International Journal of Automation and Computing 17(3), June 2020, 417-427
DOI: 10.1007/s11633-020-1229-0

Controller Optimization for Multirate Systems Based on Reinforcement Learning

Zhan Li[1] Sheng-Ri Xue[1] Xing-Hu Yu[1,2] Hui-Jun Gao[1]

[1]Research Institute of Intelligent Control and Systems, Harbin Institute of Technology, Harbin 150001, China

[2]Ningbo Institute of Intelligent Equipment Technology, Harbin Institute of Technology, Ningbo 315200, China

Abstract: The goal of this paper is to design a model-free optimal controller for the multirate system based on reinforcement learning. Sampled-data control systems are widely used in the industrial production process and multirate sampling has attracted much attention in the study of the sampled-data control theory. In this paper, we assume the sampling periods for state variables are different from periods for system inputs. Under this condition, we can obtain an equivalent discrete-time system using the lifting technique. Then, we provide an algorithm to solve the linear quadratic regulator (LQR) control problem of multirate systems with the utilization of matrix substitutions. Based on a reinforcement learning method, we use online policy iteration and off-policy algorithms to optimize the controller for multirate systems. By using the least squares method, we convert the off-policy algorithm into a model-free reinforcement learning algorithm, which only requires the input and output data of the system. Finally, we use an example to illustrate the applicability and efficiency of the model-free algorithm above mentioned.

Keywords: Multirate system, reinforcement learning, policy iteration, optimal control, controller optimization.

1 Introduction

It is well known that nearly all modern control systems are implemented digitally, which results in the research significance of sampled-data systems[1–5]. In industrial process control, there commonly exist conditions where the periods for the practical plant inputs are different. Then traditional and advanced control methods for sampled-data systems will not adapt to such multirate systems. Researchers noticed this problem in the 1950s and Kranc first used the switch decomposition method to solve this problem in [6]. Kalman and Bertram[6], Friedland[7], and Meyer[8] also made contribution to the development of multirate systems. In 1990, the lifting technique was brought out to simplify the multirate problems by converting these systems to the equivalent discrete systems. The topic became active ever since.

Based on the lifting method, standard control methods can be applied to solve the multirate problems. With the development of the advanced control theory, more and more research has been reported so far. In [9], an H_∞ controller is designed for the multirate system with the nest operator method by Chen and Qiu. Sågfors and Toivonen[10] utilized the Riccati equation to address similar H_∞ multirate sampling problems. Also H_2 problems are solved by Qiu and Tan[11] and linear quadratic Gaussian (LQG) problem are addressed by Colaneri and Nicolao[12]. Recently the control difficulties of the networked systems with multirate sampling were solved by Xiao et al.[13], Chen and Qiu[14]. Xue et al.[15] and Zhong et al.[16] utilized different methods to deal with the fault detection problems. Gao et al.[17] designed an output feedback controller for a general multirate system with finite frequency specification. However, all controllers mentioned above are designed according to the system dynamics model. When system structure is unknown or system parameters are uncertain, these controllers will not satisfy our demands. The authors in this paper aim to design a controller that can make use of the input and output data to optimize itself and we denote this kind of controller as a model-free controller.

Reinforcement learning (RL) is an important branch of machine learning. Famous research groups utilize RL to solve artificial intelligence problems and teach robots to play games[18, 19]. Through the interactions with environment, the cognitive agents can obtain the rewards of their actions. With the utilization of the value function, which is calculated by rewards, agents use the RL algorithm to optimize the policy. A similar idea was brought from control theory by Bertsekas and Tsitsiklis in 1995[20], which is adaptive dynamic programming (ADP). And a detailed introduction about ADP can be found in [21]. And in past decades, this method was utilized to deal with output regulation problems[22], switch systems[23], nonlinear systems[24], sliding mode control[25, 26]

Research Article
Manuscript received December 21, 2019; accepted February 21, 2020; published online April 14, 2020
Recommended by Associate Editor Min Wu

and so on[27–29]. Both ADP and RL are studied based on the Bellman equation and researchers combine these two algorithms and apply it for solving control problems. RL algorithms have been used to deal with H_∞ controller design problems[30]. Also the optimal regulation problem was solved by Kamalapurkar et al.[31] The RL algorithm can optimize the policy only with the use of the input and output data, which discards the requirements of system dynamics. Such model-free algorithms were applied for solving discrete systems[32] and heterogeneous systems[33]. Controller design methods based on reinforcement learning have many directions. Madady et al.[34], Li et al.[35] proposed a RL based control structure to train neuro-network controllers for a helicopter. Similar methods can also be applied in unmanned aerial vehicle (UAV)[36]. Some other learning-based control methods can also be used in the servo control systems[37] and traffic systems[38]. In this paper, authors aim to design a model-free optimal controller for multirate systems through similar schemes.

In this paper, a model-free algorithm based on RL is developed to help us to design an optimal controller for multirate systems. We assume that the sampling periods for the state variables are different from the periods of the system inputs. Instead of the lifting method, a different technique was used to convert the multirate systems into an equivalent discrete system. With matrix transformations, we put forward an algorithm to design a linear quadratic regulator (LQR) controller for multirate systems. Later, we propose the definition of the behavior policy and target policy, and then an off-policy algorithm based on RL was provided. With the utilization of the least squares (LS) method[38–40], we reformulate the off-policy algorithm into a model-free RL algorithm, which can help us to optimize the controller in an uncertain environment. Finally, an example is presented to illustrate the applicability and efficiency of the proposed methods.

The paper is organized as follows. A multirate system model with a state feedback controller is provided in Section 2. Section 3 proposes a controller design method and three controller optimization methods. Finally, Section 4 gives an industrial example to illustrate the applicability of the methods above mentioned.

Notation. This paper standardly use notation as follows. $\mathbf{R}^n$ denote the n-dimensional Euclidean space. T and –1 mean matrix transposition and inverse. $\oplus$ stands for the Kronecker product and $vec(A)$ denotes the vectorization of the matrix A.

2 Problem formulation

The multirate system we considered in this paper is a system that has multirate periods for system states and inputs, which means the sampling periods for state $x(t)$ are $p_s h$. Also we assume the periods for the holds of the $u(t)$ are all $p_u h$. Here h denotes a real positive integer referred to the basic period. Then we define this multirate system G with assumptions as

$$\dot{x}(t) = A_c x(t) + B_c u(t). \tag{1}$$

Assumption 1. The periods of samplers for $x(t)$ are all $p_s h$. The periods for the holds of the $u(t)$ are all $p_u h$.

Here $x(t)$ is the state vector and $x \in \mathbf{R}^n$. $u(t)$ is the system input and $u \in \mathbf{R}^m$. A_c and B_c are the system matrices with appropriate dimension. We first convert the multirate system G to the equivalent linear discrete system G_d with the discrete time period h as

$$x(k+1) = Ax(k) + Bu(k) \tag{2}$$

where $A = \mathrm{e}^{A_c h}$, $B = \int_0^h \mathrm{e}^{A_c \tau} \mathrm{d}\tau B_c$.

Researchers traditionally solve the multirate problems through utilization of the lifting method in [10]. According to this traditional method, a dynamic output feedback controller can be designed[18]. It is difficult to directly use reinforcement learning based method for a dynamic output feedback controller. In this paper, we address this difficulty through another lifted method. We can design a state-feedback controller under this lifting technology. Define $N = p_s \times p_u$. In the time period Nh, we have

$$\begin{aligned} x(1) &= Ax(0) + Bu(0), \\ x(2) &= Ax(1) + Bu(1), \\ &\vdots \\ x(N+1) &= Ax(N) + Bu(N). \end{aligned} \tag{3}$$

Here we define the new state vector $\bar{x}$ and new system input $\bar{u}$ in the following lines:

$$\bar{x}(k) = \begin{bmatrix} x(kN - N + s) \\ x(kN - N + 2s) \\ \vdots \\ x(kN) \end{bmatrix}, \ \bar{u}(k) = \begin{bmatrix} u(kN) \\ u(kN + p) \\ \vdots \\ u(kN + N - p) \end{bmatrix}$$

where $p = p_u$, $s = p_s$ and the initial state $\bar{x}(0) = \left[0 \cdots x(0)^{\mathrm{T}}\right]^{\mathrm{T}}$. With the utilization of the above vectors, we have the discrete-time system $\bar{G}$, which is equivalent to G_d.

$$\bar{x}(k+1) = \bar{A}\bar{x}(k) + \bar{B}\bar{u}(k) \tag{4}$$

where

$$\bar{A} = \begin{bmatrix} 0 & \cdots & 0 & A^s \\ 0 & \cdots & 0 & A^{2s} \\ \vdots & \ddots & \vdots & \vdots \\ 0 & \cdots & 0 & A^N \end{bmatrix}, \ \bar{B} = \begin{bmatrix} \bar{B}_{s1} & \bar{B}_{s2} & \cdots & \bar{B}_{1\frac{N}{p}} \\ \bar{B}_{2s1} & \bar{B}_{2s2} & \cdots & \bar{B}_{2s\frac{N}{p}} \\ \vdots & \vdots & \ddots & \vdots \\ \bar{B}_{N1} & \bar{B}_{N2} & \cdots & \bar{B}_{N\frac{N}{p}} \end{bmatrix}$$

$$\bar{B}_{ij} = \begin{cases} \sum_{k=1}^{p} A^{i-jp+k-1}B, & \text{if } i > jp+p \\ 0, & \text{if } i < jp \\ \sum_{k=1+jp-i}^{p} A^{i-jp+k-1}B, & \text{otherwise.} \end{cases}$$

In this article, we design a state feedback controller as

$$\bar{u}(k) = -K\bar{x}(k). \tag{5}$$

From (5) and system $\bar{G}$, one can see that the controller (5) utilizes the data collected in the time interval $[(kN-N+1)h,\ kNh]$. The system inputs obtained by control law are used in $[kNh,\ (kN+N-p)h]$.

Remark 1. In this paper, we use our lifting method to deal with a kind of multirate system, whose sampling periods for inputs or outputs are the same. As for the general multirate sampled-data systems, whose input and output both have different sampling periods, we can find such systems in [10–13, 16, 18]. In these papers, authors utilize the lifting technique to deal with the multirate problems. The lifting method in [16] combines the N state vectors for the new system state vector and system input in such a form:

$$\hat{x}(k) = \begin{bmatrix} x_1(kN) \\ \vdots \\ x_1(kN+N-1) \\ \vdots \\ x_n(kN) \\ \vdots \\ x_n(kN+N-1) \end{bmatrix},\ \hat{u}(k) = \begin{bmatrix} u_1(kN) \\ \vdots \\ u_1(kN+N-1) \\ \vdots \\ u_n(kN) \\ \vdots \\ u_n(kN+N-1) \end{bmatrix}. \tag{6}$$

Obviously, $\hat{x}$ and $\hat{u}$ are different from $\bar{x}$ and $\bar{u}$. The equivalent system with the vectors $\hat{x}$ and $\hat{u}$ will result in the causal constraints problem, which denotes that the control output $u(k+t)$ cannot be controlled by state $x(k+s)$. t and s are positive integers and $t > s$. The controller K(5) in this article provides the system $\bar{G}$ with the data in $[(kN-N+1)h,\ kNh]$ by utilizing the data in $[kNh,\ (kN+N-p)h]$. In other words, the method in this paper used to deal with the multirate difficulties can avoid causal constraints.

Define the cost function of the system G_d with the discounting factor γ as

$$\begin{aligned} J = &\sum_{k=0}^{\infty} \gamma^k (x^{\mathrm{T}}(k)Qx(k) + u^{\mathrm{T}}(k)Ru(k)) = \\ &\lim_{N\to\infty} (\bar{x}^{\mathrm{T}}(N)\bar{Q}\bar{x}(N) + \\ &\sum_{k=0}^{N-1} \gamma^k (\bar{x}^{\mathrm{T}}(k)\bar{Q}\bar{x}(k) + \bar{u}^{\mathrm{T}}(k)\bar{R}\bar{u}(k))). \end{aligned} \tag{7}$$

From the above description, we can obtain that the systems G, G_d and $\bar{G}$ are approximately equivalent. Thus, our purpose in this paper can be described as designing a model-free controller for the multirate system G to minimize the cost function J based on the reinforcement learning method.

3 Main results

In this section, we propose several methods to design optimal controllers for multirate system based on reinforcement learning. Based on the cost function J, the value function given in this article can be described as

$$V(\bar{x}(k)) = \sum_{i=k}^{\infty} \gamma^{i-k} r(\bar{x}(i), \bar{u}(i)) \tag{8}$$

where $r(\bar{x}(i), \bar{u}(i)) = \bar{x}^{\mathrm{T}}(i)\bar{Q}\bar{x}(i) + \bar{u}^{\mathrm{T}}(i)\bar{R}\bar{u}(i)$. Also, we find that the value function (8) can be reformulated as

$$V(\bar{x}(k)) = r(\bar{x}(k), \bar{u}(k)) + \sum_{i=k+1}^{\infty} \gamma^{i-k} r(\bar{x}(i), \bar{u}(i))$$

which yields the Bellman equation

$$V(\bar{x}(k)) = r(\bar{x}(k), \bar{u}(k)) + \gamma V(\bar{x}(k+1)). \tag{9}$$

Also for the system $\bar{G}$, the value function can be described in a quadratic form as

$$V(\bar{x}(k)) = \bar{x}^{\mathrm{T}}(k)P\bar{x}(k).$$

With the above quadratic form, the Bellman equation (9) becomes

$$\bar{x}^{\mathrm{T}}(k)P\bar{x}(k) = r(\bar{x}(k), \bar{u}(k)) + \gamma \bar{x}^{\mathrm{T}}(k+1)P\bar{x}(k+1).$$

We define the Hamiltonian function $H(\bar{x}(k), \bar{u}(k), \lambda(k))$ as

$$H = r(\bar{x}(k), \bar{u}(k)) + \lambda(k)^{\mathrm{T}}(\gamma V(\bar{x}(k+1)) - V(\bar{x}(k))). \tag{10}$$

Then we can obtain the optimal control policy after we solve this Hamiltonian function. Moreover, the optimal feedback matrix for the system $\bar{G}$ is given as

$$K^* = (\bar{R} + \gamma \bar{B}^{\mathrm{T}} P \bar{B})^{-1} \gamma \bar{B}^{\mathrm{T}} P \bar{A} \tag{11}$$

where P is the solution for the algebraic Riccati equation (ARE):

$$\gamma \bar{A}^{\mathrm{T}} P \bar{A} - P - \gamma^2 \bar{A}^{\mathrm{T}} P \bar{B} (\bar{R} + \gamma \bar{B}^{\mathrm{T}} P \bar{B})^{-1} \bar{B}^{\mathrm{T}} P \bar{A} + \bar{Q} = 0. \tag{12}$$

3.1 LQR controller design

When $\gamma = 1$, one can see that the controller designed through (11) and (12) is equivalent to a LQR controller.

From the structure of the matrix $\bar{A}$, we can find that $\bar{A}$ is singular and it is difficult to solve the Riccati equation (12). We give the following matrix transformations

$$\bar{A} = [0 \quad F],\ F^{\mathrm{T}} = [\bar{A}_1^{\mathrm{T}}, \bar{A}_2^{\mathrm{T}}],$$

$$\bar{B}^{\mathrm{T}} = [\bar{B}_1^{\mathrm{T}}, \bar{B}_2^{\mathrm{T}}],\ \bar{A}_2 = A^N,\ N_0 = N-1,$$

$$\bar{B}_1 = \begin{bmatrix} \bar{B}_{s1} & \bar{B}_{s2} & \cdots & \bar{B}_{s\frac{N}{p}} \\ \bar{B}_{2s1} & \bar{B}_{2s2} & \cdots & \bar{B}_{2s\frac{N}{p}} \\ \vdots & \vdots & \ddots & \vdots \\ \bar{B}_{N_0 1} & \bar{B}_{N_0 2} & \cdots & \bar{B}_{N_0\frac{N}{p}} \end{bmatrix},\ \bar{A}_1 = \begin{bmatrix} A^s \\ \vdots \\ A^{N-1} \end{bmatrix},$$

$$\bar{B}_2 = \begin{bmatrix} \bar{B}_{N1} & \bar{B}_{N2} & \cdots & \bar{B}_{N\frac{N}{p}} \end{bmatrix},$$

$$P = \begin{bmatrix} P_{11} & P_{12} \\ P_{21} & P_{22} \end{bmatrix},\ U = P\bar{B}(\bar{R} + \gamma \bar{B}^{\mathrm{T}} P \bar{B})^{-1} \bar{B}^{\mathrm{T}} P$$

where $\bar{B}_{ij}$ can be found in (4). Set $\bar{Q} = \mathrm{diag}\{\mathrm{Q}_1, \mathrm{Q}_2\}$ and $Q_2 \in \mathbf{R}^{N\times N}$, Q_1 has appropriate dimension. Then, with utilizing the above matrix transformations and $\gamma = 1$, the ARE can be converted to the following equations:

$$\begin{bmatrix} 0 \\ F^{\mathrm{T}} \end{bmatrix} P [0 \quad F] - P - \begin{bmatrix} 0 \\ F^{\mathrm{T}} \end{bmatrix} U [0 \quad F] + \begin{bmatrix} Q_1 & 0 \\ 0 & Q_2 \end{bmatrix} = 0. \tag{13}$$

From (13), we have that $P_{12} = P_{21}^{\mathrm{T}} = 0$ and $P_{11} = P_{11}^{\mathrm{T}} = Q_1$. Also one can see that when (13) holds, we have the following equation:

$$F^{\mathrm{T}} P F = P_{22} + F^{\mathrm{T}} U F + Q_2 = \bar{A}_1^{\mathrm{T}} Q_1 \bar{A}_1 + \bar{A}_2^{\mathrm{T}} P_{22} \bar{A}_2. \tag{14}$$

Let

$$\tilde{P} = P_{22},\ \tilde{S} = \bar{A}_1^{\mathrm{T}} Q_1 \bar{B}_1,\ \tilde{A} = \bar{A}_2,\ \ \tilde{B} = \bar{B}_2,$$
$$\tilde{Q} = Q_2 + \bar{A}_1^{\mathrm{T}} Q_1 \bar{A}_1,\ \tilde{R} = \bar{R} + \gamma \bar{B}^{\mathrm{T}} Q_1 \bar{B}.$$

Then we can obtain (15), which can solve for the solution of P_{22}.

$$\tilde{A}^{\mathrm{T}} \tilde{P} \tilde{A} - \tilde{P} - \tilde{L} + \tilde{Q} = 0,$$
$$\tilde{L} = (\tilde{A}^{\tilde{T}} \tilde{P} \tilde{B} + \tilde{S})(\tilde{R} + \tilde{B}^{\mathrm{T}} \tilde{P} \tilde{B})^{-1} (\tilde{B}^{\mathrm{T}} \tilde{P} \tilde{A} + \tilde{S}^{\mathrm{T}}). \tag{15}$$

According to the P_{22} from (15) and $P_{11} = Q_1$, we can obtain the optimal controller with $K = (\bar{R} + \bar{B}^{\mathrm{T}} P \bar{B})^{-1} \bar{B}^{\mathrm{T}} P \bar{A}$.

We conclude the Algorithm 1 to solve the LQR controller design problem for the multirate system G.

The optimal control is to find a control law under the given constraints to maximize or minimize the given cost function. The optimization algorithm is the algorithm that helps agent maximize or minimize the given cost function. We think these two optimization problems have the same target. In this paper, the main target is to obtain a data-driven parameters optimization method for multirate systems. Here Algorithm 1 is the optimal control problem and the optimization algorithm will be given in the following parts of the article.

Algorithm 1. LQR controller for multirate systems

1) Solve the (4) to get the equivalent discrete system $\bar{G}$.

2) Get P_{22} by utilizing the matrix transformations to solve ARE (15).

3) Obtain the LQR controller $K = (\bar{R} + \bar{B}^{\mathrm{T}} P \bar{B})^{-1} \bar{B}^{\mathrm{T}} P \bar{A}$.

Algorithm 2. Online PI algorithm for multirate systems

1) Initialization: Set the iteration number $j = 0$ and start with a stabilizing control law $\bar{u}^0(k) = K_0 \bar{x}(k)$.

2) Solve P^{j+1} by following equation with $P_{11}^{j+1} = Q_1$, $\bar{x}^{\mathrm{T}}(k) P^{j+1} \bar{x}(k) = r(\bar{x}(k), \bar{u}^j(k)) + \gamma \bar{x}^{\mathrm{T}}(k+1) P^{j+1} \bar{x}$.

3) Update the control policy K^{j+1} with $K^{j+1} = (\bar{R} + \gamma \bar{B}^{\mathrm{T}} P^{j+1} \bar{B})^{-1} \gamma \bar{B}^{\mathrm{T}} P^{j+1} \bar{A}$.

4) Stop if $||K^{j+1} - K^j||_2 \leq \epsilon$ for a small positive value ϵ, otherwise set $j = j + 1$ and return to Step 2).

Remark 2. From the structure of the controller, we can find that the structure of $\bar{A}$ results in $K = [0 \ \ K_m]$. It means that the valid part of the controller K is K_m and the system input $\bar{u}(k)$ is decided by $x(kN)$. When using Algorithm 1, we can directly solve (15) to obtain the controller gain K_m. Also, according to the structure of $\bar{u}(k)$, one can see that the controller calculates the control outputs each long period from kNh to $(kN + N - p)h$.

3.2 Model-based PI algorithm

In this subsection, we aim to solve the optimal controller design problem based on reinforcement learning under the condition that the model dynamic is known. The main difficulty is to solve the ARE (12) for matrix P. In Algorithm 1, we utilize the matrix transformations to reformulate a new ARE (15), and we can solve this function directly.

According to [32, 33], another popular algorithm for solving the ARE is the policy iteration (PI) algorithm, an online algorithm. In tradition, the controller updates with the optimization of the matrix P. Based on Section 3.1, we can obtain the matrix P which has such structure, $P_{12} = P_{21}^{\mathrm{T}} = 0$ and $P_{11} = P_{11}^{\mathrm{T}} = Q_1$. Motivated by this condition, the online PI algorithm for multirate sampling systems has been converted into Algorithm 2.

The Algorithm 2 is an online algorithm, which means the policy updates each step according to $\bar{u}^j(k)$. Our purpose is to design a model-free controller, which can update itself with the use of the control output. Therefore, it is inevitable for us to design an off-policy algorithm. Rewriting the system $\bar{G}$ in (4) as

$$\bar{x}(k+1) = \bar{A}_{\mathrm{k}} \bar{x}(k) + \bar{B}(\bar{u}(k) + K^j \bar{x}(k)) \tag{16}$$

where $\bar{A}_{\mathrm{k}} = \bar{A} - \bar{B} K^j$. Here in (16), $\bar{u}(k)$ denotes the

behavior policy which is applied in the practical system and we collect data for the algorithm through this policy. $\bar{u}^j(k) = -K^j\bar{x}(k)$ is the target policy that are updated by the PI algorithm. With $\bar{u}^j(k) = -K^j\bar{x}(k)$, the system in Algorithm 2 is shown as $\bar{x}(k+1) = (\bar{A} - \bar{B}K^j)\bar{x}(k)$. Then the Step 2 in Algorithm 2 is equal to the following equation:

$$\begin{aligned}&\bar{x}^{\mathrm{T}}(k)\bar{Q}\bar{x}(k) + \bar{x}^{\mathrm{T}}(k)(K^j)^{\mathrm{T}}\bar{R}K^j\bar{x}(k) = \\ &\bar{x}^{\mathrm{T}}(k)P^{j+1}\bar{x}(k) - \gamma\bar{x}^{\mathrm{T}}(k)(\bar{A}-\bar{B}K^j)^{\mathrm{T}}P^{j+1}(\bar{A}-\bar{B}K^j)\bar{x}(k).\end{aligned} \tag{17}$$

Then we can obtain the following equation according to the above statements:

$$\begin{aligned}&\bar{x}^{\mathrm{T}}(k)P^{j+1}\bar{x}(k) - \gamma\bar{x}^{\mathrm{T}}(k)\bar{A}^{\mathrm{T}}P^{j+1}\bar{A}\bar{x}(k) = \\ &\quad\bar{x}^{\mathrm{T}}(k)\bar{Q}\bar{x}(k) + \gamma\bar{x}^{\mathrm{T}}(k)(\bar{B}K^j)^{\mathrm{T}}P^{j+1}\bar{B}K^j\bar{x}(k) + \\ &\quad\bar{x}^{\mathrm{T}}(k)(K^j)^{\mathrm{T}}\bar{R}K^j\bar{x}(k) - 2\gamma\bar{x}^{\mathrm{T}}(k)(\bar{B}K^j)^{\mathrm{T}}P^{j+1}\bar{A}\bar{x}(k).\end{aligned} \tag{18}$$

For the off-policy algorithm, the state signals are obtained according to (16). Then we add polynomials into both sides of (18), an equivalent equation is given as

$$\begin{aligned}&\bar{x}^{\mathrm{T}}(k)P^{j+1}\bar{x}(k) - \gamma\bar{x}^{\mathrm{T}}(k+1)P^{j+1}\bar{x}(k+1) = \\ &\quad\bar{x}^{\mathrm{T}}(k)Q\bar{x}(k) + \bar{x}^{\mathrm{T}}(k)(K^j)^{\mathrm{T}}RK^j\bar{x}(k) - \\ &\quad2\gamma(\bar{u}(k) + K^j\bar{x}(k))^{\mathrm{T}}\bar{B}^{\mathrm{T}}P^{j+1}\bar{A}\bar{x}(k) + \\ &\quad\gamma(\bar{u}(k) + K^j\bar{x}(k))^{\mathrm{T}}\bar{B}^{\mathrm{T}}P^{j+1}\bar{B}(\bar{u}(k) - K^j\bar{x}(k)).\end{aligned} \tag{19}$$

It should be noted that $\bar{x}$ in (18) and (19) are different. But (19) is equal to (18). From the above statements, we can conclude the off-policy RL algorithm as Algorithm 3.

Algorithm 3. Off-policy algorithm for multirate systems

1) Initialization: Set the iteration number $j = 0$ and start with a stabilizing control law $\bar{u}(k) = K\bar{x}(k)$.

2) Solve P^{j+1} in (18) with $P_{11}^{j+1} = Q_1$ by using data $\bar{x}(k)$, $\bar{x}(k+1)$, $\bar{u}(k)$, K^j.

3) Update the target policy K^{j+1} with $K^{j+1} = (\bar{R} + \gamma\bar{B}^{\mathrm{T}}P^{j+1}\bar{B})^{-1}\gamma\bar{B}^{\mathrm{T}}P^{j+1}\bar{A}$.

4) Stop if $||K^{j+1} - K^j||_2 \leq \epsilon$ for a small positive value ϵ, otherwise set $j = j + 1$ and return to Step 2).

Remark 3. It is obvious that the difference between Algorithms 2 and 3 is that the policy system utilized is changed each step in Algorithm 2 and fixed in Algorithm 3. In Algorithm 3, the state vector $\bar{x}(k+1)$ is decided by the behavior policy, which means $\bar{x}(k+1) = \bar{A}\bar{x}(k) + \bar{B}\bar{u}(k) \neq \bar{A}\bar{x}(k) + \bar{B}\bar{u}^j(k)$. Algorithm 3 provides a optimization method under such a condition that system dynamic process and optimize process are uncoupled. We can later obtain the model-free algorithm based on this off-policy algorithm.

Remark 4. In the reinforcement learning field, the main difference between model-based and model-free algorithms is whether the algorithm uses the neuro-network to estimate the next state or not. The model-free algorithm directly optimizes the policy network through input and output data. The model-based algorithm will first use the input and output data to optimize a neuro-network, which can correctly predict the next state according to the present. In conclusion, model-based algorithms need the agent to have a physical dynamic plant. Both model-free and model-based algorithms are data-driven methods. However, in the control field, this will be different. There are no specific explanations for model-based and model-free control methods. In this paper, we think model-based methods rely on the system dynamics, including system structures and system parameters. Model-free methods only use input and output data to optimize controller parameters. Both these methods are data-driven but model-based methods use inputs, outputs and system dynamics. In this paper, we aim to propose a method that only uses input and output data to optimize controllers.

3.3 Model-free RL algorithm

The algorithms mentioned above require system dynamics to optimize the controller. In this subsection, we propose a method to design a model-free optimal controller. It is well known that $vec(a^{\mathrm{T}}Wb) = (b^{\mathrm{T}} \oplus a^{\mathrm{T}})vec(W)$. Set that

$$\bar{A}\bar{x}(k) = \begin{bmatrix}0 & \bar{A}_1\\ 0 & \bar{A}_2\end{bmatrix}\begin{bmatrix}\bar{x}_1(k)\\ \bar{x}_2(k)\end{bmatrix}.$$

We can obtain (20) through (18) with $P_{11} = Q_1$.

$$\begin{aligned}&\bar{x}^{\mathrm{T}}(k)\bar{Q}\bar{x}(k) + \bar{x}^{\mathrm{T}}(k)(K^j)^{\mathrm{T}}\bar{R}K^j\bar{x}(k) = \\ &\quad\bar{x}_1^{\mathrm{T}}(k)Q_1\bar{x}_1(k) - \gamma\bar{x}_1^{\mathrm{T}}(k+1)Q_1\bar{x}_1(k+1) + \\ &\quad\bar{x}_2^{\mathrm{T}}(k)P_{22}^{j+1}\bar{x}_2(k) - \gamma\bar{x}_2^{\mathrm{T}}(k+1)P_{22}^{j+1}\bar{x}_2(k+1) + \\ &\quad2\gamma(\bar{u}(k) + K^j\bar{x}(k))^{\mathrm{T}}(\bar{B}_1^{\mathrm{T}}Q_1\bar{A}_1 + \bar{B}_2P_{22}^{j+1}\bar{A}_2)\bar{x}_2(k) + \\ &\quad\gamma(\bar{u}(k) + K^j\bar{x}(k))^{\mathrm{T}}\bar{B}^{\mathrm{T}}P^{j+1}\bar{B}(\bar{u}(k) - K^j\bar{x}(k)).\end{aligned} \tag{20}$$

Define $\bar{r}(\bar{x}(k))$ as

$$\begin{aligned}&\bar{r}(\bar{x}(k)) = \bar{x}^{\mathrm{T}}(k)\bar{Q}\bar{x}(k) + \bar{x}^{\mathrm{T}}(k)(K^j)^{\mathrm{T}}\bar{R}K^j\bar{x}(k) - \\ &\bar{x}_1^{\mathrm{T}}(k)Q_1\bar{x}_1(k) + \gamma\bar{x}_1^{\mathrm{T}}(k+1)Q_1\bar{x}_1(k+1).\end{aligned}$$

With the utilization of the Kronecker product, (16) can be rewritten as

$$\begin{aligned}&\bar{r}(\bar{x}(k)) = \\ &\quad(\bar{x}_2^{\mathrm{T}}(k) \oplus \bar{x}_2^{\mathrm{T}}(k) - \bar{x}_2^{\mathrm{T}}(k+1) \oplus \bar{x}_2^{\mathrm{T}}(k+1))vec(P_{22}^{j+1}) + \\ &\quad2\gamma(\bar{x}_2^{\mathrm{T}}(k) \oplus (\bar{u}(k) + K^j\bar{x}(k))^{\mathrm{T}})vec(\bar{B}_1^{\mathrm{T}}Q_1\bar{A}_1) + \\ &\quad2\gamma(\bar{x}_2^{\mathrm{T}}(k) \oplus (\bar{u}(k) + K^j\bar{x}(k))^{\mathrm{T}})vec(\bar{B}_2^{\mathrm{T}}P_{22}^{j+1}\bar{A}_2) + \\ &\quad\gamma((\bar{u}(k) - K^j\bar{x}(k))^{\mathrm{T}} \oplus (\bar{u}(k) + K^j\bar{x}(k))^{\mathrm{T}})vec(\bar{B}^{\mathrm{T}}P^{j+1}\bar{B}).\end{aligned} \tag{21}$$

From the Bellman equation (20), one can see that this equation has $n^2+N^2m^2/p^2+Nmn/p+(N-1)(N-p)mn/p$ unknown parameters. Therefore, at least $n^2+N^2m^2/p^2+Nmn/p+(N-1)(N-p)mn/p$ data sets are required to update the control policy. It also means that in each iteration, we will collect s groups of data to calculate the policy. We set a positive integer $s \geq n^2+N^2m^2/p^2+Nmn/p+(N-1)(N-p)mn/p$, and it also means that in each iteration, we will collect s groups of data to calculate the policy. Then define the parameter matrices as

$$\Phi^j = \begin{bmatrix} \bar{r}(\bar{x}(k))^{\mathrm{T}} & \bar{r}(\bar{x}(k+1))^{\mathrm{T}} & \cdots & \bar{r}(\bar{x}(k+s-1))^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}}$$
$$\Psi^j = \begin{bmatrix} M_{(xx)1} & M_{(xu)1} & M_{(uu)1} \\ M_{(xx)2} & M_{(xu)2} & M_{(uu)2} \\ \vdots & \vdots & \vdots \\ M_{(xx)s} & M_{(xu)s} & M_{(uu)s} \end{bmatrix} \tag{22}$$

where

$$\begin{aligned} M_{(xx)i} &= \bar{x}_2^{\mathrm{T}}(k+i-1) \oplus \bar{x}_2^{\mathrm{T}}(k+i-1) - \\ &\quad \bar{x}_2^{\mathrm{T}}(k+i) \oplus \bar{x}_2^{\mathrm{T}}(k+i) \\ M_{(xu)i} &= 2\gamma(\bar{x}_2^{\mathrm{T}}(k+i-1) \oplus (\bar{u}(k+i-1) + \\ &\quad K^j \bar{x}(k+i-1))^{\mathrm{T}}) \\ M_{(uu)i} &= \gamma((\bar{u}(k+i-1) - K^j \bar{x}(k+i-1))^{\mathrm{T}} \oplus \\ &\quad (\bar{u}(k+i-1) + K^j \bar{x}(k+i-1))^{\mathrm{T}}). \end{aligned}$$

Define the unknown variables as

$$W_1^{j+1} = P_{22}^{j+1},\ W_2^{j+1} = \bar{B}_1^{\mathrm{T}} Q_1 \bar{A}_1 + \bar{B}_2^{\mathrm{T}} P_{22}^{j+1} \bar{A}_2,$$
$$W_3^{j+1} = \bar{B}^{\mathrm{T}} P^{j+1} \bar{B}. \tag{23}$$

With the utilization of (20)–(22), we can obtain that

$$\Psi^j \begin{bmatrix} vec(W_1^{j+1})^{\mathrm{T}} & vec(W_1^{j+1})^{\mathrm{T}} & vec(W_1^{j+1})^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}} = \Phi^j. \tag{24}$$

The above equation (20) can be solved by the LS method as

$$\begin{bmatrix} vec(W_1^{j+1})^{\mathrm{T}} & vec(W_1^{j+1})^{\mathrm{T}} & vec(W_1^{j+1})^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}} = ((\Psi^j)^{\mathrm{T}}\Psi^j)^{-1}(\Psi^j)^{\mathrm{T}}\Phi^j. \tag{25}$$

With the solution for W_1^{j+1}, W_2^{j+1} and W_3^{j+1}, we can have the controller gain as

$$K = (R + W_3^{j+1})^{-1} \begin{bmatrix} 0 & W_2^{j+1} \end{bmatrix}. \tag{26}$$

Conclude above statements, we can have the model-free algorithm as Algorithm 4.

Algorithm 4. Model-free controller optimization algorithm for multirate systems

1) Initialization: Set the iteration number $j=0$ and start with a stabilizing control law $\bar{u}(k) = -K\bar{x}(k) + e(k)$, where $e(k)$ is the probing noise. Set $j=0$.

2) Run the system with controller s step to collect inputs and outputs data.

3) Reformulate data to Φ^j and Ψ^j according to (22).

4) With the LS method and (24), we can obtain the matrices W_1^{j+1}, W_2^{j+1}, W_3^{j+1}.

5) Update the control policy K^{j+1} with $K^{j+1} = (R + W_3^{j+1})^{-1} \begin{bmatrix} 0 & W_2^{j+1} \end{bmatrix}$.

6) Stop if $||K^{j+1} - K^j||_2 \leq \epsilon$ for a small positive value ϵ, otherwise set $j = j+1$ and return to 2).

Remark 5. The main technology we used in this subsection is the LS method and an important point when using LS is persistent excitation condition (PE). In the reinforcement learning such as deep Q-network (DQN) and deep deterministic policy gradient (DDPG), the researchers always added noise signals in the learning process to ensure the agents explore more information about the environment, and respectively PE in the policy iteration method can guarantee the sufficient exploration of the state space. It should be noted that when the state converges to the desired value, PE will not be satisfied. In [32, 33, 41, 42], authors always utilized probing noise as PE, which consists of sinusoids of varying frequencies to ensure PE qualitatively. The amplitude for the probing noise will affect the algorithm results. The algorithm will not converge if the amplitude is too large and the probing noise will be useless if the amplitude is too little. We usually decide the parameters according to our experience in the simulation.

Remark 6. In past decades, research about multirate system paid more attention in the traditional control theory field. H_∞ problems, H_2 problems, time-delay problems and LQG problems have been solved in the existing related literatures. In recent years, papers have reported slide mode controller design, nonlinear multirate systems and multirate systems under switch condition. The above algorithms are all model-based and in this paper we propose a model-free LQR controller for multirate systems. When we are not sure about system structure or system parameters, our algorithm can efficiently help users design a LQR controller only with input and output data. Also when the parameters of the system are uncertain, the controller designed by our algorithm will have better performance.

The main results of this paper are 4 algorithms for multirate systems. Algorithms 2 and 3 are preliminary for Algorithm 4. So the main contribution can be concluded as two control system schemes. One is optimal controller design for the new multirate system. The other is model-free controller optimization for the new multirate system. Consider the multirate system described as (1). Its equivalent discrete system is (4) and its state-feedback con-

troller is in the form of (5). Then the closed-loop equivalent multirate control system can be described as

$$\bar{x}(k+1) = (\bar{A} - \bar{B}K)\bar{x}(k). \tag{27}$$

For the optimal controller design of the new multirate system, system dynamics and system parameters are known. We can use Algorithm 1 to obtain an optimal controller with $K = (\bar{R} + \bar{B}^{\mathrm{T}} P \bar{B})^{-1} \bar{B}^{\mathrm{T}} P \bar{A}$. For the matrix P, we can obtain $P_{11} = Q_1$ and P_{22} from the ARE (15).

For the model-free controller optimization for new multirate systems, system dynamics and system parameters are unknown. We first initialize with a stabilized controller and then use Algorithm 4 to optimize controller parameters. Run the closed-loop system, collect data and reformulate data according to (22)–(24). Finally, use (25) to update the controller.

The main contribution of this paper is to propose a model-free controller optimization for a class of multirate systems through a new lifting technique. For the multirate systems, the optimal controller design method is complicated[11] and there is no data-driven method for multirate systems. So the results of this article can make up for the deficiency of the multirate systems. Also, for the controller optimization, this paper presents a realization method and we think our algorithms can improve the development of the data-driven method, controller optimization for multirate systems.

4 Simulation results

In this section, we provide a continuous-stirred tank reactor (CSTR) to prove the applicability and efficiency of the proposed algorithms. And the structure of the industrial CSTR is given in Fig. 1. The main character parameters of CSTR are reaction temperature T and cooling medium temperature T_c. There are two main chemical species A and B. The input of CSTR is pure A with its concentration described as C_{Ai}, the output is the mixture of A and B with their concentration C_A.

For this model, we define state vector and system input as follows:

$$x = \begin{bmatrix} C_A \\ T \end{bmatrix}, \ u = \begin{bmatrix} T_c \\ C_{Ai} \end{bmatrix}.$$

Based on [16], we can convert the sampled-data system into an equivalent discrete system with frequency 2 Hz:

$$x(k+1) = \begin{bmatrix} 0.9719 & -0.0013 \\ -0.034 & 0.8628 \end{bmatrix} x(k) + \begin{bmatrix} -0.0839 & 0.0232 \\ 0.0761 & 0.4144 \end{bmatrix} u(k). \tag{28}$$

It is assumed in this example that the sampling periods for C_A and T are 2 s, the periods for system inputs are 3 s. With the technique of this paper, we can obtain the equivalent system $\bar{G}$ with the system matrices $\bar{A}$ and $\bar{B}$, which are 6×6 and 6×4 matrices.

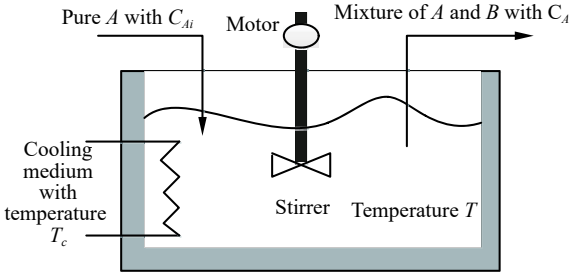


Fig. 1 Continuous-stirred tank reactor model

With the matrix transformations, we can obtain the matrices $\tilde{A}, \tilde{B}, \tilde{P}, \tilde{Q}, \tilde{R}$ by (14). Set $Q = \mathrm{diag}\{1, 1\}$, $R = \mathrm{diag}\{0.1, 0.1\}$, and through Algorithm 1 and the toolbox of Matlab, we can get the optimal controller K_m as

$$K_m = \begin{bmatrix} -2.181\,5 & 0.165\,2 \\ 0.345\,5 & 0.706\,4 \\ -0.831\,4 & -0.002\,7 \\ 0.128\,0 & -0.109\,9 \end{bmatrix}. \tag{29}$$

The target of our paper is to propose a model-free algorithm as Algorithm 4. Algorithms 2 and 3 are the important conditions for Algorithm 4 and we first prove the accuracy and efficiency of Algorithm 2, we set the discounted factor $\gamma = 0.95$ and initial controller K_m^0 as

$$K_m^0 = \begin{bmatrix} -1 & 0 & -2 & 0.4 \\ 0 & 1 & 0.1 & 1 \end{bmatrix}^{\mathrm{T}}.$$

When each iteration system runs, the controller updates itself according to Algorithm 2. With using Algorithm 2, we can obtain Fig. 2. From that one can see that after 5 iterations, the 2-norm of the error between K^j and K^* nearly converges to 0. And the final result of Algorithm 2 is K_m^* as follows:

$$K_m = \begin{bmatrix} -2.182\,3 & 0.171\,1 \\ 0.343\,5 & 0.712\,1 \\ -0.835\,1 & -0.002\,6 \\ 0.131\,1 & -0.110\,1 \end{bmatrix}. \tag{30}$$

From (28), we can get that the controller results we obtained from Algorithm 2 nearly equal to the results we get from Algorithm 1, which denotes that Algorithm 2 is also a useful way to optimize controllers. Except using Algorithm 1 to find optimal controller, we can also use Algorithm 2. Then we test the controller (27), the initial controller of Algorithm 2 and the controller after 3 iterations of Algorithm 2, and we can obtain Fig. 3. Algorithm 3 is another form of Algorithm 2, so we will not test Algorithm 2 here.

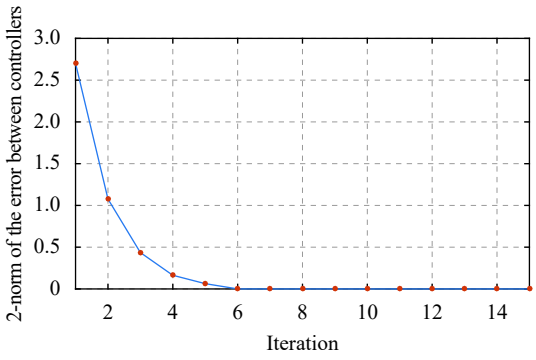


Fig. 2 Convergence of the controller in Algorithm 2

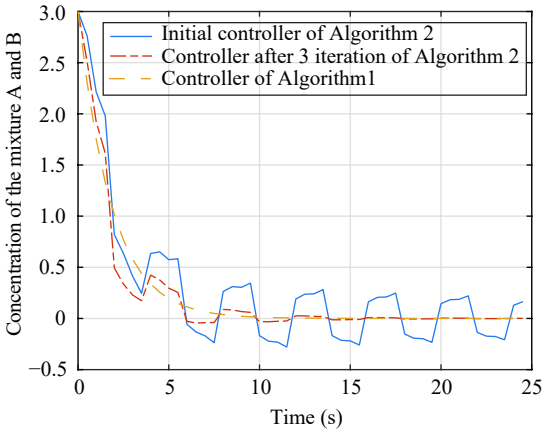


Fig. 3 System states in the Online PI Algorithm

To test the model-free Algorithm 4, we assume that the system matrices we get are different from the practical system and thus suppose

$$x(k+1) = \begin{bmatrix} 0.481\,9 & -0.001\,3 \\ -0.034 & 0.562\,8 \end{bmatrix} x(k) + \begin{bmatrix} -0.139 & 0.023 \\ 0.096 & 0.214 \end{bmatrix} u(k). \tag{31}$$

Here the example means that there is difference between the actual system and system parameters we get. This also denotes that if the system has large uncertainties, we cannot directly use Algorithms 1 and 2 and we will prove the efficiency of Algorithm 4. The actual system is (28), and the incorrect information we get is (31).

First we use Algorithm 1 to obtain the LQR controller of the system we know as follows:

$$K_m = \begin{bmatrix} -0.341\,2 & 0.161\,2 \\ 0.059\,2 & 0.450\,2 \\ -0.012\,1 & -0.016\,0 \\ 0.011\,2 & -0.016\,0 \end{bmatrix}. \tag{32}$$

Set the probing noise $e(k) = \sin(0.9\mathrm{k}) + \sin(1.009\mathrm{k}) + \cos(100\mathrm{k})$ and run system (26) with Algorithm 4, similar we have the 2-norm error between the controller of Algorithm 4 and the controller of Algorithm 1 in Fig. 4.

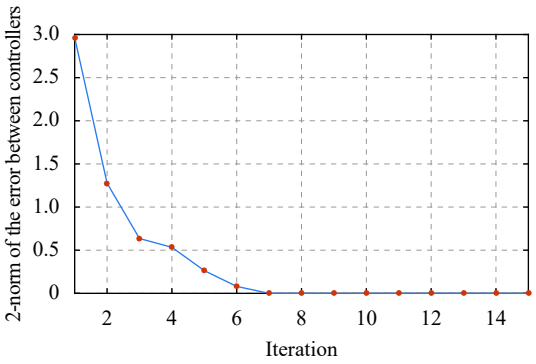


Fig. 4 Convergence of the controller in Algorithm 4

And we can obtain the final controller as follows, which nearly equals to (27).

$$K_m = \begin{bmatrix} -2.181\,2 & 0.170\,1 \\ 0.341\,5 & 0.713\,2 \\ -0.834\,2 & -0.002\,9 \\ 0.130\,9 & -0.109\,7 \end{bmatrix}. \tag{33}$$

Also when running Algorithm 4, we have the system states in Fig. 5. Different from Fig. 3 and Algorithm 2, Algorithm 4 is the off-line algorithm, which means the controller can be optimized when the system is running. In Fig. 5, Controller 1 response means that system runs with the initial controller (31) and Controller 3 denotes the system running with the final controller (32). Controller 2 response means the system runs with Algorithm 4. Here probing noise ends after 20 s.

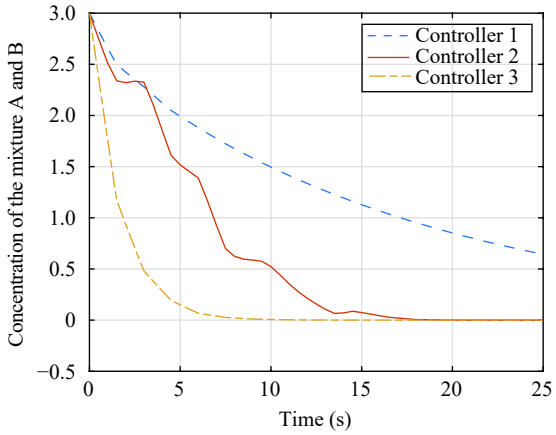


Fig. 5 System states in Algorithm 4

The above simulation results can illustrate that we can only use the input and output data to design an optimal controller with an initial stabilizing control policy, appropriate probing noise and Algorithm 4.

Remark 7. In the practical plants, the system model we obtain is almost different from the actual one due to the uncertainties. And in our opinion, we can first design the LQR controller with the mathematical system model and then utilize the model-free Algorithm 4 to get the

true optimal controller. This optimal controller can actually satisfy our demand for system performance.

5 Conclusions

In this paper, an optimal controller design problem for multirate systems with unknown dynamics is presented. A novel lifting technique is utilized to deal with the multirate sampling problems and provide an equivalent discrete-time system for authors to design algorithms. We then use the Q-learning idea to design a model based off-policy to optimize an algorithm for multirate systems. The LS method can be applied to convert the off-policy algorithm to the model-free algorithm and the utilization of the probing noise is necessary. Finally, a CSTR example is presented to illustrate the applicability of the model-free RL based algorithm.

Future research efforts will focus on the controller design with multiple targets. Due to the limitation of the policy iteration methods, we aim to use the policy gradient methods to design better controllers.

Acknowledgements

This work was supported by National Key R&D Program of China (No. 2018YFB1308404).

References

[1] P. Shi. Filtering on sampled-data systems with parametric uncertainty. *IEEE Transactions on Automatic Control*, vol. 43, no. 7, pp. 1022–1027, 1998. DOI: 10.1109/9.701119.

[2] X. J. Han, Y. C. Ma. Sampled-data robust H_∞ control for T-S fuzzy time-delay systems with state quantization. *International Journal of Control, Automation and Systems*, vol. 17, no. 1, pp. 46–56, 2019. DOI: 10.1007/s12555-018-0279-3.

[3] K. Abidi, Y. Yildiz, A. Annaswamy. Control of uncertain sampled-data systems: An adaptive posicast control approach. *IEEE Transactions on Automatic Control*, vol. 62, no. 5, pp. 2597–2602, 2017. DOI: 10.1109/TAC.2016.2600627.

[4] T. Nguyen-Van. An observer based sampled-data control for class of scalar nonlinear systems using continualized discretization method. *International Journal of Control, Automation and Systems*, vol. 16, no. 2, pp. 709–716, 2018. DOI: 10.1007/s12555-016-0739-6.

[5] R. J. Liu, J. F. Wu, D. Wang. Sampled-data fuzzy control of two-wheel inverted pendulums based on passivity theory. *International Journal of Control, Automation and Systems*, vol. 16, no. 5, pp. 2538–2648, 2018. DOI: 10.1007/s12555-018-0063-4.

[6] R. E. Kalman, J. E. Bertram. A unified approach to the theory of sampling systems. *Journal of the Franklin Institute*, vol. 267, no. 5, pp. 405–436, 1959. DOI: 10.1016/0016-0032(59)90093-6.

[7] B. Friedland. Sampled-data control systems containing periodically varying members. In *Proceedings of the 1st IFAC World Conference*, Moscow, Russia, pp. 361–367, 1961. DOI: 10.1016/s1474-6670(17)70078-X.

[8] D. G. Meyer. A new class of shift-varying operators, their shift-invariant equivalents, and multirate digital systems. *IEEE Transactions on Automatic Control*, vol. 35, no. 4, pp. 429–433, 1990. DOI: 10.1109/9.52295.

[9] T. W. Chen, L. Qiu. H_∞ design of general multirate sampled-data control systems. *Automatica*, vol. 30, no. 7, pp. 1139–1152, 1994. DOI: 10.1016/0005-1098(94)90210-0.

[10] M. F. Sågfors, H. T. Toivonen, B. Lennartson. H_∞ control of multirate sampled-data systems: A state-space approach. *Automatica*, vol. 34, no. 4, pp. 415–428, 1998. DOI: 10.1016/S0005-1098(97)00236-7.

[11] L. Qiu, K. Tan. Direct state space solution of multirate sampled-data H_2 optimal control. *Automatica*, vol. 34, no. 11, pp. 1431–1437, 1998. DOI: 10.1016/S0005-1098(98)00080-6.

[12] P. Colaneri, G. D. Nicolao. Multirate LQG control of continuous-time stochastic systems. *Automatica*, vol. 31, no. 4, pp. 591–595, 1995. DOI: 10.1016/0005-1098(95)98488-R.

[13] N. Xiao, L. H. Xie, L. Qiu. Feedback stabilization of discrete-time networked systems over fading channels. *IEEE Transactions on Automatic Control*, vol. 57, no. 9, pp. 2167–2189, 2012. DOI: 10.1109/TAC.2012.2183450.

[14] W. Chen, L. Qiu. Stabilization of networked control systems with multirate sampling. *Automatica*, vol. 49, no. 6, pp. 1528–1537, 2013. DOI: 10.1016/j.automatica.2013.02.010.

[15] S. R. Xue, X. B. Yang, Z. Li, H. J. Gao. An approach to fault detection for multirate sampled-data systems with frequency specifications. *IEEE Transactions on Systems, man, and cybernetics: Systems*, vol. 48, no. 7, pp. 1155–1165, 2018. DOI: 10.1109/TSMC.2016.2645797.

[16] M. Y. Zhong, H. Ye, S. X. Ding, G. Z. Wang. Observer-based fast rate fault detection for a class of multirate sampled-data systems. *IEEE Transactions on Automatic control*, vol. 52, no. 3, pp. 520–525, 2007. DOI: 10.1109/TAC.2006.890488.

[17] H. J. Gao, S. R. Xue, S. Yin, J. B. Qiu, C. H. Wang. Output feedback control of multirate sampled-data systems with frequency specifications. *IEEE Transactions on Control Systems Technology*, vol. 25, no. 5, pp. 1599–1608, 2017. DOI: 10.1109/TCST.2016.2616379.

[18] X. X. Guo, S. Singh, H. Lee, R. Lewis, X. S. Wang. Deep learning for real-time Atari game play using offline monte-carlo tree search planning. In *Proceedings of the 27th International Conference on Neural Information Processing Systems*, ACM, Montreal, Canada, pp. 3338-3346, 2014.

[19] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. Van Den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, S. Dieleman, D. Grewe, J. Nham, N. Kalchbrenner, I. Sutskever, T. Lillicrap, M. Leach, K. Kavukcuoglu, T. Graepel, D. Hassabis. Mastering the game of go with deep neural networks and tree search. *Nature*, vol. 529, no. 7587, pp. 484–489, 2016. DOI: 10.1038/nature16961.

[20] D. P. Bertsekas, J. N. Tsitsiklis. Neuro-dynamic program-

ming: An overview. In *Proceedings of the 34th IEEE Conference on Decision and Control*, IEEE, New Orleans, USA, pp. 560–564, 1995. DOI: 10.1109/CDC.1995.478953.

[21] F. Y. Wang, H. G. Zhang, D. R. Liu. Adaptive dynamic programming: An introduction. *IEEE Computational Intelligence Magazine*, vol. 4, no. 2, pp. 39–47, 2009. DOI: 10.1109/MCI.2009.932261.

[22] W. N. Gao, Z. P. Jiang. Adaptive dynamic programming and adaptive optimal output regulation of linear systems. *IEEE Transactions on Automatic Control*, vol. 61, no. 12, pp. 4164–4169, 2016. DOI: 10.1109/TAC.2016.2548662.

[23] W. J. Lu, P. P. Zhu, S. Ferrari. A hybrid-adaptive dynamic programming approach for the model-free control of nonlinear switched systems. *IEEE Transactions on Automatic Control*, vol. 61, no. 10, pp. 3203–3208, 2016. DOI: 10.1109/TAC.2015.2509421.

[24] Y. Yang, J. M. Lee. A switching robust model predictive control approach for nonlinear systems. *Journal of Process Control*, vol. 23, no. 6, pp. 852–860, 2013. DOI: 10.1016/j.jprocont.2013.03.011.

[25] B. Luo, H. N. Wu, T. W. Huang. Off-policy reinforcement learning for H_∞ control design. *IEEE Transactions on Cybernetics*, vol. 45, no. 1, pp. 65–76, 2015. DOI: 10.1109/TCYB.2014.2319577.

[26] H. J. Yang, M. Tan. Sliding mode control for flexible-link manipulators based on adaptive neural networks. *International Journal of Automation and Computing*, vol. 15, no. 2, pp. 239–248, 2018. DOI: 10.1007/s11633-018-1122-2.

[27] M. S. Tong, W. Y. Lin, X. Huo, Z. S. Jin, C. Z. Miao. A model-free fuzzy adaptive trajectory tracking control algorithm based on dynamic surface control. *International Journal of Advanced Robotic Systems*, vol. 17, no. 1, pp. 17–29, 2020. DOI: 10.1177/1729881419894417.

[28] I. Zaidi, M. Chtourou, M. Djemel. Robust neural control of discrete time uncertain nonlinear systems using sliding mode backpropagation training algorithm. *International Journal of Automation and Computing*, vol. 16, no. 2, pp. 213–225, 2019. DOI: 10.1007/s11633-017-1062-2.

[29] M. Zhu, J. N. Bian, W. M. Wu. A novel collaborative scheme of simulation and model checking for system properties verification. *Computers in Industry*, vol. 57, no. 8-9, pp. 752–757, 2006. DOI: 10.1016/j.compind.2006.04.006.

[30] Y. H. Zhu, D. B. Zhao, H. B. He, J. H. Ji. Event-triggered optimal control for partially unknown constrained-input systems via adaptive dynamic programming. *IEEE Transactions on Industrial Electronics*, vol. 64, no. 5, pp. 4101–4109, 2017. DOI: 10.1109/TIE.2016.2597763.

[31] R. Kamalapurkar, P. Walters, W. E. Dixon. Model-based reinforcement learning for approximate optimal regulation. *Automatica*, vol. 64, pp. 94–104, 2016. DOI: 10.1016/j.automatica.2015.10.039.

[32] B. Kiumarsi, F. L. Lewis, H. Modares, A. Karimpour, M. B. Naghibi-Sistani. Reinforcement Q-learning for optimal tracking control of linear discrete-time systems with unknown dynamics. *Automatica*, vol. 50, pp. 1167–1175, 2014. DOI: 10.1016/j.automatica.2014.02.015.

[33] H. Modares, S. P. Nageshrao, G. A. Delgado Lopes, R. Babuska, F. L. Lewis. Optimal model-free output synchronization of heterogeneous systems using off-policy reinforcement learning. *Automatica*, vol. 71, pp. 334–341, 2016. DOI: 10.1016/j.automatica.2016.05.017.

[34] A. Madady, H. R. Reza-Alikhani, S. Zamiri. Optimal N-parametric type iterative learning control. *International Journal of Control, Automation and Systems*, vol. 16, no. 5, pp. 2187–2202, 2018. DOI: 10.1007/s12555-017-0259-z.

[35] Z. Li, S. R. Xue, W. Y. Lin, M. S. Tong. Training a robust reinforcement learning controller for the uncertain system based on policy gradient method. *Neurocomputing*, vol. 316, pp. 313–321, 2018. DOI: 10.1016/j.neucom.2018.08.007.

[36] S. R. Xue, Z. Li, L. Yang. Training a model-free reinforcement learning controller for a 3-degree-of-freedom helicopter under multiple constraints. *Measurement and Control*, vol. 52, no. 7-8, pp. 844–854, 2019. DOI: 10.1177/0020294019847711.

[37] S. Preitl, R. E. Precup, Z. Preitl, S. Vaivoda, S. Kilyeni, J. K. Tar. Iterative feedback and learning control. *Servo systems applications. IFAC Proceedings Volumes*, vol. 40, no. 8, pp. 16–27, 2007. DOI: 10.3182/20070709-3-RO-4910.00004.

[38] R. P. A. Gil, Z. C. Johanyak, T. Kovacs. Surrogate model based optimization of traffic lights cycles and green period ratios using microscopic simulation and fuzzy rule interpolation. *International Journal of Artificial Intelligence*, vol. 16, no. 1, pp. 20–40, 2018.

[39] F. L. Lewis, D. Vrabie, K. G. Vamvoudakis. Reinforcement learning and feedback control: Using natural decision methods to design optimal adaptive controllers. *IEEE Control Systems Magazine*, vol. 32, no. 6, pp. 76–105, 2012. DOI: 10.1109/MCS.2012.2214134.

[40] J. X. Yu, H. Dang, L. M. Wang. Fuzzy iterative learning control-based design of fault tolerant guaranteed cost controller for nonlinear batch processes. *International Journal of Control, Automation and Systems*, vol. 16, no. 5, pp. 2518–2527, 2018. DOI: 10.1007/s12555-017-0614-0.

[41] H. Modares, F. L. Lewis, Z. P. Jiang. Optimal output-feedback control of unknown continuous-time linear systems using off-policy reinforcement learning. *IEEE Transactions on Cybernetics*, vol. 46, no. 11, pp. 2401–2410, 2016. DOI: 10.1109/TCYB.2015.2477810.

[42] B. Hu, J. C. Wang. Deep learning based hand gesture recognition and UAV flight controls. *International Journal of Automation and Computing*, vol. 17, no. 1, pp. 17–29, 2020. DOI: 10.1007/s11633-019-1194-7.

Zhan Li received the Ph. D. degree in control science and engineering from Harbin Institute of Technology, Harbin, China in 2015. He is currently an associate professor with Research Institute of Intelligent Control and Systems, School of Astronautics, Harbin Institute of Technology, China.

His research interests include motion control, industrial robot control, robust control of small unmanned aerial vehicles (UAVs), and cooperative control of multivehicle systems.

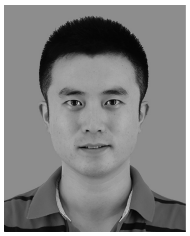
E-mail: zhanli@hit.edu.cn

ORCID iD: 0000-0002-7601-4332

Sheng-Ri Xue received the B. Sc. degree in automation engineering from Harbin Institute of Technology, China in 2015, where he is currently pursuing the Ph. D. degree with the Research Institute of Intelligent Control and Systems.

His research interests include H-infinity control, controller optimization, reinforcement learning, and their applications to sampled-data control systems design.

E-mail: srxue2015@126.com



Xing-Hu Yu received the M. M. degree in osteopathic medicine from Jinzhou Medical University, China, in 2016. He is currently a Ph. D. degree candidate in control science and engineering from Harbin Institute of Technology, China.

His research interests include intelligent control and biomedical image processing.

E-mail: yuxinghu1012@126.com

Hui-Jun Gao received the Ph. D. degree in control science and engineering from Harbin Institute of Technology, China in 2005. From 2005 to 2007, he carried out his postdoctoral research with Department of Electrical and Computer Engineering, University of Alberta, Canada. Since 2004, he has been with Harbin Institute of Technology, where he is currently a full professor, the Director of Inter-discipline Science Research Center, and the Director of the Research Institute of Intelligent Control and Systems. He is an IEEE Industrial Electronics Society Administration Committee Member, and a council member of IFAC. He is the Co-Editor-in-Chief for *IEEE Transactions on Industrial Electronics*, and an Associate Editor for *Automatica*, *IEEE Transactions on Control Systems Technology*, *IEEE Transactions on Cybernetics*, and *IEEE/ASME Transactions on Mechatronics*.

His research interests include intelligent and robust control, robotics, mechatronics, and their engineering applications.

E-mail: hjgao@hit.edu.cn (Corresponding author)
ORCID iD: 0000-0001-5554-5452