrices with

appropriate dimensions. $d_k = d(k)$ denote the bounded network-induced delays including the transmission delays from the sensor to the controller and the computing time occurred in the controller. Without loss of generality, d_k is a nonnegative integer with $0 \le d_k \le \delta_{\max}$, and $\delta_{\max}$ is some positive integer [1, 2, 6, 7].

For the case that the system matrices A, B and C cannot be modelled accurately even they are completely unknown, the reinforcement learning (RL) has been widely used to learn the optimal controller $u_k = Kx_k$ for stabilizing and optimizing the control performance of system (1) [10-14]. However, network-induced delays and unmeasurable state were not taken into account in these results [10-14]. Since the states x_k might be unmeasurable or the cost used for measuring them is very high in most of practical control system applications [15-18], such as unmanned vehicles and smart grid, etc., and network-induced delays inevitably occur when sending information from the sensor to the controller via a network in the above-mentioned practical industries [2, 3, 19, 20], then these two problems have to be considered even though it would pose challenge on designing controller for system (1) subject to inaccurate system model.

The most relevant results to our focus are [16, 21-23]. [16] utilized the past inputs and outputs to estimate the states and developed a RL algorithm to find the optimal tracking controller for system (1) with unknown dynamics. Notice that the algorithms presented in [16] cannot work when the network-induced delays occur during sending data from the sensors to the controllers. For systems with network-induced delays and unknown system matrices, [21-23] developed RL methods to find the optimal controller while assuming all states of systems to be measured, such that performance of systems was optimized.

To our best knowledge, simultaneously handling inaccurate system matrices, unmeasured state and network-induced delays for system (1) have rarely been reported up to now. To address these challenging problems, this paper will present a novel off-policy Q-learning algorithm to make networked control system (1) be optimum by introducing a Smith predictor, designing an observer state based optimal controller and doing some appropriate mathematical manipulation. The proposed algorithm is implemented using only measured data and independent of the inaccurate system matrices A, B and C.

The main contributions of this paper are highlighted below:

1. Usage of Smith predictor for observer-based networked control makes the Separation Principal tenable, such that it is feasible to separately solve the optimal controller and the optimal observer for networked control systems.
2. Compared with [16], there are two differences. One is that the states of observer other than the past inputs and outputs are used to estimate the states of systems in this paper, thus the computation complexity can be reduced to some extent. The other is that network-induced delay is compensated in this paper, while [16] neglected the existence of network-induced delay for control systems. Notice that [21-23] assumed that states of systems can be measurable. Since network-induced delay, unmeasurable states of systems and inaccurate model parameters are simultaneously taken into account in this paper, then the proposed optimal control method is more general and practical.
3. A novel off-policy Q-learning algorithm is, for the first time, developed in this paper for networked control systems subject to unmeasured states of systems, such that the optimal observer and the predicted observer state-based optimal controller can be designed without requiring accurate system matrices.
4. Rigorous proofs are presented to show convergence of algorithms and optimality of the proposed state observer and the predicted observer state based controller.

The paper is organized as follows. Section 2 formulates an optimal control problem for networked control systems with a state observer and a Smith predictor. Section 3 devotes to the optimal observer design using the off-policy Q-learning approach. Section 4 presents a novel off-policy Q-learning algorithm to find the optimal controller. The results of simulations are given in Section 5. Section 6 states the conclusions.

2. Problem Formulation

In this section, an optimal control problem of networked control system with the compensation of network-induced delays and the state observer is formulated.

As shown in Fig. 1, a state observer is put in system (1) to estimate the unmeasured state of the plant. Since there exist network-induced delays from the sensor of the plant to the controller, then the estimated states will be delayed when they arrive at the controller via the network. A Smith predictor similar to that in [21] is presented for compensating the network-induced delays and thus the control input is computed using the predicted observer state to enable the closed-loop plant stable and high performance.

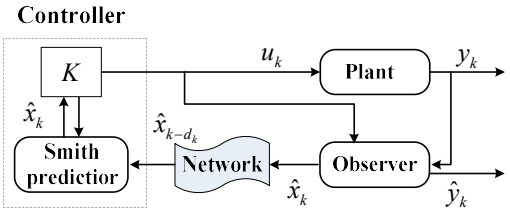

Figure 1: The Architecture of networked control system with observer and predictor

The dynamics of the added state observer is given below:

$$\begin{aligned} \hat{x}_{k+1} &= A\hat{x}_k + Bu_k + L\left(y_k - \hat{y}_k\right) \\ \hat{y}_k &= C\hat{x}_k \end{aligned} \tag{3}$$

where $\hat{x}_k$ and $\hat{y}_k$ are states and outputs of the observer, respectively. L is a gain matrix of the observer. Assume that the pair (A,B) is controllable and the pair (A,C) is observable. The controllability and observability of systems can generally be identified in practical applications even though system matrices A, B and C are inaccurate, such as some industrial control processes, unmanned aerial vehicles, etc. [2, 3, 17-20, 24]. This class of systems are what we focused on in this paper.

Let $e_k = x_k - \hat{x}_k$, combing (1) and (3) yields the following dynamics of observer error:

$$e_{k+1} = (A - LC)e_k \tag{4}$$

Remark 1. From (4), one can find that the error of observer is only dominated by the observer gain L. Compared with [16], the observer states instead of the past inputs and outputs are used to approximate the states of system (1).

From Fig. 1, one can see that the observer state $\hat{x}_k$ is unavailable at the time instant k due to the network-induced delays, while $\hat{x}_{k-d_k}$ is available. By (3), one has

$$\begin{aligned} \hat{x}_{k-d_k+1} &= A\hat{x}_{k-d_k} + Bu_{k-d_k} \\ &\quad + L(y_{k-d_k} - \hat{y}_{k-d_k}) \end{aligned}$$

$$\begin{aligned} \hat{x}_{k-d_k+2} &= A^2\hat{x}_{k-d_k} + Bu_{k-d_k+1} + ABu_{k-d_k} \\ &\quad + L(y_{k-d_k+1} - \hat{y}_{k-d_k+1}) \\ &\quad + AL(y_{k-d_k} - \hat{y}_{k-d_k}) \end{aligned}$$

…

$$\begin{aligned}\hat{x}_k &= A^{d_k}\hat{x}_{k-d_k} + \sum_{i=1}^{d_k} A^{i-1}Bu_{k-i} \\ &+ \sum_{i=1}^{d_k} A^{i-1}L(y_{k-i} - \hat{y}_{k-i})\end{aligned} \tag{5}$$

Thus $\hat{x}_k$ can be predicted according to the most recent observer states by using the novel Smith predictor (5).

A static feedback controller [25] based on the predicted observer state is chosen as

$$u_k = K\hat{x}_k \tag{6}$$

The target of this paper is to find the controller (6) to stabilize system (1) and minimize the following prescribed performance index without use of system matrices A, B and C.

$$J = \frac{1}{2}\sum_{k=0}^{\infty} y_k^T Q_p y_k + u_k^T R_p u_k \tag{7}$$

where Q_p and R_p are respectively positive semi-definite matrix and positive definite matrix. Thus, we express the concerned optimization problem in this paper as:

Problem 1:

$$\min_{u_k} \frac{1}{2}\sum_{k=0}^{\infty} y_k^T Q_p y_k + u_k^T R_p u_k \tag{8}$$

s.t. (1), (4), (5) and (6).

Since the observer state $\hat{x}_k$ can be available at the time instant k due to the usage of Smith predictor (5), the closed-loop system composed by (1), (4) and (6) can be written as an augmented form

$$\xi_{k+1} = \begin{bmatrix} A+BK & -BK \\ 0 & A-LC \end{bmatrix}\xi_k \tag{9}$$

where $\xi_k = \begin{bmatrix} x_k \\ e_k \end{bmatrix}$.

Obviously, the eigenvalues of the closed-loop (9) are those of $A+BK$ and those of $A-LC$. Thus the stability of the observer (3) and the stability and cost (8) of the system (1) are independent, which is the Separation Principal. Thus, when solving Problem 1, L and K can be separately designed by decomposing Problem 1 into two subproblems below:

Subproblem 1:

$$\begin{aligned}&\min_{u_k} \frac{1}{2}\sum_{k=0}^{\infty} y_k^T Q_p y_k + u_k^T R_p u_k \\ &\text{s.t.} \\ &\quad x_{k+1} = Ax_k + Bu_k \\ &\quad y_k = Cx_k \\ &\quad u_k = Kx_k\end{aligned} \tag{10}$$

and

Subproblem 2: find the observer gain L for (4) to make $\lim_{k\to\infty} e_k = 0$.

Remark 2. All of the pairs (K, L) that can ensure $\lim_{k\to\infty} x_k = 0$ and $\lim_{k\to\infty} e_k = 0$ compose the feasible solutions to the focused optimization problem (8). Since $y_k = Cx_k$, then the performance index in (10) can be transformed as $\min_{u_k} \frac{1}{2}\sum_{k=0}^{\infty} x_k^T \tilde{Q}_p x_k + u_k^T R_p u_k$, where $\tilde{Q}_p = C^T Q_p C$. In this sense, Subproblem 1

becomes a standard linear quadratic regulation problem, then there exists unique one solution u_k to Problem 1 since it is decomposed into separately finding optimal controller u_k and observer.

Note that Subproblem 1 and Subproblem 2 can be easily solved if system parameters A, B and C are accurately known. But if these system parameters cannot be accurately identified, then the traditional model-based observer and controller design method as well as the Smith predictor proposed in this paper cannot work. What we are interested in here is to find a data-driven method to solve the above two subproblems and implement the proposed Smith predictor using only measured data.

Before solving Subproblem 1, designing the observer gain L is going to be the first thing to do.

3. Optimal Observer Design

This section is devoted to finding the optimal observer gain L which minimizes the accumulated observer error. The idea of off-policy Q-Learning is used here to present a learning algorithm with no deed of accurate system matrices A, B and C, so that the optimal observer gain L can be found using only measured data.

Define an observer policy

$$w_k = L\left(y_k - \hat{y}_k\right) \tag{11}$$

then the dynamics of observer error (4) can be rewritten as

$$e_{k+1} = Ae_k - w_k \tag{12}$$

To obtain the optimal observer, Subproblem 2 is changed into the optimization problem below:

Problem 2:

$$\min_{w_k} \frac{1}{2}\sum_{k=0}^{\infty}\left(y_k - \hat{y}_k\right)^T Q_{\mathrm{o}}\left(y_k - \hat{y}_k\right) + w_k^T R_{\mathrm{o}} w_k \tag{13}$$

s.t. (11) and (12).

where Q_{o} and R_{o} are respectively positive semi-definite matrix and positive definite matrix. If we try to find the optimal observer gain L for Problem 2, then unique one solution can be found since it is a standard linear quadratic regulation problem if $y_k - \hat{y}_k$ is replaced by Ce_k.

The model-based Q-learning algorithm is first given, then an off-policy Q-learning algorithm is derived to find the optimal observer gain using only measured data.

3.1. Model-based optimal observer design

Let $w_k = L\left(y_k - \hat{y}_k\right)$ be an admissible policy, a value function and an action-dependent Q-function can be respectively defined as [2, 26, 27]

$$V_o(e_k) = \frac{1}{2}\sum_{i=k}^{\infty} e_i^T \tilde{Q}_o e_i + w_i^T R_o w_i \tag{14}$$

and

$$Q_{ob}(e_k, w_k) = \frac{1}{2}(e_k^{\mathrm{T}} \tilde{Q}_o e_k + w_k^{\mathrm{T}} R_o w_k) + V_o(e_{k+1}) \tag{15}$$

where $\tilde{Q}_o = C^T Q_o C$. Thus, the following relationship holds

$$\begin{aligned} V^*(e_k) &= \min_{w_k} \frac{1}{2}\sum_{i=k}^{\infty} e_i^T \tilde{Q}_o e_i + w_i^T R_o w_i \\ &= \min_{w_k} Q_{ob}(e_k, w_k) \\ &= Q_{ob}(e_k, w_k^*) \end{aligned} \tag{16}$$

According to the dynamic programming theory, the Q-function based algebraic Riccati equation (ARE) can be derived below

$$\begin{aligned} Q_{ob}(e_k, w_k^*) &= \min_{w_k}\left\{\frac{1}{2}(e_k^T\tilde{Q}_o e_k + w_k^T R_o w_k) + V_o^*(e_{k+1})\right\} \\ &= \frac{1}{2}(e_k^T\tilde{Q}_o e_k + (w_k^*)^T R_o w_k^*) + Q_{ob}(e_{k+1}, w_{k+1}^*) \end{aligned} \tag{17}$$

From (16), it follows that a unique optimal observer policy should be with the form of

$$w_k^* = \arg\min_{w_k} Q_{ob}(e_k, w_k) \tag{18}$$

The following lemma is useful to find the optimal observer policy.

Lemma 1[27, 28]: For Problem 2, under the admissible policy (11), the value function and the Q-function have the quadratic forms of

$$V_o(e_k) = \frac{1}{2}e_k^T P_o e_k \tag{19}$$

and

$$Q_{ob}(e_k, w_k) = \frac{1}{2}\begin{bmatrix} e_k \\ w_k \end{bmatrix}^T H_1 \begin{bmatrix} e_k \\ w_k \end{bmatrix} \tag{20}$$

where

$$\begin{aligned} H_1 &= \begin{bmatrix} H_{1,ee} & H_{1,ew} \\ * & H_{1,ww} \end{bmatrix} \\ &= \begin{bmatrix} \tilde{Q}_o + A^T P_0 A & -A^T P_0 \\ * & R_o + P_0 \end{bmatrix} \end{aligned} \tag{21}$$

$$P_o = \begin{bmatrix} I \\ LC \end{bmatrix}^T H_1 \begin{bmatrix} I \\ LC \end{bmatrix} \tag{22}$$

where the matrix I denotes an identity matrix with approximate dimensions. Substituting (20) into ARE (17), one has

$$\begin{aligned} \begin{bmatrix} e_k \\ w_k^* \end{bmatrix}^T H_1 \begin{bmatrix} e_k \\ w_k^* \end{bmatrix} &= e_k^T\tilde{Q}_o e_k + (w_k^*)^T R_o w_k^* + \begin{bmatrix} e_{k+1} \\ w_{k+1}^* \end{bmatrix}^T H_1 \begin{bmatrix} e_{k+1} \\ w_{k+1}^* \end{bmatrix} \\ &= e_k^T\tilde{Q}_o e_k + (w_k^*)^T R_o w_k^* \\ &+ \begin{bmatrix} e_k \\ w_k^* \end{bmatrix}^T \left(\begin{bmatrix} I \\ L^*C \end{bmatrix}\begin{bmatrix} A & -I \end{bmatrix}\right)^T H_1 \left(\begin{bmatrix} I \\ L^*C \end{bmatrix}\begin{bmatrix} A & -I \end{bmatrix}\right)\begin{bmatrix} e_k \\ w_k^* \end{bmatrix} \end{aligned} \tag{23}$$

Based on the necessary condition of optimality [12, 14, 24, 29], implementing $\frac{\partial Q_{ob}(e_k, w_k)}{\partial w_k} = 0$ yields

$$w_k^* = -H_{1,ww}^{-1}(H_{1,ew})^T e_k \tag{24}$$

Compared with (11) and (24), one has

$$L^*C = -H_{1,ww}^{-1}(H_{1,ew})^T \tag{25}$$

It means that the optimal observer policy w_k^* can be obtained if the Q-function matrix H_1 is solved from (23). To learn H_1, Algorithm 1 is provided.

Algorithm 1: Model-based policy iteration algorithm

1. Initialization: Given an admissible observer gain L^0, and let $j=0$, where j denotes the iteration index;

2. Policy evaluation: Calculate H_1^{j+1} with

$$\begin{aligned} H_1^{j+1} &= \begin{bmatrix} \tilde{Q}_o & 0 \\ 0 & R_o \end{bmatrix} \\ &+ \left(\begin{bmatrix} I \\ L^j C \end{bmatrix} \begin{bmatrix} A & -I \end{bmatrix} \right)^T H_1^{j+1} \left(\begin{bmatrix} I \\ L^j C \end{bmatrix} \begin{bmatrix} A & -I \end{bmatrix} \right) \end{aligned} \tag{26}$$

Iterative ARE equation (26) can be derived from (23) by deleting $[e_k^T \ (w_k^*)^T]$ and its transpose from the both sides of (23).

3. Policy update:

$$w_k^{j+1} = -(H_{1,ww}^{j+1})^{-1} (H_{1,we}^{j+1})^T e_k \tag{27}$$

or

$$L^{j+1} = -(H_{1,ww}^{j+1})^{-1} (H_{1,we}^{j+1})^T C^T (CC^T)^{-1} \tag{28}$$

where CC^T is invertible.

4. Stop when $\left\| H_1^j - H_1^{j+1} \right\| \le \varepsilon$ with a small constant ε $(\varepsilon > 0)$; Otherwise, let $j = j+1$ and go back to Step 2.

Remark 3. [27, 28] have proven that $\lim_{j\to\infty} H_1^{j+1} = H_1$ and $\lim_{j\to\infty} w_k^{j+1} = w_k^*$. One can find that calculating H_1^{j+1} requires the system matrices A and C to be accurately known when implementing Algorithm 1, while it is natural not to accurately model the system matrices A, B and C in real applications [13-20, 24].

Next subsection will focus on investigating an off-policy Q-learning algorithm to get optimal observer for system (1) with inaccurate system matrices A, B and C.

3.2. Data-driven off-policy Q-learning for the optimal observer

Here, we shall introduce two manipulations for reaching the goal of getting the optimal observer policy w_k^* using the data-driven off-policy learning approach. One manipulation is to define a virtual Q-function matrix $\bar{H}_1$ satisfied with.

$$H_1 = \begin{bmatrix} C & 0 \\ 0 & I \end{bmatrix}^T \bar{H}_1 \begin{bmatrix} C & 0 \\ 0 & I \end{bmatrix} \tag{29}$$

The other manipulation is to introduce an auxiliary variable $w_k^j = -(H_{1,ww}^j)^{-1} (H_{1,we}^j)^{\mathrm{T}} e_k$ into system (4), which yields

$$\begin{aligned} e_{k+1} &= Ae_k - w_k - w_k^j + w_k^j \\ &= \left(Ae_k - w_k^j\right) + \left(w_k^j - w_k\right) \end{aligned} \tag{30}$$

where w_k is called the behavior policy to generate data and w_k^j is viewed as the target policy needed to be learned.

By Lemma 1 and (29), one has

$$\begin{aligned} H_1 &= \begin{bmatrix} \tilde{Q}_o + A^T P_0 A & -A^T P_0 \\ * & R_o + P_0 \end{bmatrix} \\ &= \begin{bmatrix} C & 0 \\ 0 & I \end{bmatrix}^T \bar{H}_1 \begin{bmatrix} C & 0 \\ 0 & I \end{bmatrix} \\ &= \begin{bmatrix} C & 0 \\ 0 & I \end{bmatrix}^T \begin{bmatrix} \bar{H}_{1,11} & \bar{H}_{1,12} \\ * & \bar{H}_{1,22} \end{bmatrix} \begin{bmatrix} C & 0 \\ 0 & I \end{bmatrix} \\ &= \begin{bmatrix} C^T \bar{H}_{1,11} C & C^T \bar{H}_{1,12} \\ * & \bar{H}_{1,22} \end{bmatrix} \end{aligned} \tag{31}$$

Along the trajectory of (30) and refer to (26), one has

$$\begin{aligned} &\begin{bmatrix} e_k \\ w_k^j \end{bmatrix}^T H_1^{j+1} \begin{bmatrix} e_k \\ w_k^j \end{bmatrix} \\ &\quad - \left(y_{k+1} - \hat{y}_{k+1}\right)^T \begin{bmatrix} I \\ L^j \end{bmatrix}^T \bar{H}_1^{j+1} \begin{bmatrix} I \\ L^j \end{bmatrix} \left(y_{k+1} - \hat{y}_{k+1}\right) \\ &\quad = \begin{bmatrix} e_k \\ w_k^j \end{bmatrix}^T H_1^{j+1} \begin{bmatrix} e_k \\ w_k^j \end{bmatrix} - \left(\left(Ae_k - w_k^j\right) + w_k^j - w_k\right)^T \\ &\quad \bullet \begin{bmatrix} C \\ L^j C \end{bmatrix}^T \bar{H}_1^{j+1} \begin{bmatrix} C \\ L^j C \end{bmatrix} \left(\left(Ae_k - w_k^j\right) + w_k^j - w_k\right) \\ &\quad = \begin{bmatrix} e_k \\ w_k^j \end{bmatrix}^T H_1^{j+1} \begin{bmatrix} e_k \\ w_k^j \end{bmatrix} - \begin{bmatrix} e_k \\ w_k^j \end{bmatrix}^T \left(\begin{bmatrix} I \\ L^j C \end{bmatrix} \begin{bmatrix} A & -I \end{bmatrix}\right)^T \\ &\quad \bullet H_1^{j+1} \left(\begin{bmatrix} I \\ L^j C \end{bmatrix} \begin{bmatrix} A & -I \end{bmatrix}\right) \begin{bmatrix} e_k \\ w_k^j \end{bmatrix} \\ &\quad - 2\left(Ae_k - w_k^j\right) \begin{bmatrix} C \\ L^j C \end{bmatrix}^T \bar{H}_1^{j+1} \begin{bmatrix} C \\ L^j C \end{bmatrix} \left(w_k^j - w_k\right) \\ &\quad - \left(w_k^j - w_k\right)^T \begin{bmatrix} C \\ L^j C \end{bmatrix}^T \bar{H}_1^{j+1} \begin{bmatrix} C \\ L^j C \end{bmatrix} \left(w_k^j - w_k\right) \\ &\quad = \left(y_k - \hat{y}_k\right)^T Q_o \left(y_k - \hat{y}_k\right) + \left(w_k^j\right)^T R_o w_k^j \\ &\quad - 2\left(Ae_k - w_k^j\right) \begin{bmatrix} C \\ L^j C \end{bmatrix}^T \bar{H}_1^{j+1} \begin{bmatrix} C \\ L^j C \end{bmatrix} \left(w_k^j - w_k\right) \\ &\quad - \left(w_k^j - w_k\right)^T \begin{bmatrix} C \\ L^j C \end{bmatrix}^T \bar{H}_1^{j+1} \begin{bmatrix} C \\ L^j C \end{bmatrix} \left(w_k^j - w_k\right) \end{aligned} \tag{32}$$

Due to (22), (29) and (31), (32) becomes

$$
\begin{aligned}
&\begin{bmatrix} y_k - \hat{y}_k \\ w_k \end{bmatrix}^T \bar{H}_1^{j+1} \begin{bmatrix} y_k - \hat{y}_k \\ w_k \end{bmatrix} \\
&-\left(y_{k+1} - \hat{y}_{k+1}\right)^T \begin{bmatrix} I \\ L^j \end{bmatrix}^T \bar{H}_1^{j+1} \begin{bmatrix} I \\ L^j \end{bmatrix} \left(y_{k+1} - \hat{y}_{k+1}\right) \\
&= \left(y_k - \hat{y}_k\right)^T Q_o \left(y_k - \hat{y}_k\right) + (w_k^j)^T R_o w_k^j \\
&- 2e_k^T A^T P_o^{j+1} \left(w_k^j - w_k\right) + 2w_k^j P_o^{j+1} \left(w_k^j - w_k\right) \\
&-\left(w_k^j - w_k\right)^T P_o^{j+1} \left(w_k^j - w_k\right) \\
&= \left(y_k - \hat{y}_k\right)^T Q_o \left(y_k - \hat{y}_k\right) + (w_k^j)^T R_o w_k^j \\
&+ 2\left(y_k^T - \hat{y}_k^T\right)^T \bar{H}_{1,12}^{j+1} \left(w_k^j - w_k\right) \\
&+ w_k^j \left(\bar{H}_{1,22}^{j+1} - R_o\right)\left(w_k^j - w_k\right) \\
&+ w_k^T \left(\bar{H}_{1,22}^{j+1} - R_o\right)\left(w_k^j - w_k\right)
\end{aligned}
\tag{33}
$$

Due to (31), one has

$$
H_{1,ww}^{j+1} = \bar{H}_{1,22}^{j+1} \quad \text{and} \quad H_{1,we}^{j+1} = C^T \bar{H}_{1,12}^{j+1} \tag{34}
$$

Thus, iterative observer policy (27) can be rewritten as

$$
w_k^{j+1} = -(\bar{H}_{1,22}^{j+1})^{-1} (\bar{H}_{1,12}^{j+1})^T (y_k - \hat{y}_k) \tag{35}
$$

which indicates

$$
L^{j+1} = -(\bar{H}_{1,22}^{j+1})^{-1} (\bar{H}_{1,12}^{j+1})^T \tag{36}
$$

Theorem 1 is developed to prove iterative observer policy w_k^{j+1} (35) is going to converge to the optimal observer policy w_k^*, i.e. $\lim_{j\to\infty} w_k^{j+1} = w_k^*$.

Theorem 1: If the matrix CC^T is invertible, then there exists unique matrix $\bar{H}_1^{j+1}$, which satisfied with

$$
H_1^{j+1} = \begin{bmatrix} C & 0 \\ 0 & I \end{bmatrix}^T \bar{H}_1^{j+1} \begin{bmatrix} C & 0 \\ 0 & I \end{bmatrix} \tag{37}
$$

and (33), such that (35) converge to the optimal observer policy, i.e. $\lim_{j\to\infty} w_k^{j+1} = w_k^*$.

Proof: First, we shall prove that if $\bar{H}_1^{j+1}$ is the solution of (33), then the matrix H_1^{j+1} derived by (37) is the solution of (26). We know $y_k - \hat{y}_k = Ce_k$ and the dynamics of e_k is (30). If $\bar{H}_1^{j+1}$ is the solution of (33), then $\bar{H}_1^{j+1}$ will satisfy with the form below

$$\begin{aligned}
&\begin{bmatrix} y_k-\hat{y}_k \\ w_k \end{bmatrix}^T \bar{H}_1^{j+1} \begin{bmatrix} y_k-\hat{y}_k \\ w_k \end{bmatrix} - \left(\left(Ae_k - w_k^j \right) + w_k^j - w_k \right)^T \\
&\quad \bullet \begin{bmatrix} C \\ L^j C \end{bmatrix}^T \bar{H}_1^{j+1} \begin{bmatrix} C \\ L^j C \end{bmatrix} \left(\left(Ae_k - w_k^j \right) + w_k^j - w_k \right) \\
&\quad = \left(y_k - \hat{y}_k \right)^T Q_o \left(y_k - \hat{y}_k \right) + (w_k^j)^T R_o w_k^j \\
&\quad -2e_k^T A^T P_o^{j+1} \left(w_k^j - w_k \right) + 2w_k^j P_o^{j+1} \left(w_k^j - w_k \right) \\
&\quad -\left(w_k^j - w_k \right)^T P_o^{j+1} \left(w_k^j - w_k \right)
\end{aligned} \tag{38}$$

By (22) of Lemma 1 and from (38), H_1^{j+1} defined in (37)makes (26) hold. And then, we shall prove that there is only one solution $\bar{H}_1^{j+1}$ satisfied with (33). If there are two different solutions $\bar{H}_1^{j+1}$ and $\bar{W}_1^{j+1}$ both of which make (33) hold, then we get H_1^{j+1} computed by (37) and W_1^{j+1} computed by

$$W_1^{j+1} = \begin{bmatrix} C & 0 \\ 0 & I \end{bmatrix}^T \bar{W}_1^{j+1} \begin{bmatrix} C & 0 \\ 0 & I \end{bmatrix} \tag{39}$$

Since the matrix CC^T is invertible, we have

$$\begin{bmatrix} (CC^T)^{-1} & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} C & 0 \\ 0 & I \end{bmatrix} W_1^{j+1} \begin{bmatrix} C^T & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} (CC^T)^{-1} & 0 \\ 0 & I \end{bmatrix} = \bar{W}_1^{j+1} \tag{40}$$

and

$$\begin{bmatrix} (CC^T)^{-1} & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} C & 0 \\ 0 & I \end{bmatrix} H_1^{j+1} \begin{bmatrix} C^T & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} (CC^T)^{-1} & 0 \\ 0 & I \end{bmatrix} = \bar{H}_1^{j+1} \tag{41}$$

If H_1^{j+1} and W_1^{j+1} are the same, then $\bar{H}_1^{j+1}$ and $\bar{W}_1^{j+1}$ are equal. So the distinct H_1^{j+1} and W_1^{j+1} satisfy (26). However, there is only one solution of (26) for Problem 2. By contradiction, there is only one solution of (33). Due to (34), one has

$$\begin{aligned}
w_k^{j+1} &= -(\bar{H}_{1,22}^{j+1})^{-1} (\bar{H}_{1,12}^{j+1})^{\mathrm{T}} (y_k - \hat{y}_k) \\
&= -(H_{1,ww}^{j+1})^{-1} (H_{1,we}^{j+1})^T e_k
\end{aligned} \tag{42}$$

Since $\lim_{j\to\infty} w_k^{j+1} = \lim_{j\to\infty} \{-(H_{1,ww}^{j+1})^{-1} (H_{1,we}^{j+1})^T e_k\} = w_k^*$, then iterative observer policy w_k^{j+1} (35) converges to the optimal observer policy w_k^*. This completes the proof. ■

Remark 4. Theorem 1 shows that learning w_k^* by solving $\bar{H}_1^{j+1}$ from (33) requires the reversibility of CC^T. If the matrix C is full row rank then CC^T is invertible. Here, we require the matrix C can be identified is invertible or not based on the practical application, even it is inaccurate.

Let $e_{y_k} = y_k - \hat{y}_k$, (33) can be further rewritten as

$$\theta^j(k)h_o^{j+1} = \rho_k^j \tag{43}$$

where

$$\rho_k^j = e_{y_k}^T Q_o e_{y_k} + w_k^T R_o w_k ,$$

$$h_o^{j+1} = \left[(vec(\bar{H}_{1,11}^{j+1}))^T \quad (vec(\bar{H}_{1,11}^{j+1}))^T \quad (vec(\bar{H}_{1,22}^{j+1}))^T \right]^{\mathrm{T}},$$

$$\theta^j(k) = \left[\theta_1^j \quad \theta_2^j \quad \theta_3^j \right],$$

$$\begin{aligned}
\theta_1^j &= e_{y_k}^T \otimes e_{y_k}^T - e_{y_{k+1}}^T \otimes e_{y_{k+1}}^T \\
\theta_2^j &= 4e_{y_k}^T \otimes w_k - 2e_{y_{k+1}}^T \otimes \left(L^j e_{y_{k+1}} \right)^T - 2e_{y_k}^T \otimes \left(L^j e_{y_k} \right)^T \\
\theta_3^j &= 2w_k^T \otimes w_k^T - \left(L^j e_{y_{k+1}} \right)^T \otimes \left(L^j e_{y_{k+1}} \right)^T \\
&\quad - \left(L^j e_{y_k} \right)^T \otimes \left(L^j e_{y_k} \right)^T
\end{aligned} \tag{44}$$

Algorithm 2 is presented to learn $\bar{H}_1^{j+1}$.

Algorithm 2: Off-policy Q-learning algorithm

1. Data collection: Choose an arbitrary admissible observer policy as a behavior policy w_k , system data e_{y_k} are collected and stored in the sample sets $\theta^j(k)$ and ρ_k^j ;

2. Initiation: Choose an initial gain L^0 , such that $w_k^0 = L^0 e_{y_k}$ makes (4) stable. Let $j = 0$;

3. Implementing off-policy Q-learning: By using least squares methods for (43), $\bar{H}_1^{j+1}$ can be estimated using the collected data in Step 1, and then L^{j+1} can be updated in terms of (36);

4. If $\left\| L^{j+1} - L^j \right\| \le \varepsilon$ (ε is some small positive numbers), then stop the iteration and the optimal control policy has been obtained; Otherwise, let $j = j + 1$ and go back to Step 3.

Remark 5. Different from [16] where the past inputs and outputs are used to estimate the states of systems, a state observer is employed in this paper, such that the unmeasured states can be approximated by the observer states. Introducing of the matrix $\bar{H}_1$ makes it possible to learn the optimal observer gain using only measured data. It is worth pointing out that it is the first time to design optimal observer by using a way that is independent of the system matrices A , B and C , except for requiring CC^T to be invertible.

Remark 6. If a proportional and integral observer is employed instead of proportional observer (3) for estimating the state of discrete-time linear system (1), then it is potential that the optimal observer policy with proportional gain and integral gain can be learned by using the proposed data-driven off-policy Q-learning method since the separation principle still holds for the case of proportional and integral (PI) observer state-based feedback control for linear systems [30, 31].

4. Optimal Controller Design

In this section, we shall solve Subproblem 1 to find the optimal control policy using only measured data.

4.1. On-policy Q-learning for the optimal controller

Let $u_k = Kx_k$ be an admissible control policy. For solving Subproblem 1, a value function and an action-dependent Q-function can be respectively defined as [2, 26, 27]

$$V(x_k) = \frac{1}{2}\sum_{i=k}^{\infty} y_i^T Q_p y_i + u_i^T R_p u_i \tag{45}$$

and

$$Q(x_k, u_k) = \frac{1}{2}(y_k^T Q_p y_k + u_k^T R_p u_k) + V(x_{k+1}) \tag{46}$$

Thus, the following relation holds

$$\begin{aligned} V^*(x_k) &= \min_{u_k} \frac{1}{2}\sum_{i=k}^{\infty} y_i^T Q_p y_i + u_i^T R_p u_i \\ &= \min_{u_k} Q(x_k, u_k) \\ &= Q(x_k, u_k^*) \end{aligned} \tag{47}$$

According to the dynamic programming theory, the Q-function based ARE can be derived below

$$\begin{aligned} Q(x_k, u_k^*) &= \min_{w_k}\left\{\frac{1}{2}(y_k^T Q_p y_k + u_k^T R_p u_k) + V^*(x_{k+1})\right\} \\ &= \frac{1}{2}(y_k^T Q_p y_k + (u_k^*)^T R_p u_k^*) + Q(x_{k+1}, u_{k+1}^*) \end{aligned} \tag{48}$$

From (47), it follows that the optimal control policy should be with the form of

$$u_k^* = \arg\min_{u_k} Q(x_k, u_k) \tag{49}$$

The following lemma is useful to find the optimal control policy.

Lemma 2 [27, 28]: For Subproblem 1, under the admissible policy $u_k = Kx_k$, the value function and the Q-function have the quadratic forms of

$$V(x_k) = \frac{1}{2}x^T(k) P_2 x(k) \tag{50}$$

and

$$Q(x_k, u_k) = \frac{1}{2}\begin{bmatrix} x_k \\ u_k \end{bmatrix}^T H_2 \begin{bmatrix} x_k \\ u_k \end{bmatrix} \tag{51}$$

where

$$\begin{aligned} H_2 &= \begin{bmatrix} H_{2,xx} & H_{2,xu} \\ * & H_{2,uu} \end{bmatrix} \\ &= \begin{bmatrix} A^T P_2 A + \tilde{Q}_p & A^T P_2 B \\ B^T P_2 A & B^T P_2 B + R_p \end{bmatrix} \end{aligned} \tag{52}$$

$$P_2 = \begin{bmatrix} I \\ K \end{bmatrix}^T H_2 \begin{bmatrix} I \\ K \end{bmatrix} \tag{53}$$

where $P_2 = P_2^T > 0$ and $\tilde{Q}_p = C^T Q_p C$.

Substituting (51) into ARE (48), one has

$$\begin{bmatrix} x_k \\ u_k^* \end{bmatrix}^T H_2 \begin{bmatrix} x_k \\ u_k^* \end{bmatrix} = y_k^T Q_p y_k + (u_k^*)^T R_p u_k^* + \begin{bmatrix} x_{k+1} \\ u_{k+1}^* \end{bmatrix}^T H_2 \begin{bmatrix} x_{k+1} \\ u_{k+1}^* \end{bmatrix} \tag{54}$$

Based on the necessary condition of optimality, implementing $\frac{\partial Q(x_k, u_k)}{\partial u_k} = 0$ yields

$$u_k^* = -\left(H_{2,uu}\right)^{-1} H_{2,ux} x_k \tag{55}$$

Since $u_k = Kx_k$, one has

$$K^* = -\left(H_{2,uu}\right)^{-1} H_{2,ux} \tag{56}$$

Theorem 2: If $K^* = -\left(H_{2,uu}\right)^{-1} H_{2,ux}$ and H_2 makes ARE (54) hold, then $u_k = K^* \hat{x}_k$ is the optimal control policy for Problem 1.

Proof: Note that $K^* = -\left(H_{2,uu}\right)^{-1} H_{2,ux}$ is the optimal controller gain of Subproblem 1. From (9), the performance index (8) of system (1) is only affected by controller gain K which is independent of the observer gain L in terms of Separation Principle. And the performance index in Problem 1 is the same as that in Subproblem 1. Therefore $u_k = K^* \hat{x}_k$ can minimize the performance index in Problem 1. This completes the proof. ■

The optimal controller gain K^* can be obtained if the Q-function matrix H_2 is solved from (54). To learn H_2, Algorithm 3 is provided by referring to [11, 12, 27, 28].

Algorithm 3: On-policy iteration algorithm

1. Initialization: Given stabilizing controller gain K^0, and let $j = 0$, where j denotes iteration index;

2. Policy evaluation by solving Q-function matrix H_2^{j+1}:

$$\begin{aligned} &\begin{bmatrix} x_k \\ K^j x_k \end{bmatrix}^T H_2^{j+1} \begin{bmatrix} x_k \\ K^j x_k \end{bmatrix} = y_k^T Q_p y_k + (K^j x_k)^T R_p K^j x_k \\ &+ \begin{bmatrix} x_{k+1} \\ K^j x_{k+1} \end{bmatrix}^T H_2^{j+1} \begin{bmatrix} x_{k+1} \\ K^j x_{k+1} \end{bmatrix} \end{aligned} \tag{57}$$

where $x_{k+1} = Ax_k + BK^j x_k$.

3. Policy update:

$$K^{j+1} = -\left(H_{2,uu}^{j+1}\right)^{-1} H_{2,ux}^{j+1} \tag{58}$$

4. Stop when $\left\| H_2^{j+1} - H_2^j \right\| \le \varepsilon$ with a small constant ε $(\varepsilon > 0)$; Otherwise, let $j = j + 1$ and go back to Step 2.

Remark 7. [27, 28] have proven $\lim_{j \to \infty} H_2^{j+1} = H_2$ (H_2 in (54)) and $\lim_{j \to \infty} K^{j+1} = K^*$ if online implementing algorithm 3. It is worth pointing out that Algorithm 3 does not need to know the system matrices A, B and C, but the states x_k should be measurable when solving H_2^{j+1} in (57).

However, the network-induced delay and indirectly measured state make state x_k unavailable. Even though state x_k is actually replaced by the predicted $\hat{x}_k$ at the time instant k, it is impossible to predict the observer states $\hat{x}_k$ in (5) without knowing the accurate system matrices A and B. Therefore, Algorithm 3 cannot work for learning the optimal controller gain K^* at this situation.

For overcoming these obstacles, we shall develop an off-policy Q-learning algorithm. First, let us analyze the controller in the networked control system in Fig. 1.

According to the different cases of network-induced delays, a new variable $\bar{z}(k)$ is defined correspondingly using the similar way to [21].

Case 1: If d_k=0, then

$$\bar{z}(k)=[\underbrace{\hat{x}_k^T \quad 0 \quad \dots \quad 0}_{\delta_{\max}+1}\underbrace{0 \quad \dots \quad 0}_{\delta_{\max}} \underbrace{0 \quad \dots \quad 0}_{\delta_{\max}}]^T \tag{59}$$

Case 2: If d_k=1, then

$$\bar{z}(k)=[\underbrace{0 \quad \hat{x}_{k-1}^T \quad \dots \quad 0}_{\delta_{\max}+1}\underbrace{u_{k-1}^T \quad \dots \quad 0}_{\delta_{\max}} \underbrace{Le_{k-1}^T \quad \dots \quad 0}_{\delta_{\max}}]^T \tag{60}$$

...

Case $\delta_{\max}+1$: If $d_k=\delta_{\max}$, then

$$\bar{z}(k)=[\underbrace{0\cdots\hat{x}_{k-\delta_{\max}}^T}_{\delta_{\max}+1}\underbrace{u_{k-1}^T \quad \dots \quad u_{k-\delta_{\max}}^T}_{\delta_{\max}} \underbrace{Le_{k-1}^T \quad \dots \quad Le_{k-\delta_{\max}}^T}_{\delta_{\max}}]^T \tag{61}$$

Referring to Smith predictor (5), the predicted observer state can be rewritten as

$$\hat{x}_k = M\bar{z}_k \tag{62}$$

where $M=[M_1\ A^{\delta_{\max}}\ M_2\ M_1]$, $M_1=[I\ A\cdots A^{\delta_{\max}-1}]$, $M_2=[B\ AB\cdots A^{\delta_{\max}-1}B]$. Then, the predicted observer state-based feedback controller in Fig. 1 is in fact

$$u_k = K\hat{x}_k = \bar{K}\bar{z}_k \tag{63}$$

where $\bar{K}=KM$.

Now, we are in the position of finding $\bar{K}^*$ which satisfies $\bar{K}^*=K^*M$. Here, another new variable $\hat{z}(k)$ is defined as:

Case 1: If d_k=0, then

$$\hat{z}(k)=[\underbrace{x_k^T \quad 0 \quad \dots \quad 0}_{\delta_{\max}+1}\underbrace{0 \quad \dots \quad 0}_{\delta_{\max}}]^T \tag{64}$$

Case 2: If d_k=1, then

$$\hat{z}(k)=[\underbrace{0 \quad x_{k-1}^T \quad \dots \quad 0}_{\delta_{\max}+1}\underbrace{u_{k-1}^T \quad \dots \quad 0}_{\delta_{\max}}]^T \tag{65}$$

...

Case $\delta_{\max}+1$: If $d_k=\delta_{\max}$, then

$$\hat{z}(k)=[\underbrace{0\cdots x_{k-\delta_{\max}}^T}_{\delta_{\max}+1}\underbrace{u_{k-1}^T \quad \cdots \quad u_{k-\delta_{\max}}^T}_{\delta_{\max}}]^T \tag{66}$$

Similar to the derivation of (5), one has $x_k=\hat{M}\hat{z}_k$, where $\hat{M}=[M_1\ A^{\delta_{\max}}\ M_2]$. In terms of the definitions of M and $\hat{M}$, one can find

$$M=\hat{M}\hat{I}_1\hat{I}_2+[\hat{M}\ 0] \tag{67}$$

where $\hat{I}_1=[I\ 0\ 0]^T$ and $\hat{I}_2=[0\ 0\ 0\ I]$. Let $\hat{K}^*=K^*\hat{M}$, then

$$\bar{K}^*=K^*M=\hat{K}^*\hat{I}_1\hat{I}_2+[\hat{K}^*\ 0] \tag{68}$$

So, if $\hat{K}^*$ can be learned even though the system matrices A, B and C are inaccurate or unknown, then $\bar{K}^*$ can be obtained for systems with the inaccurate system matrices A, B and C.

Next, an off-policy Q-learning algorithm is present to learn $\hat{K}^*$ in subsection B.

4.2. Off-policy Q-learning for the optimal controller

Introducing an auxiliary variable BK^jx_k into system (1) yields

$$\begin{aligned} x_{k+1}&=Ax_k+BK^jx_k+B\left(u_k-K^jx_k\right)\\ &=\left(A+BK^j\right)\hat{M}\hat{z}_k+B\left(u_k-K^j\hat{M}\hat{z}_k\right)\end{aligned} \tag{69}$$

where u_k is called as the behavior policy to generate data and $K^jx_k=\hat{K}^j\hat{z}_k$ ($\hat{K}^j=K^j\hat{M}$) is viewed as the target policy needed to be learned.

Actually, (57) is equivalent to the form below

$$\begin{aligned}&\begin{bmatrix}I\\K^j\end{bmatrix}^T H_2^{j+1}\begin{bmatrix}I\\K^j\end{bmatrix}=C^TQ_pC_k+(K^j)^TR_pK^j\\&+(A+BK^j)^T\begin{bmatrix}I\\K^j\end{bmatrix}^T H_2^{j+1}\begin{bmatrix}I\\K^j\end{bmatrix}(A+BK^j)\end{aligned} \tag{70}$$

Along the trajectory of (69), one has

$$\begin{aligned}&x_k^T\begin{bmatrix}I\\K^j\end{bmatrix}^T H_2^{j+1}\begin{bmatrix}I\\K^j\end{bmatrix}x_k\\&-x_{k+1}^T\begin{bmatrix}I\\K^j\end{bmatrix}^T H_2^{j+1}\begin{bmatrix}I\\K^j\end{bmatrix}x_{k+1}\\&=x_k^T\begin{bmatrix}I\\K^j\end{bmatrix}^T H_2^{j+1}\begin{bmatrix}I\\K^j\end{bmatrix}x_k\\&-\left(Ax_k+BK^jx_k+B\left(u_k-K^jx_k\right)\right)^T\begin{bmatrix}I\\K^j\end{bmatrix}^T\\&\bullet H_2^{j+1}\begin{bmatrix}I\\K^j\end{bmatrix}\left(Ax_k+BK^jx_k+B\left(u_k-K^jx_k\right)\right)\end{aligned} \tag{71}$$

Due to (70) and $x_k=\hat{M}\hat{z}_k$, (71) becomes

$$\begin{aligned}
&\begin{bmatrix} \hat{z}_k \\ \hat{K}^j \hat{z}_k \end{bmatrix}^T \bar{H}_2^{j+1} \begin{bmatrix} \hat{z}_k \\ \hat{K}^j \hat{z}_k \end{bmatrix} - \begin{bmatrix} \hat{z}_{k+1} \\ \hat{K}^{j+1} \hat{z}_{k+1} \end{bmatrix}^T \bar{H}_2^{j+1} \begin{bmatrix} \hat{z}_{k+1} \\ \hat{K}^{j+1} \hat{z}_{k+1} \end{bmatrix} \\
&= \begin{bmatrix} \hat{z}_k \\ \hat{K}^j \hat{z}_k \end{bmatrix}^T \begin{bmatrix} M^T \tilde{Q}_p M & 0 \\ 0 & R_p \end{bmatrix} \begin{bmatrix} \hat{z}_k \\ \hat{K}^j \hat{z}_k \end{bmatrix} \\
&-2x_k^T (A + BK^j)^T \begin{bmatrix} I \\ K^j \end{bmatrix}^T H_2^{j+1} \begin{bmatrix} I \\ K^j \end{bmatrix} B(u_k - \hat{K}^j \hat{z}_k) \\
&-(u_k - \hat{K}^j \hat{z}_k)^T B^T \begin{bmatrix} I \\ K^j \end{bmatrix}^T H_2^{j+1} \begin{bmatrix} I \\ K^j \end{bmatrix} B(u_k - \hat{K}^j \hat{z}_k)
\end{aligned} \tag{72}$$

where

$$\bar{H}_2^{j+1} = \begin{bmatrix} \hat{M} & 0 \\ 0 & I \end{bmatrix}^T H_2^{j+1} \begin{bmatrix} \hat{M} & 0 \\ 0 & I \end{bmatrix} \tag{73}$$

By Lemma 2, one has

$$\begin{aligned}
\bar{H}_2^{j+1} &= \begin{bmatrix} \hat{M} & 0 \\ 0 & I \end{bmatrix}^T \begin{bmatrix} A^{\mathrm{T}} P_2^{j+1} A + \tilde{Q}_p & A^{\mathrm{T}} P_2^{j+1} B \\ B^{\mathrm{T}} P_2^{j+1} A & B^{\mathrm{T}} P_2^{j+1} B + R_p \end{bmatrix} \begin{bmatrix} \hat{M} & 0 \\ 0 & I \end{bmatrix} \\
&= \begin{bmatrix} \hat{M}^T (A^{\mathrm{T}} P_2^{j+1} A + \tilde{Q}_p) \hat{M} & \hat{M}^T A^{\mathrm{T}} P_2^{j+1} B \\ * & B^{\mathrm{T}} P_2^{j+1} B + R_p \end{bmatrix} \\
&= \begin{bmatrix} \bar{H}_{2,zz}^{j+1} & \bar{H}_{2,zu}^{j+1} \\ * & \bar{H}_{2,uu}^{j+1} \end{bmatrix}
\end{aligned} \tag{74}$$

and

$$\begin{aligned}
&-2x_k^T (A + BK^j)^T \begin{bmatrix} I \\ K^j \end{bmatrix}^T H_2^{j+1} \begin{bmatrix} I \\ K^j \end{bmatrix} B(u_k - \hat{K}^j \hat{z}_k) \\
&= -2\hat{z}_k^T (A\hat{M} + BK^j \hat{M})^T P^{j+1} B(u_k - \hat{K}^j \hat{z}_k) \\
&= -2\hat{z}_k^T \bar{H}_{2,zu}^{j+1} (u_k - \hat{K}^j \hat{z}_k) - 2u_k^j (\bar{H}_{2,uu}^{j+1} - R_p)(u_k - \hat{K}^j \hat{z}_k)
\end{aligned} \tag{75}$$

and

$$\begin{aligned}
&-(u_k - \hat{K}^j \hat{z}_k)^T B^T \begin{bmatrix} I \\ K^j \end{bmatrix}^T H_2^{j+1} \begin{bmatrix} I \\ K^j \end{bmatrix} B(u_k - \hat{K}^j \hat{z}_k) \\
&= -(u_k - \hat{K}^j \hat{z}_k)^T B^T P_2^{j+1} B(u_k - \hat{K}^j \hat{z}_k) \\
&= -(u_k - \hat{K}^j \hat{z}_k)^T (\bar{H}_{2,uu}^{j+1} - R_p)(u_k - \hat{K}^j \hat{z}_k)
\end{aligned} \tag{76}$$

Thus, (72) becomes

$$\begin{aligned}
&\begin{bmatrix} \hat{z}_k \\ \hat{K}^j \hat{z}_k \end{bmatrix}^T \bar{H}_2^{j+1} \begin{bmatrix} \hat{z}_k \\ \hat{K}^j \hat{z}_k \end{bmatrix} - \begin{bmatrix} \hat{z}_{k+1} \\ \hat{K}^{j+1} \hat{z}_{k+1} \end{bmatrix}^T \bar{H}_2^{j+1} \begin{bmatrix} \hat{z}_{k+1} \\ \hat{K}^{j+1} \hat{z}_{k+1} \end{bmatrix} \\
&= \begin{bmatrix} \hat{z}_k \\ \hat{K}^j \hat{z}_k \end{bmatrix}^T \begin{bmatrix} \hat{M}^T \tilde{Q}_p \hat{M} & 0 \\ 0 & R_p \end{bmatrix} \begin{bmatrix} \hat{z}_k \\ \hat{K}^j \hat{z}_k \end{bmatrix} \\
&-2\hat{z}_k^T \bar{H}_{2,zu}^{j+1} (u_k - \hat{K}^j \hat{z}_k) + (\hat{K}^j \hat{z}_k)^T (\bar{H}_{2,uu}^{j+1} - R_p) \hat{K}^j \hat{z}_k \\
&-u_k^T (\bar{H}_{2,uu}^{j+1} - R_p) u_k
\end{aligned} \tag{77}$$

Further, (77) can rewritten as

$$\varphi^j(k) h_c^{j+1} = \beta_k^j \tag{78}$$

where

$$\beta_k^j = y_k^{\mathrm{T}} Q_p y_k + u_k^{\mathrm{T}} R_p u_k ,$$

$$h_c^{j+1} = \left[(vec(\bar{H}_{2,zz}^{j+1}))^{\mathrm{T}} \quad (vec(\bar{H}_{2,zu}^{j+1}))^{\mathrm{T}} \quad (vec(\bar{H}_{2,uu}^{j+1}))^{\mathrm{T}} \right]^{\mathrm{T}},$$

$$\varphi^j(k) = \left[\varphi_1^j(k) \quad \varphi_2^j(k) \quad \varphi_3^j(k) \right], \tag{79}$$

$$\begin{aligned}
\varphi_1^j(k) &= \hat{z}_k^T \otimes \hat{z}_k^T - \hat{z}_{k+1}^T \otimes \hat{z}_{k+1}^T \\
\varphi_2^j(k) &= 2\left(\hat{z}_k^T \otimes \left(\hat{K}^j \hat{z}_k \right)^T - \hat{z}_{k+1}^T \otimes \left(\hat{K}^j \hat{z}_{k+1} \right)^T \right) \\
&\quad + 2\hat{z}_k^T \otimes \left(u_k - \hat{K}^j \hat{z}_k \right)^T \\
\varphi_3^j(k) &= u_k^T \otimes u_k^T - \left(\hat{K}^j \hat{z}_{k+1} \right)^T \otimes \left(\hat{K}^j \hat{z}_{k+1} \right)^T
\end{aligned}$$

Theorem 3: There exists a unique matrix $\bar{H}_2^{j+1}$ which satisfied with (78), and $\hat{K}^j$ converges to $\hat{K}^*$ as $j \to \infty$.

Proof: Because there exists a unique solution H_2^{j+1} to (70), there exists a matrix $\bar{H}_2^{j+1}$ that makes (78) by the derivation of (78) from (70). Now, the uniqueness of solution to (78) is going to be proved. Assume there are two distinct solutions $\bar{H}_2^{j+1}$ and $\bar{W}_2^{j+1}$ to (78), from (73), one has two distinct matrices H_2^{j+1} and W_2^{j+1}. By contradiction, there is only one solution $\bar{H}_2^{j+1}$ to (78).

By (58) and (68), one has

$$\hat{K}^{j+1} = K^{j+1}\hat{M} = -\left(H_{2,uu}^{j+1} \right)^{-1} H_{2,ux}^{j+1} \hat{M} \tag{80}$$

By Lemma 2, one has

$$\begin{aligned}
H_2^{j+1} &= \begin{bmatrix} H_{2,xx}^{j+1} & H_{2,xu}^{j+1} \\ * & H_{2,uu}^{j+1} \end{bmatrix} \\
&= \begin{bmatrix} A^{\mathrm{T}} P_2^{j+1} A + \tilde{Q}_p & A^{\mathrm{T}} P_2^{j+1} B \\ B^{\mathrm{T}} P_2^{j+1} A & B^{\mathrm{T}} P_2^{j+1} B + R_p \end{bmatrix}
\end{aligned} \tag{81}$$

Compared (74) with (81), it is not difficult to find

$$\bar{H}_{2,zu}^{j+1} = \hat{M} H_{2,xu}^{j+1} \quad \text{and} \quad \bar{H}_{2,uu}^{j+1} = H_{2,uu}^{j+1} \tag{82}$$

Then (80) becomes

$$\hat{K}^{j+1} = -\left(\bar{H}_{2,uu}^{j+1} \right)^{-1} \bar{H}_{2,u\bar{z}}^{j+1} \tag{83}$$

and

$$\begin{aligned}
\lim_{j\to\infty} \hat{K}^{j+1} &= \lim_{j\to\infty} \left(-\left(\bar{H}_{2,uu}^{j+1} \right)^{-1} \bar{H}_{2,u\bar{z}}^{j+1} \right) \\
&= \lim_{j\to\infty} \left(-\left(H_{2,uu}^{j+1} \right)^{-1} H_{2,ux}^{j+1} \hat{M} \right) \\
&= K^* \hat{M} = \hat{K}^*
\end{aligned} \tag{84}$$

This completes the proof. ■

If $\hat{K}^*$ can be found by learning $\bar{H}_2$, then $\bar{K}^* = \hat{K}^* \hat{I}_1 \hat{I}_2 + [\hat{K}^* \ 0]$ can be calculated. The approximate optimal control law $u_k^* = \bar{K}^* \bar{z}_k = K^* \hat{x}_k$ can be derived for Problem 1. To learn $\bar{H}_2^{j+1}$, Algorithm 4 is presented as follows:

Algorithm 4: Off-policy Q-learning algorithm for learning $\bar{H}_2^{j+1}$

1. Data collection: Choose a behavior policy u_k and a behavior observer policy w_k to act the plant and the observer in Fig. 1. When $\hat{y}_{k-d_k}$ approaches the output y_{k-d_k} of the system as close as it can, x_{k-d_k} in $\hat{z}_k$ is replaced by $\hat{x}_{k-d_k}$ and store $\hat{z}_k$ and u_k in the sample sets φ^j and β_k^j. The system data e_{y_k} are collected and stored in the sample sets $\theta^j(k)$ and ρ_k^j;
2. Learning the optimal observer gain: Implementing step 2-step 4 in Algorithm 2 to find the approximately optimal observer gain;
3. Learning $\hat{K}^*$:

3.1 Choose the initial stabilizing gain $\hat{K}^0$ and let $j = 0$;

3.2 By using the least squares method, $\bar{H}_2^{j+1}$ in (78) can be estimated using the collected data in Step 1, and then $\hat{K}^{j+1}$ can be updated in terms of (83);

3.3 If $\left\| \bar{H}_2^{j+1} - \bar{H}_2^j \right\| \le \varepsilon$ ($\varepsilon > 0$), then stop the iteration and $\hat{K}^*$ can be estimated by $\hat{K}^{j+1}$, thus $\bar{K}^*$ can further be calculated (68); Otherwise, let $j = j + 1$ and go back to Step 3.2.

Remark 8. As shown by the proof of Theorem 2 and the derivation of (78) and (83), Algorithm 4 used for learning $\hat{K}^*$ is proposed, such that $\bar{K}^*$ can be finally calculated just with the help of (57) and (58) in Algorithm 3.

Remark 9. For system (1) subject to inaccurate system matrices, network-induced delays and unmeasured states, the optimal observer gain and the optimal controller gain can be found by implementing Algorithm 4 using only measured inputs, outputs and observer errors, not system matrices A, B and C. This is different from [10-14, 22, 26-29] without consideration of network-induced delays and unmeasurable states, and [21, 23] where states of systems are assumed to be measurable. Moreover, it worth pointing out that until now it has not been reported about off-policy Q-learning algorithm used for solving optimal controller for networked control systems with accurate model parameters, network-induced delay and together with unavailable sates information.

Remark 10. Notice that the number of unknown elements in the iterative matrix $\bar{H}_2^{j+1}$ is $(m(\delta_{\max}+1)+n(\delta_{\max}+1)+1)\times \ (m(\delta_{\max}+1)+n(\delta_{\max}+1))/2$ in (78). If no network-induced delay is considered, then it is $(m+n+1)\times(m+n)/2$. While, the number of unknown elements of the iterative matrix $\bar{P}^{j+1}$ in [16] without network-induced delay is $(Nm+Np)\times(Nm+Np+1)/2$ $(Np \ge n,\ N \ge 1)$. So, the computational complexity is less than the method in [16].

5. Simulation Results

In this section, the proposed off-policy Q-learning algorithm for predicted observer state based feedback control of system (1) is verified respectively for systems with and without network-induced delays. Moreover, simulations show the efficiency of the designed observer and predictor for alleviating the negative influence of network-induced delays on the control performance of systems.

Consider the following discrete-time model of F-16 aircraft autopilot [27, 28]:

$$\begin{aligned} x_{k+1} &= \begin{bmatrix} 0.906488 & 0.0816012 & -0.0005 \\ 0.074349 & 0.90121 & -0.000708383 \\ 0 & 0 & 0.132655 \end{bmatrix} x_k \\ &\quad + \begin{bmatrix} -0.00150808 \\ -0.0096 \\ 0.867345 \end{bmatrix} u_k \end{aligned} \tag{85}$$

$$y_k = \begin{bmatrix} 1 & 0 & 0 \\ 2 & 1 & 1 \\ 1 & 0 & 3 \end{bmatrix} x_k$$

Choose $Q_1 = Q_2 = \mathrm{diag}(10,10,10)$, $R_1 = \mathrm{diag}(1,1,1)$ and $R_2 = 1$.

5.1. Model-based the optimal observer and the optimal controller

Using the command "dare" in Matlab yields the optimal Q-function matrix H_1^* and the optimal observer gain L^*C in terms of (21) and (24). Further, $\bar{H}_1^*$ can be calculated by (29).

$$\begin{aligned} \bar{H}_1^* &= \begin{bmatrix} 21.4051 & 1.2230 & 0.4671 & -15.8013 \\ 1.2230 & 22.0225 & -3.4352 & -23.1629 \\ 0.4671 & -3.4352 & 11.1489 & 5.5249 \\ -15.8013 & -23.1629 & 5.5249 & 61.7846 \\ 0.0737 & -11.2431 & 3.3115 & 20.1982 \\ -19.8418 & -13.0895 & -0.0488 & 50.0020 \end{bmatrix} \\ &\quad \begin{bmatrix} 0.0737 & -19.8418 \\ -11.2431 & -13.0895 \\ 3.3115 & -0.0488 \\ 20.1982 & 50.0020 \\ 11.6467 & 9.9968 \\ 9.9968 & 101.0173 \end{bmatrix} \end{aligned} \tag{86}$$

and the optimal observer gain

$$L^*C = \begin{bmatrix} 0.8539 & 0.1610 & 0.0025 \\ 0.1426 & 0.7030 & -0.0038 \\ 0.0193 & -0.0197 & 0.1301 \end{bmatrix} \tag{87}$$

For Subproblem 1, using the command "dare" in Matlab yields the optimal Q-function matrix H_2^* and the gain K^*.

$$H_2^* = \begin{bmatrix} 761.9176 & 630.3343 & 55.3593 & 33.2776 \\ 630.3343 & 569.9991 & 11.0594 & 5.2257 \\ 55.3593 & 11.0594 & 101.7525 & 11.4653 \\ 33.2776 & 5.2257 & 11.4653 & 76.0158 \end{bmatrix} \tag{88}$$

$$K^* = \begin{bmatrix} -0.4378 & -0.0687 & -0.1508 \end{bmatrix} \tag{89}$$

5.2. Learning the optimal observer and the optimal controller

We assume that the system matrices A, B and C of system (85) are not accurately known. First, the network-induced delay is not taken into account. In this case, the maximum delay upper bound $\delta_{\max} = 0$, $\hat{z}_k = x_k$ and $M = I$. Here x_k will be replaced by $\hat{x}_k$ when e_k is very close to zero. By Algorithm 4 (similar to the algorithms in [11, 12, 27]). Fig. 2 plots the results of convergence of $\bar{H}_1^j$ and L^j when implementing the off-policy Q-learning after 6 iterations, and we get

$$L^6 = \begin{bmatrix} 0.8294 & 0.0755 & -0.0005 \\ 0.0689 & 0.8246 & -0.0006 \\ 0 & 0 & 0.1206 \end{bmatrix} \tag{90}$$

and

$$\begin{aligned} L^6 * C &= \begin{bmatrix} 0.8294 & 0.0755 & -0.0005 \\ 0.0689 & 0.8246 & -0.0006 \\ 0 & 0 & 0.1206 \end{bmatrix} \\ &= L^* C \end{aligned} \tag{91}$$

which means that the optimal observer gain L^* has been learned using Algorithm 4 using only available data.

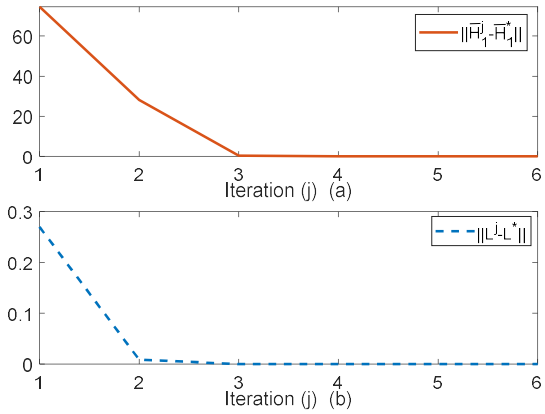

Figure 2: Convergence of matrices $\bar{H}_1^j$ and L^j

Moreover, implementing Algorithm 4 also yields

$$\bar{H}_2^{11} = \begin{bmatrix} 761.9176 & 630.3343 & 55.3593 & 33.2776 \\ 630.3343 & 569.9991 & 11.0594 & 5.2257 \\ 55.3593 & 11.0594 & 101.7525 & 11.4653 \\ 33.2776 & 5.2257 & 11.4653 & 76.0158 \end{bmatrix} \tag{92}$$

and

$$\hat{K}^{11} = \begin{bmatrix} -0.4378 & -0.0687 & -0.1508 \end{bmatrix} \tag{93}$$

When no network-induced delay occurred, one can notice that $\bar{H}_2^{11}$ and $\hat{K}^{11}$ respectively converge to H_2^* and K^*.

Second, if the maximum network-induced delay bound $\delta_{\max} = 1$, then $\hat{M} = [I \; A \; B]$ and $M = [I \; A \; B \; I]$. Implementing Algorithm 4 yields

$$\begin{aligned} \hat{K}^{10} = \begin{bmatrix} -0.4378 & -0.0687 & -0.1508 & -0.4019 \\ -0.0977 & -0.0197 & -0.1295 \end{bmatrix} \end{aligned} \tag{94}$$

and $L^6 = L^*$. One can find that $\hat{K}^{10}$ is equal to $K^*\hat{M}$. Further, we can calculate

$$\begin{aligned} \bar{K}^{10} &= \hat{K}^{10}\hat{I}_1\hat{I}_2 + [\hat{K}^{10} \; 0] \\ &= [-0.4378 \;\; -0.0687 \;\; -0.1508 \;\; -0.4019 \;\; -0.0977 \\ &\quad -0.0197 \;\; -0.1295 \;\; -0.4378 \;\; -0.0687 \;\; -0.1508] \end{aligned} \tag{95}$$

where

$$\hat{I}_1 = \begin{bmatrix} I_{3\times 3} \\ 0 \\ 0 \end{bmatrix}, \quad \hat{I}_2 = \begin{bmatrix} 0 & 0 & 0 & I_{3\times 3} \end{bmatrix}, \quad I_{3\times 3} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

which shows $\bar{K}^{10} = \bar{K}^* = K^*M$. Then $u_k^* = \bar{K}^*\bar{z}_k$ can be obtained. Fig. 3 presents the detailed the convergence results.

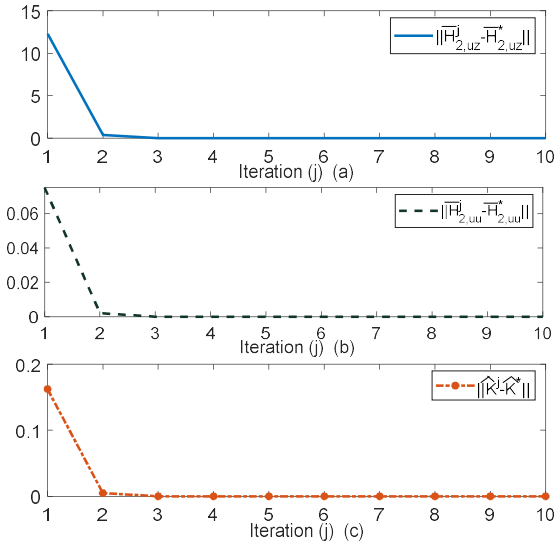

Figure 3: Convergence of matrices $\bar{H}_2^j$ and $\hat{K}^j$

5.3. Simulations and comparisons

Choose the initial states of the system and its observer $x_0 = [-1 \quad 2 \quad 1]^T$ and $\hat{x}_0 = [0 \quad 1 \quad -1]^T$. For the case of absence of network-induced delays, Fig. 4 is given to show the system sate trajectories and estimated system state trajectories, as well as the estimated errors using the optimal controller gain (93) and the optimal observer gain (90) (see Fig. 5).

For the case there exist network-induced delays plotted in Fig. 6, we choose the initial states of the system and its observer $x_0 = [-1 \quad 2 \quad 1]^T$, $\hat{x}_0 = [0 \quad 1 \quad -1]^T$, $x_{-1} = [1.5 \quad 0 \quad -1]^T$ and $\hat{x}_{-1} = [-1 \quad -1 \quad 0]^T$, and initial control input $u_0 = 1$. Fig. 7 shows the system sate trajectories and estimated system state trajectories, as well as the estimated errors under the approximate optimal control policy (95) and using the approximate optimal observer (90) (see Fig. 8). Compared Fig. 7 with Fig. 4, even though there exist network-induced delays, the good control performance (the same convergence speed and overshoot as those under the case of absence of network-induced delays) can be obtained by using the proposed data-drive learning algorithm by combining prediction control and state observer estimation.

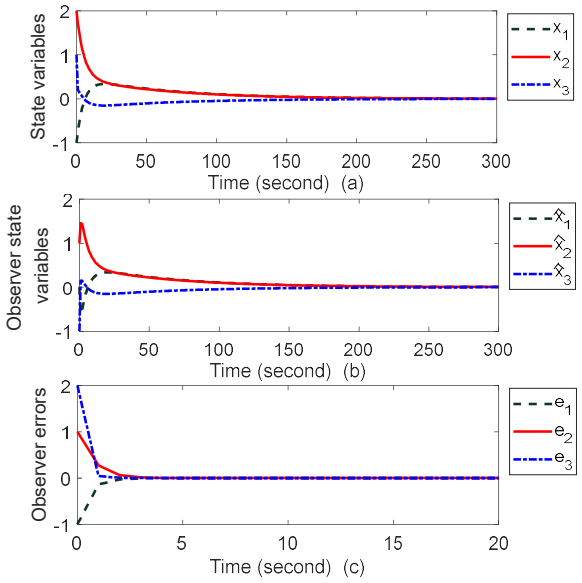

Figure 4: Trajectories of states, estimated states and estimation errors of control system without network-induced delay

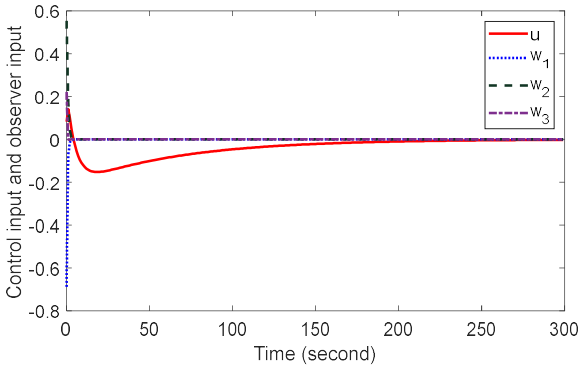

Figure 5. Curves of the optimal control policy and the optimal observer policy

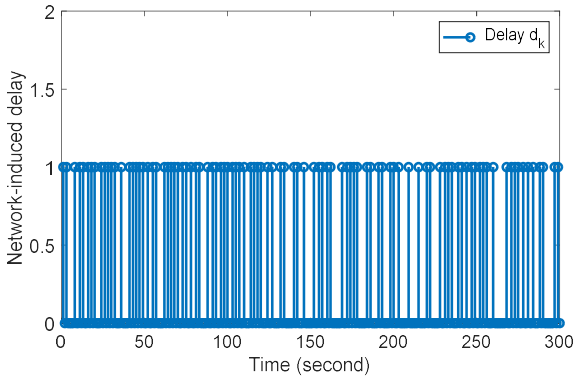

Figure 6: Network-induced delays

6. Conclusions

In this paper, a novel off-policy Q-learning algorithm is developed for handling both network-induced delays and unmeasured state information for discrete-time linear systems with inaccurate system matrices. An optimal control problem is formulated first, wherein a Smith predictor using the delayed estimated states is employed for compensating network-induced delay and satisfying the Separation Principal. Off-policy Q-learning is utilized to respectively find the optimal observer gain and the optimal control policy. Further, a novel off-policy Q-learning algorithm is developed for deriving the predicted observer state based feedback controller by using dynamic programming, RL and appropriate mathematical manipulation, so that the prescribed performance index can be minimized. The proposed algorithm does not require system matrices to be known accurately, and it is implemented using only measured data. Moreover, the good control performance can be ensured by using the predicted observer state-based controller learned by the proposed data-driven off-policy Q-learning algorithm, even though there exist network-induced delays in linear systems subject to unmeasured system states.

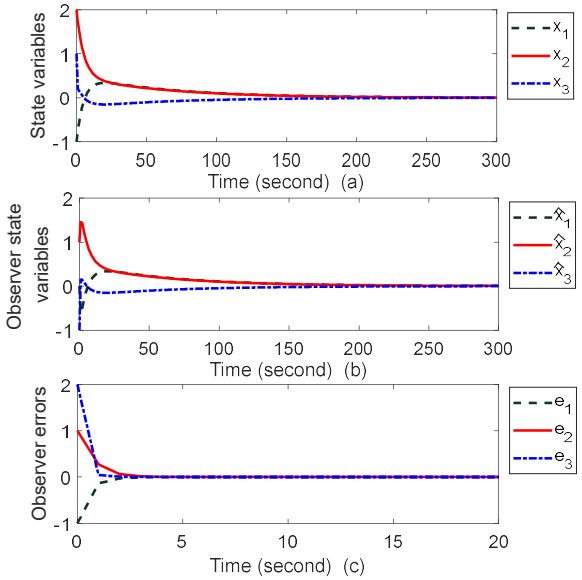

Figure 7: Trajectories of states, estimated states and estimation errors of networked control system

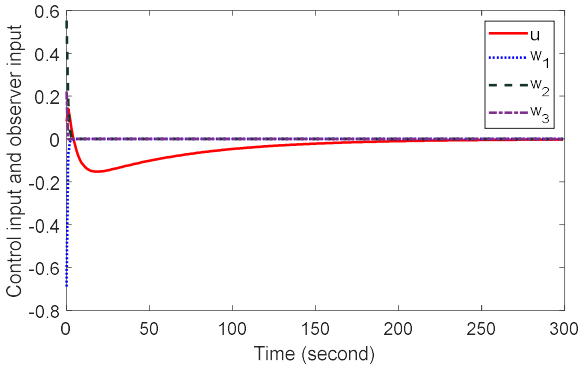

Figure 8: Curves of the optimal control policy and the optimal observer policy under time-varying network-induced delays

Acknowledgement

This work was supported by National Natural Science Foundation of China under Grants 61673280, 61673199, 61590922, 61503257, the Open Project of Key Field Alliance of Liaoning Province under Grant 2019-KF-03-06, the Project of Liaoning Province under Grant LR2017006 and the Project of Liaoning Shihua University under Grant 2018XJJ-005.

References

[1] W. Zhang, M.S. Branicky, and S.M. Phillips, "Stability of networked control systems," *IEEE Control Syst. Mag, vol.* 21, no. 1, pp. 84-89, Feb. 2001.

[2] Y. L. Wang and Q. L. Han, "Network-based modelling and dynamic output feedback control for unmanned marine vehicles," *Automatica*, vol. 91, no. pp. 43-53, May 2018.

[3]. Y. L. Wang, Q. L. Han, M. R. Fei, and C. Peng, "Network-based T-S fuzzy dynamic positioning controller design for unmanned marine vehicles," *IEEE Transactions on Cybernetics*, vol. 48, no. 9, Sep. 2018.

[4] D. Yue, Q. L. Han, C. Peng, "State feedback controller design of networked control systems," *IEEE Trans. Circ. Syst. II: Express Briefs*, vol. 51, no. 11, pp. 640–644, Nov. 2004.

[5] Y. Li , H. Li, X. Ding, and G. Zhao, "Leader-follower consensus of multi-agent systems with time delays over finite fields," *IEEE Transactions on Cybernetics*, to be published, DOI: 10.1109/TCYB.2018.2839892.

[6] S. C. Liu, X. P. Liu, and X. Y. Wang, "Stability analysis and compensation of network-induced delays in communication-based power system control: A survey," *ISA Transactions*, vol. 66, no. 6, pp. 143-153, Jan. 2017.

[7] X. M.Tang, H. C. Qu, P. Wang, and M. Zhao, "Constrained off-line synthesis approach of model predictive control for networked control systems with network-induced delays," *ISA Transactions*, vol. 55, pp. 135-144, Mar. 2015.

[8] D. E. Quevedo, E. I. Silva, and G. C. Goodwin, "Subband coding for networked control systems," *International Journal of Robust and Nonlinear Control*, vol. 19, no. 16, pp. 1817-1836, Nov. 2009.

[9] S. Liu, L. H. Xie, and D. E. Quevedo, "Event-triggered quantized communication-based distributed convex optimization," *IEEE Transactions on Control of Network Systems*, vol. 5, no. 1, Mar. 2018.

[10] J. N. Li, H. Modares, T. Y. Chai, F. L. Lewis, and L. H. Xie, "Off-policy reinforcement learning for synchronization in multi-agent graphical games," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 28, no. 10, pp. 2434-2445, April 2017.

[11] J. N. Li, T. Y. Chai, F. L. Lewis, Z. T. Ding, and Y. Jiang, "Off-policy interleaved Q-learning: optimal control for affine nonlinear discrete-time systems," *IEEE Transactions on Neural Networks and Learning Systems*, Early access, DOI: 10.1109/TNNLS.2018.2861945.

[12] B. Kiumarsi, F. L. Lewis, H. Modaresa, A. Karimpour, and M.B. Naghibi-Sistani, "Reinforcement Q-learning for optimal tracking control of linear discrete-time systems with unknown dynamics," *Automatica*, vol. 50, no. 4, pp. 1167-1175, April 2014.

[13] B. Luo, H. N. Wu, and T. Huang, "Off-policy reinforcement learning for H∞ control design," *IEEE Transactions on Cybernetics*, vol. 45, no. 1, pp. 65-76, Jan. 2015.

[14] Q. L. Wei, D. R. Liu, and G. Shi, "A novel dual iterative Q-learning method for optimal battery management in smart residential environments," *IEEE Transactions on Industrial Electronics*, vol. 62, no. 4, pp. 2509-2518, Apr. 2015.

[15] E. D Sontag, Mathematical Control Theory: Deterministic Finite Dimensional Systems. Second Edition, Springer, New York, ISBN 0-387-98489-5, 315-345,1998.

[16] B. Kiumarsi, F. L. Lewis, M.B. Naghibi-Sistani, and A. Karimpour, "Optimal tracking control of unknown discrete-time linear systems using input-output measured data", *IEEE Transactions on Cybernetics,* vol. 45, no. 12, pp. 2770-2779, Dec. 2015.

[17] T. N. Pham, H. Trinh, and L. V. Hien, "Load frequency control of power systems with electric vehicles and diverse transmission links using distributed functional observers", *IEEE Transactions on Smart Grid*, vol. 7, no. 1, pp. 238-252, 2016.

[18] B. C. Wang, Y. X. Xu, Z. Y. Shen, J. B. Zou, C. Q. Li, and H. Liu, "Current control of grid-connected inverter with LCL filter based on extended-state observer estimations using single sensor and achieving improved robust observation dynamics", *IEEE Transactions on Industrial Electronics*, vol. 64, no. 7, pp. 5428-5439, Feb. 2017.

[19] S. C. Liu, P. X. Liu, and A. E. Saddik, "Modeling and stability analysis of automatic generation control over cognitive radio networks in smart grids", *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 45, no. 2, pp. 223-234, Feb. 2015.

[20] G. Chen and Z. J. Guo, "Distributed secondary and optimal active power sharing control for islanded microgrids with communication delays", *IEEE Transactions on Smart Grid*, vol.10, no. 2, pp. 2002-2014, March 2019.

[21] Y. Jiang, J. L. Fan, T. Y. Chai, J. N. Li, and F. L. Lewis, "Tracking control for linear discrete-time networked control systems with unknown dynamics and dropout," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 10, Oct. 2018.

[22] M. B. Radac, R. E. Precup, and R. C. Roman, "Data-driven model reference control of MIMO vertical tank systems with model-free VRFT and Q-Learning", *ISA Transactions*, vol. 73, pp. 227-238, Feb. 2018.

[23] H. P. Zhang, D. Yue, C. X. Dou, W. Zhao, and X. P. Xie, "Data-driven distributed optimal consensus control for unknown multiagent systems with input-delay," *IEEE Transactions on Cybernetics*, vol. 49, no. 6, pp. 2095-2105, Jun. 2019.

[24] J. N. Li, T. Y. Chai, F. L. Lewis, J. L. Fan, Z. T. Ding, and J. L. Ding, "Off-policy Q-learning: set-point design for optimizing dual-rate rougher flotation operational processes," *IEEE Transactions on Industrial Electronics*, vol. 65, no. 5, pp. 4092-4102, May 2018.

[25] Y. Li, H. Li, and W. Sun, "Event-triggered control for robust set stabilization of logical control networks," Automatica, vol. 95, pp. 556-560, Sep. 2018.

[26] D. Wang, H. B. He, and D. R. Liu, "Adaptive critic nonlinear robust control: A survey," *IEEE Transactions on Cybernetics*, vol. 47, no. 10, pp. 3429-3451, Jul. 2017.

[27] A. A. Tamimi, F. L. Lewis, and M. Abu-Khalaf, "Model-free Q-learning designs for linear discrete-time zero-sum games with application to H-infinity control," *Automatica*, vol. 43, no. 3, pp. 473-481, Mar. 2007.

[28] B. Kiumarsi, F. L. Lewis, and Z. P. Jiang, "H∞ tracking control of completely unknown continuous-time systems via off-policy reinforcement learning," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 26, no. 10, 2550-2565, Oct. 2015.

[29] H. G. Zhang, X. H. Cui, Y. H. Luo, H. Jiang, "Finite-horizon H∞ tracking control for unknown nonlinear systems with saturating actuators", *IEEE Transactions on Neural Networks & Learning Systems*, vol. 29, no. 4, pp. 1200-1213, April 2018.

[30] S. Beale and B. Shafai, "Robust control system design with a proportional integral observer", *Int J Control*, vol. 50, no. 1, pp. 97-111, Jul. 1989.

[31] A. G. Wu, G. R. Duan and A. G. Wu, "Proportional multiple-integral observer design for continuous time descriptor linear systems", *Asian Journal of Control*, vol. 14, no. 2, pp. 476-488, Mar. 2012.