Fuzzy Control Based on Reinforcement Learning and Subsystem Error Derivatives for Strict-Feedback Systems With an Observer

Dongdong Li and Jiuxiang Dong, *Member, IEEE*

***Abstract*—In this article, a novel optimized fuzzy adaptive control method based on tracking error derivatives of subsystems is proposed for strict-feedback systems with unmeasurable states. A cost function based on the tracking error derivative is used. It not only solves the problem that the traditional input quadratic cost function at the infinite time is unbounded, but also solves the problem that the optimal control input derived from the cost function with exponential discount factor cannot make the error asymptotically stable. Considering the case where the states are unmeasurable, a fuzzy state observer is designed that removes the restriction of the Hurwitz equation for the gain parameters. Based on reinforcement learning, the observer, and error derivative cost function, an improved optimized backstepping control method is given. Using observed information and actor-critic structure to train fuzzy logic systems online, the control inputs are obtained to achieve approximate optimal control. Finally, all closed-loop signals are proved to be bounded by the Lyapunov method, and the effectiveness and advantages of the proposed algorithm are verified through two examples.**

***Index Terms*—Adaptive dynamic programming (ADP), fuzzy adaptive control, fuzzy logic systems (FLSs), fuzzy state observer, optimized backstepping control (OBC), reinforcement learning (RL).**

I. Introduction

OPTIMAL control was first proposed in the last century by Bellman [1] to optimize control performance. After decades of development, the optimal control theory has been one of the main research directions in the field of cost control and has been widely studied. Optimal control is achieved by minimizing a prescribed performance function (also called cost function or value function). The optimal control policy can be derived from the Hamilton–Jacobi–Bellman (HJB) equation obtained according to the Bellman's optimality principle [2]. However, the analytical solution of HJB cannot be obtained at present. To solve this problem, using the adaptive dynamic programming [or called reinforcement learning (RL)] method to obtain the approximate optimal solution of the HJB equation online has been widely studied [3], [4]. RL was originally used to study discrete-time (DT) systems to obtain approximate solutions by iterative optimization. It was later extended to the studies of continuous-time (CT) system control problems, resulting in many classical results [5], [6], [7], [8], [9], [10], [11].

RL is often used in the studies of optimal tracking control (OTC) problems such as [6], [12], [13], [14], and [15]. However, one issue that is often ignored when studying OTC problems is the presence of a steady-state feedforward part in the optimal control input. The optimal control input of the OTC problem is divided into two parts, i.e., the feedforward part and the feedback part [16]. The feedforward term is a steady-state term that enables the system to perfectly track the reference trajectory, while the feedback part is obtained by minimizing the cost function [17]. Since there is a steady-state feedforward component in the optimal control input, the integrated part of the traditional cost function has a steady-state input quadratic term that cannot be eliminated, thus the integration at the infinite time is unbounded, and the significance of optimization is lost [17]. To solve this problem, a cost function with an exponential term discount factor is used [17], [18], which allows the integrated part of the cost function to monotonically decrease with time. However, the optimal control policy derived from this cost function can only make the tracking error bounded unless the discount factor is zero [17], [18]. When we study the OTC problem, the discount factor cannot be chosen to be zero. For DT systems, the authors in [19] proposed a new value function based on the tracking error to solve these two problems. However, these problems still exist in CT systems.

Strict-feedback systems are nonlinear systems with classical triangular structure that have been extensively studied in recent decades. Many advanced control techniques for strict-feedback systems are involved in various fields, such as optimal control [13], [20], [21], [22], fuzzy control [23], [24], [25], [26],

Manuscript received 14 July 2022; revised 19 October 2022 and 19 November 2022; accepted 6 December 2022. Date of publication 9 December 2022; date of current version 4 August 2023. This work was supported in part by the National Natural Science Foundation of China under Grant 62273079 and Grant 61420106016, in part by the Fundamental Research Funds for the Central Universities in China under Grant N2004002, Grant N2104005, and Grant N182608004, and in part by the Research Fund of State Key Laboratory of Synthetical Automation for Process Industries in China under Grant 2013ZCX01. *(Corresponding author: Jiuxiang Dong.)*

The authors are with the College of Information Science and Engineering, Northeastern University, Shenyang 110819, China, with the Key Laboratory of Vibration and Control of Aero-Propulsion Systems Ministry of Education of China, Northeastern University, Shenyang 110819, China, and also with the State Key Laboratory of Synthetical Automation of Process Industries, Northeastern University, Shenyang 110819, China (e-mail: lidongdongyq@163.com; dongjiuxiang@163.com).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TFUZZ.2022.3227993.

Digital Object Identifier 10.1109/TFUZZ.2022.3227993

1063-6706 © 2022 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See https://www.ieee.org/publications/rights/index.html for more information.

Authorized licensed use limited to: TIANJIN UNIVERSITY. Downloaded on November 21,2023 at 11:34:39 UTC from IEEE Xplore. Restrictions apply.

multiagent systems [7], [27], [28], fault-tolerant control [27], etc. And, for the OTC problem of strict-feedback systems, the authors in [20], [21], [22], and [27] investigated the complete RL-based optimized backstepping technique. For strict-feedback systems, the most effective analysis method is the backstepping technique [20]. In [20], [21], [22], and [27], the authors cleverly introduced RL into the backstepping technique to obtain the OBC algorithms. These OBC algorithms relax the persistent excitation condition and have simpler weight adaptive laws.

Fuzzy control, one of the earliest applications of fuzzy sets and systems, has proven to be a successful control method for many complex nonlinear systems and even nonanalytical systems [29]. In many cases, it has been suggested as an alternative to traditional control techniques [29]. The fuzzy control algorithms consist of a set of heuristic control rules, and fuzzy sets and fuzzy logic are used to represent linguistic terms and evaluation rules, respectively [29]. Since the FLSs can approximate unknown nonlinear functions, they are commonly used in fuzzy control [23], [24], [25], [26], [30], [31]. However, most control methods cannot be used directly when the system states are not measurable or the dynamic functions are completely unknown. To solve this problem, the approximate system state information is generally obtained by designing a state observer, and many articles have reported on this classical approach such as [23], [24], [26], [32], [33], [34], [35], [36], [37], and [38]. However, the methods in [23], [24], [26], [32], [33], [34], [35], [36], [37], and[38] need to design the observer gain constants to satisfy the Hurwitz equation. In the recently reported study [39], an observer based on the FLSs was proposed to avoid this problem. Inspired by the aforementioned studies, the near-optimal control can be achieved by training the controller online through RL based on the information from the fuzzy observer.

In existing OBC methods [20], [21], [27], [28], the cost functions used are all $\int_t^\infty (e^2(\tau)+u^2(\tau))d\tau$. Although this is intuitive for input optimization, the problem exists that the cost function is unbounded, as mentioned in the second paragraph of the introduction. Then, what cost function can be used to solve the following two problems:

1) the cost function under infinite time is bounded;
2) the optimal control input derived from the cost function allows the error to be asymptotically stable, i.e., the tracking error is completely eliminated.

It is also important to consider how to analyze and apply such cost functions in strict-feedback systems with unmeasurable states.

Inspired by the aforementioned problems, a new FLSs online learning algorithm is proposed. The main contributions of this article are summarized as follows.

1) The traditional cost function in [20], [21], [22], and[28] is unbounded at infinite time, which is not considered. The optimal control input derived from the cost function in [17] and[18] does not make the system asymptotically stable, and the tracking error cannot be further eliminated. In this article, a cost function based on the tracking error derivative is used and its advantages are demonstrated. By introducing the input to the error derivative, the cost function does not contain the steady-state quadratic term. Thus, there is no problem that the cost function is unbounded. Furthermore, it is proved that the optimal control input obtained from the cost function can make the error system asymptotically stable.
2) Based on this cost function and RL, the improved OBC algorithm is designed to train the FLSs online and achieve approximate optimal control. Since the controller is obtained by optimizing the cost function based on the tracking error and its derivative, it has a significant optimization effect on the error and its rate of change. The algorithm not only reduces the tracking error but also suppresses overshoot and jitters of error, which is verified by two examples. The controller contains the error and its correlation term, so the input overshoot and jitters are also suppressed.
3) Since the system states are unmeasurable, an FLSs-based state observer is designed. Different from [23], [24], [32], [33], [34], [35], [36], [37], and [38], it allows the observed state errors to be bounded without the requirement of the Hurwitz equation for the gain parameters. A novel RL algorithm based on the state observer and the error derivative cost function is proposed. The approximate error of the HJB equation is used to train the fuzzy weights online. Finally, all closed-loop signals are proved to be semi-globally ultimately uniformly bounded (SGUUB) by the Lyapunov method.

II. System Description and Preliminary Knowledge

A. System Description

Consider the following nonlinear system in strict-feedback form as

$$\begin{cases} \dot{x}_i(t) = x_{i+1}(t) + f_i(\underline{x}_i(t)), i = 1, \ldots, n-1 \\ \dot{x}_n(t) = u + f_n(\underline{x}_n(t)) \\ y(t) = x_1(t) \end{cases} \tag{1}$$

where $x_1 \in \mathbb{R}$ is a measurable state, $y(t) \in \mathbb{R}$ is output of the system, $u \in \mathbb{R}$ is actual input, and $x_2 \in \mathbb{R}, \ldots, x_n \in \mathbb{R}$ are the unmeasurable states. In other words, only the information of x_1 is available, while the information of other states is unknown and unavailable in the algorithm. Moreover, $f_i(\underline{x}_i(t)) \in \mathbb{R}$ is unknown smooth function with $\underline{x}_i = [x_1, \ldots, x_i]^T \in \mathbb{R}^i$, $i = 1, \ldots, n$.

Definition 1: (SGUUB) [22] For an initial value $x(0)$ located in a compact set Ω, if there is $\mathcal{Z} > 0$ and $\mathcal{T}(x(0), \mathcal{Z}) > 0$ such that

$$x(t) \in \{x(t) | \parallel x(t) \parallel \leq \mathcal{Z}\} \quad \forall t > \mathcal{T}(x(0), \mathcal{Z}). \tag{2}$$

Then, solution $x(t)$ of $\dot{x}(t) = f(t, x)$ is said to be SGUUB.

Assumption 1: The reference trajectory x_r is known, bounded, and differentiable and the derivative $\dot{x}_r$ is known and bounded.

Control Objective: Design an algorithm to achieve the following three objectives:

Authorized licensed use limited to: TIANJIN UNIVERSITY. Downloaded on November 21,2023 at 11:34:39 UTC from IEEE Xplore. Restrictions apply.

1) solve the unbounded problem of the cost function in traditional optimized backstepping control [20], [21], [27], [28];
2) the output $y(t)$ can track the reference trajectory x_r;
3) all closed-loop error signals are SGUUB.

B. Fuzzy Logic Systems (FLSs)

An FLS consists of a fuzzy rule base, fuzzification, and defuzzification operators. The fuzzy rule base consists of the following inference rules:

$\mathcal{R}^j$: If x_1 is $\mathcal{A}_1^j$ and x_2 is $\mathcal{A}_2^j$ and ... and x_n is $\mathcal{A}_n^j$
then y is B^j, $j = 1, \ldots, m$
where $\mathcal{A}^j$ and B^j are fuzzy sets in $\mathbb{R}$.

Through singleton fuzzifier, product inference machine, center average defuzzifier, and fuzzy rule base with Gaussian membership function, the FLS can be described as

$$y(x) = \frac{\sum_{j=1}^{m} \bar{y}_j \prod_{i=1}^{n} \mu_{\mathcal{A}_i^j}(x_i)}{\sum_{j=1}^{m} [\prod_{i=1}^{n} \mu_{\mathcal{A}_i^j}(x_i)]}$$

where the $\bar{y}_j \in \mathbb{R}$ is the point which maximizes the function $\mu_{B_i^j}(\cdot)$. Define the fuzzy basis functions as

$$\phi^j = \frac{\prod_{i=1}^{n} \mu_{\mathcal{A}_i^j}(x_i)}{\sum_{j=1}^{m} [\prod_{i=1}^{n} \mu_{\mathcal{A}_i^j}(x_i)]}.$$

Let $\mathcal{W} = [\bar{y}_1, \bar{y}_2, \ldots, \bar{y}_m]^T = [\mathcal{W}_1, \mathcal{W}_2, \ldots, \mathcal{W}_m]^T$ and $\phi(x) = [\phi^1(x), \phi^2(x), \ldots, \phi^m(x)]^T$, then the FLS is

$$y(x) = \mathcal{W}^T \phi(x).$$

Since the FLSs are required in the fuzzy state observer and RL. Therefore, the following Lemma on FLSs is introduced.

Lemma 1 (See [4] and [40]): A continuous unknown function $\mathcal{Y}(x)$ is defined on a compact set $\mathcal{Q}$, for any positive constant ε, there exists the FLSs $\hat{\mathcal{Y}}(x \mid \mathcal{W}^*) = \mathcal{W}^{*T}\phi(x)$ such that

$$\sup_{x \in \mathcal{Q}} \mid \mathcal{Y}(x) - \mathcal{W}^{*T}\phi(x) \mid < \varepsilon$$

where $x \in \mathbb{R}^n$ is the input of FLS, $\mathcal{W}^* \in \mathbb{R}^m$ is ideal weight vector, and ε is fuzzy approximate error with $\varepsilon \to 0$ as $j \to \infty$. Moreover, $\phi^j(x), j = 1, \ldots, m$, are the fuzzy basis functions, $j > 1$ is the fuzzy rule number. It is worth noting that the Gaussian basis functions satisfy $0 < \phi(x)^T\phi(x) \leq 1$. In addition, the ideal weight vector $\mathcal{W}^*$ is bounded constant vector.

For RL, the training part is mainly the weight vector $\hat{\mathcal{W}}$, which is an approximation of the ideal weight vector. Similar to [4], [7], [9], and[10], FLSs online learning is achieved by training $\hat{\mathcal{W}}$ online.

III. Discussion of Different Cost Functions

To achieve control objective 1), the existing problems are discussed in this section. In fact, a strict-feedback system can be seen as a combination of n subsystems. First, the subsystem 1 is singled out for discussion

$$\dot{x}_1 = f_1(x_1) + x_2. \tag{3}$$

When using the backstepping method to study the tracking problem, we hope that the designed virtual controller α_1 satisfies $\alpha_1 \to x_2$. The ideal case is "$\alpha_1^* = x_2$" where α_1^* is the optimal virtual control. Thus, the ideal subsystem can be obtained as

$$\dot{x}_1 = f_1(x_1) + \alpha_1^*. \tag{4}$$

The ideal subsystem (4) is a single-input single-output system. When considering the OTC problem, the optimal control α_1^* is divided into two parts, i.e., feedforward input u_d and feedback input $\alpha_{e_1}^*$ [16]. If $x_1 = x_r$, then it can be considered that perfect tracking is achieved. By substituting $x_1 = x_r$ into system (4), the reference trajectory dynamics can be obtained as

$$\dot{x}_r = f_1(x_r) + u_d. \tag{5}$$

In fact, the tracking objective is to design a controller α_1 so that the state of system (5) can be tracked perfectly by output of the system (4). If the system dynamic function f_1, the reference trajectory and its derivative are known, then feedforward control u_d can be obtained by

$$u_d = \dot{x}_r - f_1(x_r). \tag{6}$$

Define the output tracking error as

$$\begin{aligned} \dot{z}_1 &= \dot{x}_1 - \dot{x}_r \\ &= f_1(z_1 + x_r) + \alpha_1^* - \dot{x}_r \\ &= f_1(z_1 + x_r) + \alpha_{e_1}^* + \dot{x}_r - f(x_r) - \dot{x}_r. \end{aligned} \tag{7}$$

Let $f_{z_1} = f_1(z_1 + x_r) + \dot{x}_r - f(x_r) - \dot{x}_r = f_1(z_1 + x_r) - f(x_r)$, then

$$\dot{z}_1 = f_{z_1} + \alpha_{e_1}^* \tag{8}$$

where $\alpha_{e_1}^*$ is actually obtained by optimizing the following cost function:

$$\mathcal{J}_1^*(\alpha_{e_1}, z_1(t)) = \min_{\alpha_{e_1} \in \Omega} \int_t^\infty (z_1^2(\tau) + \alpha_{e_1}^2) d\tau. \tag{9}$$

Subsequently, it is easy to obtain α_1^* through

$$\alpha_1^* = u_d + \alpha_{e_1}^*. \tag{10}$$

In fact, this is equivalent to obtaining α_1^* by minimizing the following cost function:

$$\mathcal{J}_1^*(\alpha_1, z_1(t)) = \min_{\alpha_1 \in \Omega} \int_t^\infty (z_1^2(\tau) + \alpha_1^2) d\tau \tag{11}$$

and the optimal control input is

$$\alpha_1^* = -\frac{1}{2} \frac{\partial \mathcal{J}_1^*(\alpha_1, z_1(t))}{\partial z_1}. \tag{12}$$

This is exactly the approach used by the existing RL-based OBC algorithms [20], [21], [27], [28].

Remark 1: Although the optimized input in (11) is α_1, α_1 also contains u_d that cannot be optimized and we can only make α_{e_1} as small as possible. This also indicates the fact: even if α_{e_1} is optimized to 0, there is still u_d that cannot be eliminated. In this way, there is a steady-state quadratic term in the integrated part of the cost function $\mathcal{J}_1^*(\alpha_1, z_1(t))$, then the integral function $\mathcal{J}_1^*$

Authorized licensed use limited to: TIANJIN UNIVERSITY. Downloaded on November 21,2023 at 11:34:39 UTC from IEEE Xplore. Restrictions apply.

will be infinite at infinite time, and clearly the optimization is defective.

With the intensive research of RL, the following improved cost function has been proposed such as

$$\mathcal{J}_1^*(z_1(t)) = \min_{\alpha_1 \in \Omega} \int_t^\infty e^{-\gamma(t-\tau)}(z_1^2(\tau) + \alpha_1^2)d\tau \tag{13}$$

where $0 < \gamma < 1$ and $\mathcal{J}_1 = \int_t^\infty e^{-\gamma(t-\tau)}(z_1^2(\tau) + \alpha_1^2)d\tau$. The tracking error system is

$$\dot{z}_1 = f_1 + \alpha_1^* - \dot{x}_r \tag{14}$$

and the HJB equation can be obtained as

$$\mathcal{H}^*\left(z_1, \alpha_1^*, \frac{\partial \mathcal{J}_1^*}{\partial z_1}\right) = z_1^2 + \alpha_1^{*2} - \gamma \mathcal{J}_1^* + \frac{\partial \mathcal{J}_1^*}{\partial z_1}(f_1 + \alpha_1^* - \dot{x}_r) = 0. \tag{15}$$

The derivation process of the HJB equation is similar to (17)–(20). Let $\partial \mathcal{H}(z_1, \alpha_1, \partial \mathcal{J}_1^*/\partial z_1)/\partial \alpha_1 = 0$, the optimal control is

$$\alpha_1^* = -\frac{1}{2}\frac{\partial \mathcal{J}_1^*}{\partial z_1}. \tag{16}$$

Lemma 2 (See [17] and [18]): Consider the tracking error system (14), the optimal cost function (13), and the HJB (15). If $\gamma = 0$, the optimal control input (16) can make the system (14) asymptotically stable.

Remark 2: By adding an exponential term with a discount factor, the integrated part of $\mathcal{J}_1^*$ is made to be monotonically decreasing with time t. If the value of $\mathcal{J}_1$ is infinite, then it is meaningless to consider optimizing the value of $\mathcal{J}_1$. In this way, the value of $\mathcal{J}_1$ is bounded at infinite time, so it makes sense to consider optimizing $\mathcal{J}_1$. Then, the problem in Remark 1 is solved. However, this leads to the following two problems:

1) the choice of the discount factor affects the magnitude of the tracking error;
2) the optimal control input cannot completely eliminate the tracking error.

$\gamma = 0$ can be chosen only when the reference trajectory converges asymptotically, otherwise it can only be concluded that the error is bounded. However, when the reference trajectory converges asymptotically, the tracking problem degenerates to the stabilization problem.

Obviously, this is not the desired result. So, how can we solve the problem in Remark 1 and get asymptotically stable result? Define the following optimal cost function as

$$\mathcal{J}_1^*(z_1(t)) = \min_{\alpha_1 \in \Omega} \int_t^\infty (z_1^2(\tau) + \dot{z}_1^2(\tau))d\tau. \tag{17}$$

According to the property of integral operator and considering an interval Δt, there exists

$$\mathcal{J}_1^*(z_1(t)) = \min_{\alpha_1 \in \Omega} \int_t^{t+\Delta t} (z_1^2(\tau) + \dot{z}_1^2(\tau))d\tau + \mathcal{J}_1^*(z_1(t+\Delta t)). \tag{18}$$

Taking the limit of both sides of (18) for $\Delta t \to 0$, we can get

$$\lim_{\Delta t \to 0} \frac{1}{\Delta t} \min_{\alpha_1 \in \Omega} \int_t^{t+\Delta t} (z_1^2(\tau) + \dot{z}_1^2(\tau))d\tau + \lim_{\Delta t \to 0} \frac{\mathcal{J}_1^*(z_1(t+\Delta t)) - \mathcal{J}_1^*(z_1(t))}{\Delta t} = 0. \tag{19}$$

The aforementioned equation can be rewritten as $z_1^2(\alpha_1^*(t), x_1(t)) + \dot{z}_1^2(\alpha_1^*(t), x_1(t)) + d\mathcal{J}_1^*(z_1(t))/dt = 0$. Thus, we can obtain the HJB equation as

$$\mathcal{H}^*\left(z_1, \alpha_1^*, \frac{\partial \mathcal{J}_1^*}{\partial z_1}\right) = z_1^2 + \dot{z}_1^2 + \frac{\partial \mathcal{J}_1^*}{\partial z_1}(f_1 + \alpha_1^* - \dot{x}_r) = 0. \tag{20}$$

Let $\partial \mathcal{H}(z_1, \alpha_1, \partial \mathcal{J}_1^*/\partial z_1)/\partial \alpha_1 = 0$, the optimal control is

$$\alpha_1^* = -\frac{1}{2}\frac{\partial \mathcal{J}_1^*}{\partial z_1} - f_1 + \dot{x}_r. \tag{21}$$

Remark 3: As seen in (17), there is no steady-state quadratic term in the integrated part, therefore the problem in Remark 1 does not exist. Expanding $\dot{z}_1^2$ reveals that the cost function (17) also has optimization effect for input α_1. Moreover, it can be seen from (21) that information of the reference trajectory is introduced into the optimal controller by using the new cost function, which will improve the tracking effect.

Theorem 1: Considering the error system (14), the cost function (17), and the HJB (20), the optimal control input (21) can make the system (14) asymptotically stable.

Proof: We choose the Lyapunov function as

$$\mathcal{L} = \mathcal{J}_1^*(z_1). \tag{22}$$

Transforming the form of (21) yields

$$\frac{\partial \mathcal{J}_1^*}{\partial z_1} = -2(\alpha_1^* + f_1 - \dot{x}_r). \tag{23}$$

The time derivative of $\mathcal{L}$ is

$$\dot{\mathcal{L}} = \frac{\partial \mathcal{J}_1^*}{\partial z_1}(f_1 + \alpha_1^* - \dot{x}_r) = -2(\alpha_1^* + f_1 - \dot{x}_r)^2 \leq 0. \tag{24}$$

Therefore, the optimal input α_1^* can make the tracking error asymptotically stable. ■

Remark 4: The previous discussions have shown the differences between the three cost functions. The cost function (17) can solve the problems of traditional cost functions by introducing the input into the error derivative. Thus, it is independent of the discount factor and can completely eliminate the tracking error. Since the error dynamics depends on the state and the input, the improved cost function can also optimize the control input indirectly and does not lose the optimization effect for the input. The aforementioned discussions are about subsystem 1, in fact $1, \ldots, n$ subsystems are similar. Moreover, in addition, all previous discussions are based on the optimal cost function $\mathcal{J}_1^*$ being known. Because $\mathcal{J}_1^*$ contains the information about future time, its value cannot be directly obtained. When the cost function and optimal input are approximated and trained online using FLSs, Theorem 3 can only conclude that all signals are SGUUB due to the effects of unknown states and approximate errors and the use of the FLSs.

Authorized licensed use limited to: TIANJIN UNIVERSITY. Downloaded on November 21,2023 at 11:34:39 UTC from IEEE Xplore. Restrictions apply.

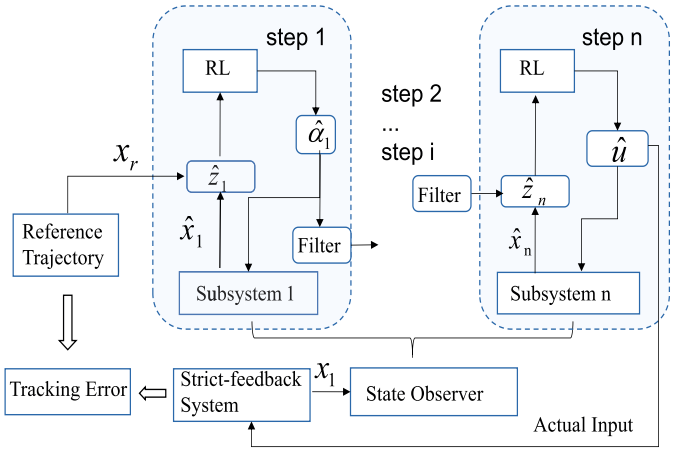

Fig. 1. Block diagram of the proposed algorithm.

IV. Application of the Improved Cost Function

The advantages of the improved cost function are discussed in Section III, however, the analytical solution of the HJB equation cannot be obtained, so an online learning algorithm is proposed in Section IV to obtain an approximate solution of the HJB. The block diagram of the proposed algorithm is shown in Fig. 1.

A. Adaptive Fuzzy State Observer

Since the states of the system (1) are unmeasurable, their information cannot be used directly. The general observers require the gain parameters to satisfy the Hurwitz equation. To solve the problem and relax the requirement, a fuzzy state observer is designed in this section. System (1) is rewritten as

$$\begin{cases} \dot{x}_1(t) = x_2(t) + f_1(\underline{x}_1(t)) \\ \dot{x}_2(t) = x_3(t) - \delta_2 x_2(t) + \mathcal{F}_2(\underline{x}_2(t)) \\ \dots \\ \dot{x}_n(t) = u - \delta_n x_n(t) + \mathcal{F}_n(\underline{x}_n(t)) \end{cases} \tag{25}$$

where $\delta_2, \dots, \delta_n > 0$ are designed constants. Let $\underline{\hat{x}}_i = [\hat{x}_1, \dots, \hat{x}_i]^T \in \mathbb{R}^i$ is estimated state vector and $i = 1, \dots, n$. We use FLSs to reconstruct $f_1(\underline{x}_1(t))$ and $\mathcal{F}_i(\underline{x}_i(t))$, i.e., $f_1(\underline{x}_1(t)) = \varpi_1^{*T}\varphi_1(\underline{x}_1(t)) + \epsilon_1(\underline{x}_1(t))$ and $\mathcal{F}_i(\underline{x}_i(t)) = \varpi_i^{*T}\varphi_i(\underline{x}_i(t)) + \epsilon_i(\underline{x}_i(t))$, where ϵ_1 and ϵ_i are the approximate errors of FLSs and are bounded [41]. The following fuzzy state observer can be obtained as

$$\begin{cases} \dot{\hat{x}}_1(t) = \hat{x}_2(t) + \delta_1\tilde{x}_1(t) + \hat{\varpi}_1^T\varphi_1(\underline{x}_1(t)) \\ \dot{\hat{x}}_2(t) = \hat{x}_3(t) - \delta_2\hat{x}_2(t) + \hat{\varpi}_2^T\varphi_2(\underline{\hat{x}}_2(t)) \\ \dots \\ \dot{\hat{x}}_n(t) = u - \delta_n\hat{x}_n(t) + \hat{\varpi}_n^T\varphi_n(\underline{\hat{x}}_n(t)) \end{cases} \tag{26}$$

where $\hat{\varpi}_i$ and φ_i are approximate weight vector and the basis function with $i = 1, \dots, n$. Based on the actual system (25) and the observed system (26), the observed error dynamics can be obtained as

$$\begin{cases} \dot{\tilde{x}}_1(t) = \tilde{x}_2(t) - \delta_1\tilde{x}_1(t) + f_1(\underline{x}_1(t)) - \hat{\varpi}_1^T\varphi_1(\underline{x}_1(t)) \\ \dot{\tilde{x}}_2(t) = \tilde{x}_3(t) - \delta_2\tilde{x}_2(t) + \mathcal{F}_2(\underline{x}_2(t)) - \hat{\varpi}_2^T\varphi_2(\underline{\hat{x}}_2(t)) \\ \dots \\ \dot{\tilde{x}}_n(t) = -\delta_n\tilde{x}_n(t) + \mathcal{F}_n(\underline{x}_n(t)) - \hat{\varpi}_n^T\varphi_n(\underline{\hat{x}}_n(t)) \end{cases} \tag{27}$$

where $\tilde{x}_i = x_i - \hat{x}_i$ is the error between the observed state and the actual state. The adaptive laws of the FLSs are designed as

$$\dot{\hat{\varpi}}_1 = -\bar{\gamma}_1\hat{\varpi}_1 + \varphi_1(\underline{\hat{x}}_1)\tilde{x}_1, \quad \dot{\hat{\varpi}}_i = -\bar{\gamma}_i\varphi_i(\underline{\hat{x}}_i)\varphi_i(\underline{\hat{x}}_i)^T\hat{\varpi}_i \tag{28}$$

where $\bar{\gamma}_1, \bar{\gamma}_i > 0$ are designed parameters and $i = 2, \dots, n$.

Theorem 2: Considering that the states of system (1) are unmeasurable and the initial value of x_1 is bounded and available, the observed errors $\tilde{x}_i$ and the weight errors $\tilde{\varpi}_i$ are SGUUB using the fuzzy state observer (26) and the adaptive laws (28).

Proof: Select the following Lyapunov functions as

$$\mathcal{V}^o = \frac{1}{2}\sum_{i=1}^{n}\tilde{x}_i^2(t) + \frac{1}{2}\sum_{i=1}^{n}\tilde{\varpi}_i^T\tilde{\varpi}_i \tag{29}$$

where $\tilde{\varpi}_i = \varpi_i^* - \hat{\varpi}_i$ is FLSs weight error. For simplicity, the latter $f_i(\underline{x}_i(t))$ will be abbreviated to f_i, and the other symbols are similar. The time derivative of $\mathcal{V}^o$ is

$$\begin{aligned} \dot{\mathcal{V}}^o =& \tilde{x}_1\{\tilde{x}_2 - \delta_1\tilde{x}_1 + f_1 - \hat{\varpi}_1^T\varphi_1\} + \sum_{i=2}^{n-1}\tilde{x}_i\{\tilde{x}_{i+1} - \delta_i\tilde{x}_i \\ &+ \mathcal{F}_i - \hat{\varpi}_i^T\varphi_i\} + \tilde{x}_n\{-\delta_n\tilde{x}_n + \mathcal{F}_n - \hat{\varpi}_n^T\varphi_n\} + \tilde{\varpi}_1^T \\ &\times\{\bar{\gamma}_1\hat{\varpi}_1 - \varphi_1\tilde{x}_1\} + \sum_{i=2}^{n}\tilde{\varpi}_i^T\{\bar{\gamma}_i\varphi_i\varphi_i^T\hat{\varpi}_i\} \\ =& \tilde{x}_1\{\tilde{x}_2 - \delta_1\tilde{x}_1 - \tilde{\varpi}_1^T\varphi_1 + \epsilon_1\} + \sum_{i=2}^{n-1}\tilde{x}_i\{\tilde{x}_{i+1} - \delta_i\tilde{x}_i \\ &- \tilde{\varpi}_i^T\varphi_i + \epsilon_i\} + \tilde{x}_n\{-\delta_n\tilde{x}_n - \tilde{\varpi}_n^T\varphi_n + \epsilon_n\} \\ &+ \tilde{\varpi}_1^T\{\bar{\gamma}_1\hat{\varpi}_1 - \varphi_1\tilde{x}_1\} + \sum_{i=2}^{n}\tilde{\varpi}_i^T\{\bar{\gamma}_i\varphi_i\varphi_i^T\hat{\varpi}_i\} \\ =& -\sum_{i=1}^{n}\delta_i\tilde{x}_i^2 + \sum_{i=1}^{n-1}\tilde{x}_i\tilde{x}_{i+1} + \sum_{i=1}^{n}\tilde{x}_i\tilde{\varpi}_i^T\varphi_i + \sum_{i=1}^{n}\tilde{x}_i\epsilon_i \\ &- \tilde{x}_1\tilde{\varpi}_1^T\varphi_1 + \bar{\gamma}_1\tilde{\varpi}_1^T\hat{\varpi}_1 + \sum_{i=2}^{n}\bar{\gamma}_i\tilde{\varpi}_i^T\varphi_i\varphi_i^T\hat{\varpi}_i. \end{aligned} \tag{30}$$

According to Young's inequality, it is easy to get

$$\begin{aligned} &\tilde{x}_i\tilde{x}_{i+1} \le \frac{1}{2}\tilde{x}_i^2 + \frac{1}{2}\tilde{x}_{i+1}^2, \tilde{x}_i\epsilon_i \le \frac{1}{2}\tilde{x}_i^2 + \frac{1}{2}\epsilon_i^2 \\ &\tilde{x}_i\tilde{\varpi}_i^T\varphi_i \le \frac{1}{2}\tilde{x}_i^2 + \frac{1}{2}\tilde{\varpi}_i^T\varphi_i\varphi_i^T\tilde{\varpi}_i \end{aligned} \tag{31}$$

and since $\tilde{\varpi}_i = \varpi_i^* - \hat{\varpi}_i$, there is

$$\begin{aligned} \tilde{\varpi}_1^T\hat{\varpi}_1 =& -\frac{1}{2}\tilde{\varpi}_1^T\tilde{\varpi}_1 - \frac{1}{2}\hat{\varpi}_1^T\hat{\varpi}_1 + \frac{1}{2}\varpi_1^{*T}\varpi_1^* \\ \tilde{\varpi}_i^T\varphi_i\varphi_i^T\hat{\varpi}_i =& -\frac{1}{2}(\tilde{\varpi}_i^T\varphi_i)^2 - \frac{1}{2}(\hat{\varpi}_i^T\varphi_i)^2 + \frac{1}{2}(\varpi_i^{*T}\varphi_i)^2. \end{aligned} \tag{32}$$

Authorized licensed use limited to: TIANJIN UNIVERSITY. Downloaded on November 21,2023 at 11:34:39 UTC from IEEE Xplore. Restrictions apply.

Substituting (31) and (32) into (30) yields

$$\begin{aligned}\dot{\mathcal{V}}^o \leq & -\sum_{i=1}^{n}\left(\delta_i-\frac{3}{2}\right)\tilde{x}_i^2-\frac{\bar{\gamma}_1}{2}\tilde{\varpi}_1^T\tilde{\varpi}_1 \\ & -\sum_{i=2}^{n}\left(\frac{\bar{\gamma}_i}{2}-\frac{1}{2}\right)\tilde{\varpi}_i^T\varphi_i\varphi_i^T\tilde{\varpi}_i+\mathcal{B}^o \\ \leq & -\mathcal{A}^o\mathcal{V}^o+\mathcal{B}^o \end{aligned} \tag{33}$$

where $\mathcal{B}^o=1/2\sum_{i=1}^{n}\epsilon_i^2+\bar{\gamma}_1\varpi_1^{*T}\varpi_1^*/2+\sum_{i=2}^{n}\bar{\gamma}_i\varpi_i^{*T}\varphi_i\varphi_i^T\varpi_i^*/2$, $\mathcal{A}^o=\min\{2\delta_1-3,\ldots,2\delta_n-3,\bar{\gamma}_1,(\bar{\gamma}_2-1)\lambda_{\min}^2(\varphi_i\varphi_i^T),\ldots,(\bar{\gamma}_n-1)\lambda_{\min}^n(\varphi_n\varphi_n^T)\}$ and the $\lambda_{\min}^i(\varphi_i\varphi_i^T)$ is the minimum eigenvalue of the $\varphi_i\varphi_i^T$ matrix. According to Lemma 1, it is clear that both $\varphi_i\varphi_i^T$ and ϖ^* are bounded. The FLSs ideal residuals ϵ_1 and ϵ_i are also bounded [41]. Then, $\mathcal{B}^o$ can be considered as bounded.

Therefore, the observed errors and the FLSs weight errors are SGUUB. The proof of Theorem 2 is completed. ■

According to (33), a smaller $\mathcal{B}^o$ and a larger $\mathcal{A}^o$ will lead to the smaller observed errors and the weight errors. Therefore, choosing larger σ_i can make the errors smaller.

B. RL-Based Improved Optimized Backstepping

This section focuses on how to design an online learning algorithm for OTC by using RL, improved cost function (17), and fuzzy state observer (26). Define the following coordinate transformations as

$$\begin{aligned} z_1=x_1-x_r,\quad \hat{z}_1=\hat{x}_1-x_r,\quad \hat{z}_i=\hat{x}_i-\bar{\alpha}_i \\ \hbar_i=\bar{\alpha}_i-\alpha_{i-1},\quad i=2,\ldots,n. \end{aligned} \tag{34}$$

where z_1 is output tracking error and α_{i-1} is the virtual input. In addition, α_{i-1} is the input of the filter, $\hbar_i$ is filtered error, $\bar{\alpha}_i$ is the output of the filter, and the filter is defined as

$$\kappa_i\dot{\bar{\alpha}}_i+\bar{\alpha}_i=\alpha_{i-1},\quad \bar{\alpha}_i(0)=\alpha_{i-1}(0) \tag{35}$$

where $\kappa_i>0$ is the designed filter parameter.

Step 1: Using (26), the estimated tracking error is

$$\dot{\hat{z}}_1=\dot{\hat{x}}_1-\dot{x}_r=\hat{x}_2(t)+\delta_1\tilde{x}_1(t)+\hat{\varpi}_1^T\varphi_1(\underline{x}_1(t))-\dot{x}_r. \tag{36}$$

Define the optimal cost function as

$$\mathcal{J}_1^*(\hat{z}_1(t))=\min_{\alpha_1\in\Omega}\int_t^{\infty}(\hat{z}_1^2(\tau)+\dot{\hat{z}}_1^2(\tau))d\tau. \tag{37}$$

The advantages of the improved cost function have been discussed in Section III. α_1^* is the optimal virtual input and replaces $\hat{x}_2$ to obtain the HJB equation as

$$\begin{aligned}\mathcal{H}^*\left(\hat{z}_1,\alpha_1^*,\frac{\partial\mathcal{J}_1^*}{\partial\hat{z}_1}\right)=\hat{z}_1^2+\dot{\hat{z}}_1^2+\frac{\partial\mathcal{J}_1^*}{\partial\hat{z}_1}\{\alpha_1^*+\delta_1\tilde{x}_1 \\ +\hat{\varpi}_1^T\varphi_1(\underline{x}_1(t))-\dot{x}_r\}=0.\end{aligned} \tag{38}$$

Let $\partial\mathcal{H}(\hat{z}_1,\alpha_1,\partial\mathcal{J}_1^*/\partial\hat{z}_1)/\partial\alpha_1=0$, the optimal control is

$$\alpha_1^*(\hat{z}_1)=-\frac{1}{2}\frac{\partial\mathcal{J}_1^*}{\partial\hat{z}_1}-\{\delta_1\tilde{x}_1(t)+\hat{\varpi}_1^T\varphi_1(\underline{x}_1(t))-\dot{x}_r\}. \tag{39}$$

The $\partial\mathcal{J}_1^*/\partial\hat{z}_1$ is rewritten as

$$\frac{\partial\mathcal{J}_1^*}{\partial\hat{z}_1}=2k_1\hat{z}_1+\mathcal{J}_1^0(\hat{z}_1) \tag{40}$$

where $k_1>0$ is a designed constant and $\mathcal{J}_1^0=-2k_1\hat{z}_1+\partial\mathcal{J}_1^*/\partial\hat{z}_1$. Substituting (40) into (39) gives

$$\alpha_1^*=-k_1\hat{z}_1-\frac{1}{2}\mathcal{J}_1^0(\hat{z}_1)-\{\delta_1\tilde{x}_1(t)+\hat{\varpi}_1^T\varphi_1(\underline{x}_1(t))-\dot{x}_r\}. \tag{41}$$

Using the FLS to approximate $\mathcal{J}_1^0(\hat{z}_1)$ yields

$$\mathcal{J}_1^0(\hat{z}_1)=\mathcal{W}_1^{*T}\phi_1(\hat{z}_1)+\varepsilon_1(\hat{z}_1) \tag{42}$$

where $\mathcal{W}_1^*$ is ideal weight, $\phi_1(\hat{z}_1)$ is fuzzy basis function, and $\varepsilon_1(\hat{z}_1)$ is the ideal residual of FLS. They are subsequently abbreviated as ϕ_1 and ε_1, respectively.

Use actor FLS to generate a virtual controller $\hat{\alpha}_1$ as

$$\hat{\alpha}_1=-k_1\hat{z}_1-\frac{1}{2}\hat{\mathcal{W}}_{a1}^T\phi_1-\{\delta_1\tilde{x}_1+\hat{\varpi}_1^T\varphi_1-\dot{x}_r\}. \tag{43}$$

Using critic FLS to train the cost function, we have

$$\frac{\partial\hat{\mathcal{J}}_1}{\partial\hat{z}_1}=2k_1\hat{z}_1+\hat{\mathcal{W}}_{c1}^T\phi_1. \tag{44}$$

Actor FLS and critic FLS are collaboratively used and are referred to as actor-critic structure, which exists in [4], [7], [9], and [10]. Based on actor-critic FLSs, when an observer is used to observe the unknown states, it is considered an actor-critic-observer structure. Substituting (43) and (44) into HJB (38), the approximate error $\mathcal{E}_1$ is obtained as

$$\begin{aligned}\mathcal{E}_1=\hat{z}_1^2+\left(-k_1\hat{z}_1-\frac{1}{2}\hat{\mathcal{W}}_{a1}^T\phi_1\right)^2+\left(2k_1\hat{z}_1+\hat{\mathcal{W}}_{c1}^T\phi_1\right) \\ \times\left\{-k_1\hat{z}_1-\frac{1}{2}\hat{\mathcal{W}}_{a1}^T\phi_1\right\}=\hat{\mathcal{H}}\left(\hat{z}_1,\hat{\alpha}_1,\frac{\partial\hat{\mathcal{J}}_1}{\partial\hat{z}_1}\right).\end{aligned} \tag{45}$$

We need to find the optimal $\hat{\mathcal{W}}_{a1}$ making $\hat{\mathcal{H}}=0$. Let $\partial\hat{\mathcal{H}}/\partial\hat{\mathcal{W}}_{a1}=0$, it is easy to get

$$\frac{\partial\hat{\mathcal{H}}}{\partial\hat{\mathcal{W}}_{a1}}=\frac{1}{2}\phi_1\phi_1^T(\hat{\mathcal{W}}_{a1}-\hat{\mathcal{W}}_{c1})=0. \tag{46}$$

Assuming that $e_{\mathcal{W}_1}=\hat{\mathcal{W}}_{a1}-\hat{\mathcal{W}}_{c1}$ is the error, we construct the Lyapunov function as

$$\mathcal{S}(t)=\frac{1}{2}e_{\mathcal{W}_1}^Te_{\mathcal{W}_1}. \tag{47}$$

The time derivative of $\mathcal{S}(t)$ is

$$\dot{\mathcal{S}}(t)=e_{\mathcal{W}_1}^T(\dot{\mathcal{W}}_{a1}-\dot{\mathcal{W}}_{c1}). \tag{48}$$

If the weight update laws of actor-critic FLSs are designed as

$$\begin{aligned}\dot{\hat{\mathcal{W}}}_{a1}=-\phi_1\phi_1^T\left[\beta_{a1}(\hat{\mathcal{W}}_{a1}-\hat{\mathcal{W}}_{c1})+\beta_{c1}\hat{\mathcal{W}}_{c1}\right] \\ \dot{\hat{\mathcal{W}}}_{c1}=-\beta_{c1}\phi_1\phi_1^T\hat{\mathcal{W}}_{c1}\end{aligned} \tag{49}$$

where $\beta_{a1},\beta_{c1}>0$ are designed constants, then (48) can be written as

$$\dot{\mathcal{S}}(t)=-\beta_{a1}e_{\mathcal{W}_1}^T\phi_1\phi_1^Te_{\mathcal{W}_1}\leq 0. \tag{50}$$

Authorized licensed use limited to: TIANJIN UNIVERSITY. Downloaded on November 21,2023 at 11:34:39 UTC from IEEE Xplore. Restrictions apply.

Thus, $e_{\mathcal{W}_1}$ can converge asymptotically to 0. Then, (46) is satisfied. The optimal weights can be obtained by using (49).

Remark 5: The HJB equation is established when the optimal input α_1^* and the optimal cost function $\mathcal{J}_1^*$ are obtained. Since the analytical solution of the HJB equation is not available, α_1^* and $\mathcal{J}_1^*$ cannot be obtained. Thus, two sets of FLSs and other relevant nonlinear functions are used to form $\hat{\alpha}_1$ and $\partial\hat{\mathcal{J}}_1/\partial\hat{z}_1$. This will produce an approximate residual (45), and the residual drives the two sets of FLSs to keep learning online until the residual converge to 0. Then, the approximate optimal input and approximate optimal cost function are obtained. This is the principle of running the RL algorithm using FLSs in this article. Using the aforementioned design steps (45)–(50) to obtain the weight update laws has the following two advantages.

1) The forms of the obtained weight update laws are simpler than the update laws in [22] obtained by using the gradient descent algorithm.
2) The strict conditions of persistence excitation are relaxed.

The conditions are $\mathcal{W}_1 \neq 0$ and satisfies $\xi_1 I_m \leq \mathcal{W}_1\mathcal{W}_1^T \leq \xi_2 I_m$, where ξ_1 and ξ_2 are positive constants and $\mathcal{W}_1$ is weight vector.

Consider the following Lyapunov functions as

$$\mathcal{V}_1(t) = \frac{1}{2}\hat{z}_1^2 + \frac{1}{2}\tilde{\mathcal{W}}_{a1}^T\tilde{\mathcal{W}}_{a1} + \frac{1}{2}\tilde{\mathcal{W}}_{c1}^T\tilde{\mathcal{W}}_{c1} \tag{51}$$

where $\tilde{\mathcal{W}}_{a1} = \hat{\mathcal{W}}_{a1} - \mathcal{W}_1^*$ and $\tilde{\mathcal{W}}_{c1} = \hat{\mathcal{W}}_{c1} - \mathcal{W}_1^*$, thus $\dot{\tilde{\mathcal{W}}}_{c1} = \dot{\hat{\mathcal{W}}}_{c1}$ and $\dot{\tilde{\mathcal{W}}}_{a1} = \dot{\hat{\mathcal{W}}}_{a1}$. Using (34), (36), (43), and (49), we have

$$\begin{aligned}\dot{\mathcal{V}}_1(t) &= \hat{z}_1\left\{\hat{z}_2 + \hbar_2 - k_1\hat{z}_1 - \frac{1}{2}\hat{\mathcal{W}}_{a1}^T\phi_1\right\} - \tilde{\mathcal{W}}_{a1}^T\left\{\phi_1\phi_1^T\left(\beta_{a1}\right.\right.\\ &\quad\left.\left.\times(\hat{\mathcal{W}}_{a1} - \hat{\mathcal{W}}_{c1}) + \beta_{c1}\hat{\mathcal{W}}_{c1}\right)\right\} - \beta_{c1}\tilde{\mathcal{W}}_{c1}^T\phi_1\phi_1^T\hat{\mathcal{W}}_{c1}\\ &= \hat{z}_1\left\{\hat{z}_2 + \hbar_2 - k_1\hat{z}_1 - \frac{1}{2}\hat{\mathcal{W}}_{a1}^T\phi_1\right\} - \beta_{a1}\tilde{\mathcal{W}}_{a1}^T\phi_1\phi_1^T\hat{\mathcal{W}}_{a1}\\ &\quad + (\beta_{a1} - \beta_{c1})\tilde{\mathcal{W}}_{a1}^T\phi_1\phi_1^T\hat{\mathcal{W}}_{c1} - \beta_{c1}\tilde{\mathcal{W}}_{c1}^T\phi_1\phi_1^T\hat{\mathcal{W}}_{c1}.\end{aligned} \tag{52}$$

Using Young's inequality, one has

$$\begin{aligned}\hat{z}_1\hat{z}_2 &\leq \frac{1}{2}\hat{z}_1^2 + \frac{1}{2}\hat{z}_2^2, \hat{z}_1\hbar_2 \leq \frac{1}{2}\hat{z}_1^2 + \frac{1}{2}\hbar_2^2\\ \hat{z}_1\hat{\mathcal{W}}_{a1}^T\phi_1 &\leq \frac{1}{2}\hat{z}_1^2 + \frac{1}{2}(\hat{\mathcal{W}}_{a1}^T\phi_1)^2\\ \tilde{\mathcal{W}}_{a1}^T\phi_1\phi_1^T\hat{\mathcal{W}}_{c1} &\leq \frac{1}{2}(\hat{\mathcal{W}}_{c1}^T\phi_1)^2 + \frac{1}{2}(\tilde{\mathcal{W}}_{a1}^T\phi_1)^2.\end{aligned} \tag{53}$$

Moreover, according to $\tilde{\mathcal{W}}_{a1} = \hat{\mathcal{W}}_{a1} - \mathcal{W}_1^*$ and $\tilde{\mathcal{W}}_{c1} = \hat{\mathcal{W}}_{c1} - \mathcal{W}_1^*$, it is easy to get

$$\begin{aligned}\tilde{\mathcal{W}}_{a1}^T\phi_1\phi_1^T\hat{\mathcal{W}}_{a1} &= \frac{1}{2}(\tilde{\mathcal{W}}_{a1}^T\phi_1)^2 + \frac{1}{2}(\hat{\mathcal{W}}_{a1}^T\phi_1)^2 - \frac{1}{2}(\mathcal{W}_1^{*T}\phi_1)^2\\ \tilde{\mathcal{W}}_{c1}^T\phi_1\phi_1^T\hat{\mathcal{W}}_{c1} &= \frac{1}{2}(\tilde{\mathcal{W}}_{c1}^T\phi_1)^2 + \frac{1}{2}(\hat{\mathcal{W}}_{c1}^T\phi_1)^2 - \frac{1}{2}(\mathcal{W}_1^{*T}\phi_1)^2.\end{aligned} \tag{54}$$

Substituting (53) and (54) into (52) yields

$$\begin{aligned}\dot{\mathcal{V}}_1(t) \leq\ & -\left(k_1 - \frac{5}{4}\right)\hat{z}_1^2 - \left(\frac{\beta_{a1}}{2} - \frac{1}{4}\right)(\hat{\mathcal{W}}_{a1}^T\phi_1)^2 - \left(\beta_{c1}\right.\\ & \left. - \frac{\beta_{a1}}{2}\right)(\hat{\mathcal{W}}_{c1}^T\phi_1)^2 - \frac{\beta_{c1}}{2}(\tilde{\mathcal{W}}_{a1}^T\phi_1)^2 - \frac{\beta_{c1}}{2}(\tilde{\mathcal{W}}_{c1}^T\phi_1)^2\\ & + \frac{\beta_{a1} + \beta_{c1}}{2}(\mathcal{W}_1^{*T}\phi_1)^2 + \frac{1}{2}\hbar_2^2 + \frac{1}{2}\hat{z}_2^2.\end{aligned} \tag{55}$$

Step i $(i = 2, \ldots, n-1)$*:* Invoking (26) and (34), the estimated error dynamics of the subsystem i is

$$\dot{\hat{z}}_i = \hat{x}_{i+1} - \delta_2\hat{x}_i + \hat{\varpi}_i^T\varphi_i(\underline{\hat{x}}_i(t)) - \dot{\hat{\alpha}}_i. \tag{56}$$

Define the optimal cost function as

$$\mathcal{J}_i^*(\hat{z}_i(t)) = \min_{\alpha_{i-1}\in\Omega}\int_t^\infty(\hat{z}_i^2(\tau) + \dot{\hat{z}}_i^2(\tau))d\tau. \tag{57}$$

Based on the discussions in Section III, we can obtain $\alpha_i^* = \hat{x}_{i+1}$, and α_i^* is optimal virtual input. According to (56) and (57), the HJB equation is derived as

$$\begin{aligned}\mathcal{H}^*\left(\hat{z}_i, \alpha_{i-1}^*, \frac{\partial\mathcal{J}_i^*}{\partial\hat{z}_i}\right) &= \hat{z}_i^2 + \dot{\hat{z}}_i^2 + \frac{\partial\mathcal{J}_i^*}{\partial\hat{z}_i}\{\alpha_i^* - \delta_i\hat{x}_i\\ &\quad + \hat{\varpi}_i^T\varphi_i(\underline{\hat{x}}_i(t)) - \dot{\hat{\alpha}}_i\} = 0.\end{aligned} \tag{58}$$

Let $\partial\mathcal{H}(\hat{z}_i, \alpha_{i-1}, \partial\mathcal{J}_i^*/\partial\hat{z}_i)/\partial\alpha_{i-1} = 0$, the optimal control is

$$\alpha_{i-1}^*(\hat{z}_i) = -\frac{1}{2}\frac{\partial\mathcal{J}_i^*}{\partial\hat{z}_i} + \delta_i\hat{x}_i - \hat{\varpi}_i^T\varphi_i(\underline{\hat{x}}_i(t)) + \dot{\hat{\alpha}}_i. \tag{59}$$

We can write $\partial\mathcal{J}_i^*/\partial\hat{z}_i = 2k_i\hat{z}_i + \mathcal{J}_i^0(\hat{z}_i)$ and $\mathcal{J}_i^0 = -2k_i\hat{z}_i + \partial\mathcal{J}_i^*/\partial\hat{z}_i$, where $k_i > 0$ is designed constant. Then, (59) can be rewritten as

$$\alpha_i^* = -k_i\hat{z}_i - \frac{1}{2}\mathcal{J}_i^0(\hat{z}_i) + \delta_i\hat{x}_i - \hat{\varpi}_i^T\varphi_i(\underline{\hat{x}}_i(t)) + \dot{\hat{\alpha}}_i. \tag{60}$$

Using an FLS to approximate the nonlinear function $\mathcal{J}_i^0$, there is

$$\mathcal{J}_i^0(\hat{z}_i) = \mathcal{W}_i^{*T}\phi_i(\hat{z}_i) + \varepsilon_i(\hat{z}_i) \tag{61}$$

$\phi_i(\hat{z}_i)$ and $\varepsilon_i(\hat{z}_i)$ are subsequently abbreviated as ϕ_i and ε_i, respectively. The controller and gradient of cost function are trained using actor FLS and critic FLS respectively, thus

$$\hat{\alpha}_i = -k_i\hat{z}_i - \frac{1}{2}\hat{\mathcal{W}}_{ai}^T\phi_i(\hat{z}_i) + \delta_i\hat{x}_i - \hat{\varpi}_i^T\varphi_i(\underline{\hat{x}}_i(t)) + \dot{\hat{\alpha}}_i \tag{62}$$

$$\frac{\partial\hat{\mathcal{J}}_i}{\partial\hat{z}_i} = 2k_i\hat{z}_i + \hat{\mathcal{W}}_{ci}^T\phi_i(\hat{z}_i). \tag{63}$$

Remark 6: It is worth noting that the optimal virtual controller in this article is $\alpha_i^* = -k_i\hat{z}_i - 1/2\mathcal{J}_i^0(\hat{z}_i) - f(\underline{x}_i(t)) + \dot{\hat{\alpha}}_i$, while the optimal virtual controller in [20] and[21] is $\alpha_i^* = -k_1\hat{z}_i - 1/2\mathcal{J}_1^0(\hat{z}_i, \underline{x}_i(t)) - f(\underline{x}_i(t))$. Although the form is similar, the optimal virtual controller in [20] and[21] is made to contain $f(\underline{x}_i(t))$ by constructing a complex $\mathcal{J}_i^0(\hat{z}_i, \underline{x}_i(t))$. Therefore, the function $\mathcal{J}_i^0(\hat{z}_i, \underline{x}_i(t))$ being approximated is more complex when using the FLSs approximation. In addition the input of the FLSs is $[z_i, \underline{x}_i]^T$, and as the dimensionality

Authorized licensed use limited to: TIANJIN UNIVERSITY. Downloaded on November 21,2023 at 11:34:39 UTC from IEEE Xplore. Restrictions apply.

of the system states increase, the input of the FLSs will have dimensional pressure and it is more difficult to approximate $\mathcal{J}_i^0(\hat{z}_i, \underline{x}_i(t))$. In this article, the $\mathcal{J}_i^0(\hat{z}_i)$ approximated by the FLSs is a nonlinear function with respect to $\hat{z}_i$ only, so the approximation is better. In addition, optimizing the cost function in this article allows the tracking error to converge quickly and its jitters to be suppressed. Since there is error and its related terms in the controller, the jitters of input are also suppressed.

Similar to Step 1, the weight update laws of actor-critic FLSs can be obtained as

$$\begin{aligned}\dot{\hat{\mathcal{W}}}_{ai} &= -\phi_i\phi_i^T\{\beta_{ai}(\hat{\mathcal{W}}_{ai}-\hat{\mathcal{W}}_{ci})+\beta_{ci}\hat{\mathcal{W}}_{ci}\}\\ \dot{\hat{\mathcal{W}}}_{ci} &= -\beta_{ci}\phi_i\phi_i^T\hat{\mathcal{W}}_{ci}.\end{aligned} \tag{64}$$

Define the Lyapunov function as

$$\mathcal{V}_i(t) = \mathcal{V}_{i-1} + \frac{1}{2}\hat{z}_i^2 + \frac{1}{2}\tilde{\mathcal{W}}_{ai}^T\tilde{\mathcal{W}}_{ai} + \frac{1}{2}\tilde{\mathcal{W}}_{ci}^T\tilde{\mathcal{W}}_{ci} \tag{65}$$

where $\tilde{\mathcal{W}}_{ai} = \hat{\mathcal{W}}_{ai} - \mathcal{W}_i^*$ and $\tilde{\mathcal{W}}_{ci} = \hat{\mathcal{W}}_{ci} - \mathcal{W}_i^*$, thus $\dot{\tilde{\mathcal{W}}}_{ci} = \dot{\hat{\mathcal{W}}}_{ci}$ and $\dot{\tilde{\mathcal{W}}}_{ai} = \dot{\hat{\mathcal{W}}}_{ai}$. Using (34), (56), (62), (64), and (65), there is

$$\begin{aligned}\dot{\mathcal{V}}_i(t) = \dot{\mathcal{V}}_{i-1} &+ \hat{z}_i\left\{\hat{z}_{i+1}+\hbar_{i+1}-k_i\hat{z}_i-\frac{1}{2}\hat{\mathcal{W}}_{ai}^T\phi_i\right\}\\ &+(\beta_{ai}-\beta_{ci})\tilde{\mathcal{W}}_{ai}^T\phi_i\phi_i^T\hat{\mathcal{W}}_{ci}-\beta_{ci}\tilde{\mathcal{W}}_{ci}^T\phi_i\phi_i^T\hat{\mathcal{W}}_{ci}\\ &-\beta_{ai}\tilde{\mathcal{W}}_{ai}^T\phi_i\phi_i^T\hat{\mathcal{W}}_{ai}.\end{aligned} \tag{66}$$

After the same operations as (52)–(55), we get

$$\begin{aligned}\dot{\mathcal{V}}_i(t) \le \dot{\mathcal{V}}_{i-1} &- \left(k_i-\frac{5}{4}\right)\hat{z}_i^2 - \left(\frac{\beta_{ai}}{2}-\frac{1}{4}\right)(\hat{\mathcal{W}}_{ai}^T\phi_i)^2 - \Bigg(\beta_{ci}\\ &-\frac{\beta_{ai}}{2}\Bigg)(\hat{\mathcal{W}}_{ci}^T\phi_i)^2 - \frac{\beta_{ci}}{2}(\tilde{\mathcal{W}}_{ai}^T\phi_i)^2 - \frac{\beta_{ci}}{2}(\tilde{\mathcal{W}}_{ci}^T\phi_i)^2\\ &+\frac{\beta_{ai}+\beta_{ci}}{2}(\mathcal{W}_i^{*T}\phi_i)^2 + \frac{1}{2}\hbar_{i+1}^2 + \frac{1}{2}\hat{z}_{i+1}^2.\end{aligned} \tag{67}$$

It is worth noting that $1/2\hat{z}_i^2$ is included in $\dot{\mathcal{V}}_{i-1}$ so that $-(k_i - 7/4)\hat{z}_i^2$ exists in (67). Thus, (79) is written as $-(k_i - 7/4)\hat{z}_i^2$ to ensure the correctness of the second inequality.

Step n: Combining (26) and (34), the estimated error dynamics of subsystem n can be obtained as

$$\dot{\hat{z}}_n = u - \delta_n\hat{x}_n(t) + \hat{\varpi}_n^T\varphi_n(\underline{\hat{x}}_n(t)) - \dot{\hat{\alpha}}_n. \tag{68}$$

The optimal cost function is

$$\mathcal{J}_n^*(\hat{z}_n(t)) = \min_{u\in\Omega}\int_t^\infty(\hat{z}_n^2(\tau)+\dot{\hat{z}}_n^2(\tau))d\tau. \tag{69}$$

Invoking (68) and (69), the HJB equation can be obtained as

$$\begin{aligned}\mathcal{H}^*\left(\hat{z}_n, u^*, \frac{\partial\mathcal{J}_n^*}{\partial\hat{z}_n}\right) = \hat{z}_n^2 + \dot{\hat{z}}_n^2 + \frac{\partial\mathcal{J}_n^*}{\partial\hat{z}_n}\{u^* - \delta_n\hat{x}_n\\ + \hat{\varpi}_n^T\varphi_n(\underline{x}_n(t)) - \dot{\hat{\alpha}}_n\} = 0.\end{aligned} \tag{70}$$

Let $\partial\mathcal{H}(\hat{z}_n, u, \partial\mathcal{J}_n^*/\partial\hat{z}_n)/\partial u = 0$, the optimal control is

$$u^* = -\frac{1}{2}\frac{\partial\mathcal{J}_n^*}{\partial\hat{z}_n} + \delta_n\hat{x}_n - \hat{\varpi}_n^T\varphi_n(\underline{x}_n(t)) + \dot{\hat{\alpha}}_n. \tag{71}$$

Let $\partial\mathcal{J}_n^*/\partial\hat{z}_n = 2k_n\hat{z}_n + \mathcal{J}_n^0(\hat{z}_n)$ and $\mathcal{J}_n^0 = -2k_n\hat{z}_n + \partial\mathcal{J}_n^*/\partial\hat{z}_n$, where $k_n > 0$ is designed constant. Then, (85) can be rewritten as

$$u^* = -k_n\hat{z}_n - \frac{1}{2}\mathcal{J}_n^0(\hat{z}_n) + \delta_n\hat{x}_n - \hat{\varpi}_n^T\varphi_n(\underline{x}_n(t)) + \dot{\hat{\alpha}}_n. \tag{72}$$

Using an FLS to approximate $\mathcal{J}_n^0$, one has

$$\mathcal{J}_n^0(\hat{z}_n) = \mathcal{W}_n^{*T}\phi_n(\hat{z}_n) + \varepsilon_n(\hat{z}_n) \tag{73}$$

$\phi_n(\hat{z}_n)$ and $\varepsilon_n(\hat{z}_n)$ are subsequently abbreviated as ϕ_n and ε_n, respectively. The controller u^* and gradient of cost function $\partial\mathcal{J}_n^*/\partial\hat{z}_n$ are trained using actor FLS and critic FLS, respectively, thus

$$\hat{u} = -k_n\hat{z}_n - \frac{1}{2}\hat{\mathcal{W}}_{an}^T\phi_n + \delta_n\hat{x}_n - \hat{\varpi}_n^T\varphi_n + \dot{\hat{\alpha}}_n \tag{74}$$

$$\frac{\partial\hat{\mathcal{J}}_n}{\partial\hat{z}_n} = 2k_n\hat{z}_n + \hat{\mathcal{W}}_{cn}^T\phi_n. \tag{75}$$

Similar to Step 1, the weight update laws of actor-critic FLSs can be obtained as

$$\begin{aligned}\dot{\hat{\mathcal{W}}}_{an} &= -\phi_n\phi_n^T\{\beta_{an}(\hat{\mathcal{W}}_{an}-\hat{\mathcal{W}}_{cn})+\beta_{cn}\hat{\mathcal{W}}_{cn}\}\\ \dot{\hat{\mathcal{W}}}_{cn} &= -\beta_{cn}\phi_n\phi_n^T\hat{\mathcal{W}}_{cn}.\end{aligned} \tag{76}$$

Theorem 3: Consider Theorem 2 with the fuzzy observer (26), the control inputs (43), (62), and (74), the weight update laws (49), (64), and (76). If (80) is satisfied, then all closed-loop error signals are SGUUB.

Proof: Define the Lyapunov function as

$$\mathcal{V}_n(t) = \mathcal{V}_{n-1} + \frac{1}{2}\hat{z}_n^2 + \frac{1}{2}\tilde{\mathcal{W}}_{an}^T\tilde{\mathcal{W}}_{an} + \frac{1}{2}\tilde{\mathcal{W}}_{cn}^T\tilde{\mathcal{W}}_{cn} \tag{77}$$

where $\tilde{\mathcal{W}}_{an} = \hat{\mathcal{W}}_{an} - \mathcal{W}_n^*$ and $\tilde{\mathcal{W}}_{cn} = \hat{\mathcal{W}}_{cn} - \mathcal{W}_n^*$, thus $\dot{\tilde{\mathcal{W}}}_{cn} = \dot{\hat{\mathcal{W}}}_{cn}$ and $\dot{\tilde{\mathcal{W}}}_{an} = \dot{\hat{\mathcal{W}}}_{an}$. Using (68), (74), (76), and (77), one has

$$\begin{aligned}\dot{\mathcal{V}}_n(t) = \dot{\mathcal{V}}_{n-1} &+ \hat{z}_n\left\{-k_i\hat{z}_n-\frac{1}{2}\hat{\mathcal{W}}_{an}^T\phi_n\right\}\\ &-\beta_{an}\tilde{\mathcal{W}}_{an}^T\phi_n\phi_n^T\hat{\mathcal{W}}_{an}+(\beta_{an}-\beta_{cn})\tilde{\mathcal{W}}_{an}^T\phi_n\phi_n^T\hat{\mathcal{W}}_{cn}\\ &-\beta_{ci}\tilde{\mathcal{W}}_{cn}^T\phi_n\phi_n^T\hat{\mathcal{W}}_{cn}.\end{aligned} \tag{78}$$

After the same operations as (52)–(55), combining (55) and (67) yields

$$\begin{aligned}\dot{\mathcal{V}}_n(t) \le \dot{\mathcal{V}}_{n-1} &- \left(k_i-\frac{1}{4}\right)\hat{z}_n^2\\ &-\left(\frac{\beta_{an}}{2}-\frac{1}{4}\right)(\hat{\mathcal{W}}_{an}^T\phi_n)^2 - \Bigg(\beta_{cn}\\ &-\frac{\beta_{an}}{2}\Bigg)(\hat{\mathcal{W}}_{cn}^T\phi_n)^2 - \frac{\beta_{cn}}{2}(\tilde{\mathcal{W}}_{an}^T\phi_n)^2\\ &-\frac{\beta_{cn}}{2}(\tilde{\mathcal{W}}_{cn}^T\phi_n)^2 + \frac{\beta_{an}+\beta_{cn}}{2}(\mathcal{W}_n^{*T}\phi_n)^2\\ \le &-\sum_{i=1}^n\left(k_i-\frac{7}{4}\right)\hat{z}_i^2 - \sum_{i=1}^n\left(\frac{\beta_{ai}}{2}-\frac{1}{4}\right)(\hat{\mathcal{W}}_{ai}^T\phi_i)^2\end{aligned}$$

Authorized licensed use limited to: TIANJIN UNIVERSITY. Downloaded on November 21,2023 at 11:34:39 UTC from IEEE Xplore. Restrictions apply.

TABLE I
SYMBOLS AND VALUES OF SINGLE-LINK ROBOT SYSTEM

Symbols	Definitions	Values
$\mathcal{M}$	Moment of inertia	1 kg·m^2
ℓ	Length of the link	1 m
g	Acceleration due to gravity	9.8 m/s^2
m	Mass	1 kg
$\mathcal{B}$	Damping coefficient	1 N·m·s

$$-\sum_{i=1}^{n}\left(\beta_{ci}-\frac{\beta_{ai}}{2}\right)(\hat{\mathcal{W}}_{ci}^{T}\phi_i)^2-\sum_{i=1}^{n}\frac{\beta_{ci}}{2}(\tilde{\mathcal{W}}_{ai}^{T}\phi_i)^2$$
$$-\sum_{i=1}^{n}\frac{\beta_{ci}}{2}(\tilde{\mathcal{W}}_{ci}^{T}\phi_i)^2+\sum_{i=1}^{i}\frac{\beta_{ai}+\beta_{ci}}{2}(\mathcal{W}_i^{*T}\phi_i)^2$$
$$+\sum_{i=2}^{n}\frac{1}{2}\hbar_i^2. \tag{79}$$

If $k_i > 7/4, \beta_{ai}/2 > 1/4, \beta_{ci}-\beta_{ai}/2 > 0, \beta_{ai} > 0, \beta_{ci} > 0$, i.e.,

$$k_i > \frac{7}{4}, \quad \beta_{ci} > \beta_{ai}/2 > \frac{1}{4} \tag{80}$$

are satisfied, then

$$\dot{\mathcal{V}}_n \le -\mathcal{A}_n\mathcal{V}_n + \mathcal{B}_n \tag{81}$$

where $\mathcal{A}_n = \min\{2k_i - 7/2, \beta_{ci}\lambda_{\min}(\phi_i\phi_i^T)\}$, and $\mathcal{B}_n = \sum_{i=1}^{i}(\beta_{ai}+\beta_{ci})(\mathcal{W}_i^{*T}\phi_i)^2/2 + \sum_{i=2}^{n}\hbar_i^2/2$. According to Lemma 1, it is clear that both ϕ_i and $\mathcal{W}_i^*$ are bounded. And according to [23] and [42], the filtered error $\hbar_i$ is also considered to be bounded. Therefore, $\mathcal{B}_n$ is bounded. The proof of Theorem 3 is completed. ■

According to (81), a smaller $\mathcal{B}_n$ and a larger $\mathcal{A}_n$ will lead to the smaller tracking errors and the weight errors. Therefore, choosing larger k_i and β_{ci} or smaller β_{ai} can make the errors smaller.

V. SIMULATION

Example 1: Consider the following single-link robot model [43], [44] as

$$\mathcal{M}\ddot{\theta} + \frac{1}{2}mg\ell\sin\theta + \mathcal{B}\dot{\theta} = u, \quad y = \theta \tag{82}$$

where u is input torque, θ is the angle, $\dot{\theta}$ is the angular velocity, and $\ddot{\theta}$ is the angular acceleration. The specific definitions and values are shown in Table I .

Let $x_1 = \theta$ and $x_2 = \dot{\theta}$, the system can be described as

$$\begin{cases}\dot{x}_1(t) = x_2\\ \dot{x}_2(t) = -4.9\sin(x_1) - x_2 + u\\ y(t) = x_1(t).\end{cases} \tag{83}$$

The actual initial states of the system are $x_1(0) = 0.2$ and $x_2(0) = -0.5$, respectively. The reference trajectory is $x_r = -0.4e^{-0.3t} + 0.6\sin(0.5t) + 0.5\cos(0.7t)$.

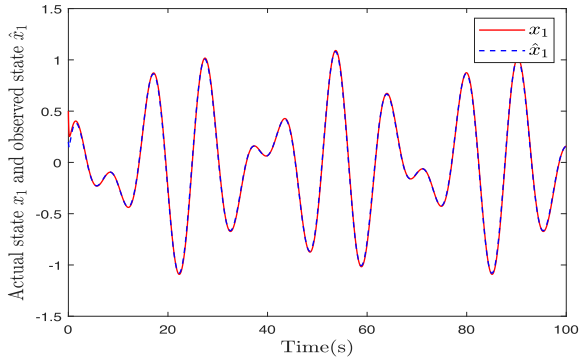

Fig. 2. System output y_1 and reference x_r.

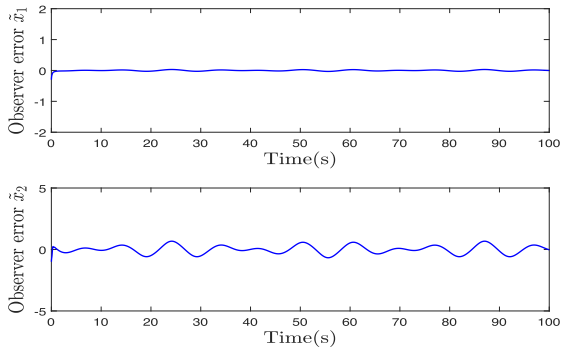

Fig. 3. State errors $\tilde{x}_1$ and $\tilde{x}_2$ of the observer.

Case 1: Verify the validity of the proposed algorithm.

1) *The fuzzy state observer:* The initial states of the observer are $\hat{x}_1(0) = 0.5$ and $\hat{x}_1(0) = 0.5$, respectively. We choose $\delta_1 = 20, \delta_2 = 18, \bar{\gamma}_1 = 1.4$, and $\bar{\gamma}_2 = 4.2$. In addition, the fuzzy membership functions of FLSs used to approximate f_1 and $\mathcal{F}_2$ were all selected as
$$\mu_{\mathcal{A}_i^j}(\hat{x}_i) = e^{(\frac{x_i+5-j}{4})}, j = 1, 2, \ldots, 10.$$
The initial weight vectors are $\hat{\varpi}_1(0) = 0.6 * \text{ones}(10, 1)$ and $\hat{\varpi}_2(0) = 0.6 * \text{ones}(10, 1)$, respectively.
2) *The improved optimized backstepping:* The parameters are $k_1 = 6, k_2 = 6, \beta_{a1} = 4, \beta_{c1} = 5, \beta_{a2} = 4, \beta_{c2} = 5$, and $\kappa = 0.02$. The two FLSs used for the acotr-critic structure are identical. The fuzzy membership functions can be chosen as
$$\mu_{\mathcal{A}_i^j}(\hat{x}_i) = e^{(\frac{x_i+5-j}{4})}, j = 1, 2, \ldots, 10.$$

In Step 1, the initial weight vectors are $\hat{\mathcal{W}}_{a1}(0) = \text{ones}(10, 1)$ and $\hat{\mathcal{W}}_{c1}(0) = \text{ones}(10, 1)$, respectively. In Step 2, the FLSs used are exactly the same as in Step 1. The initial weight vectors of Step 2 are $\hat{\mathcal{W}}_{a2}(0) = \text{ones}(10, 1)$ and $\hat{\mathcal{W}}_{c2}(0) = \text{ones}(10, 1)$, respectively.

The simulation results are shown in Figs. 2–7. Fig. 2 shows the output tracking effect of the system, Fig. 3 shows the state error of the observer, and Fig. 4 shows the actual tracking error and the actual input. Figs. 5–7 show the weights of the FLSs used by the observer and RL.

Authorized licensed use limited to: TIANJIN UNIVERSITY. Downloaded on November 21,2023 at 11:34:39 UTC from IEEE Xplore. Restrictions apply.

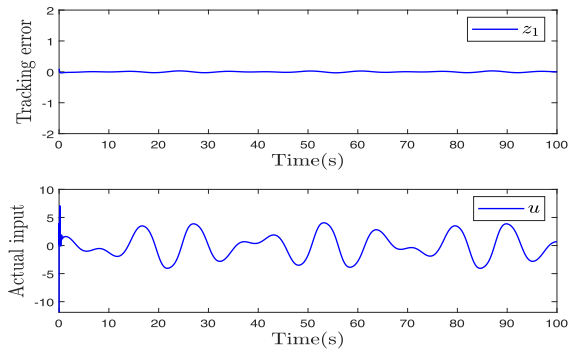

Fig. 4. Tracking error and actual input.

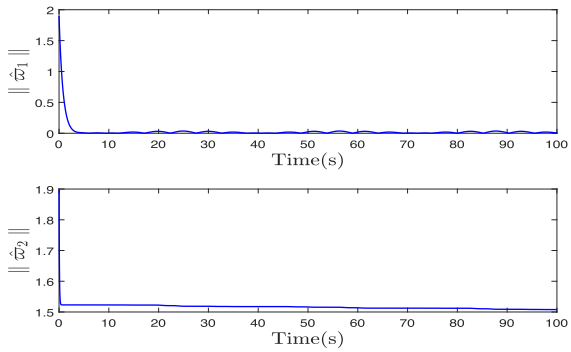

Fig. 5. Weights of the fuzzy observer.

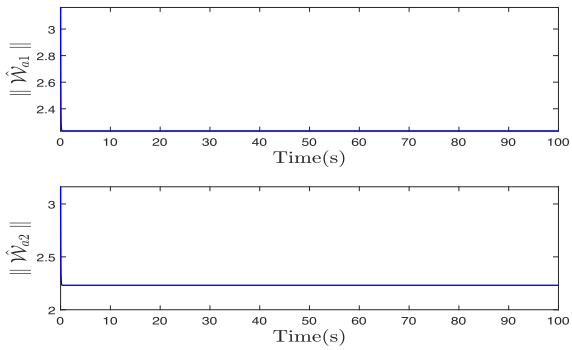

Fig. 6. Weights of actor FLSs.

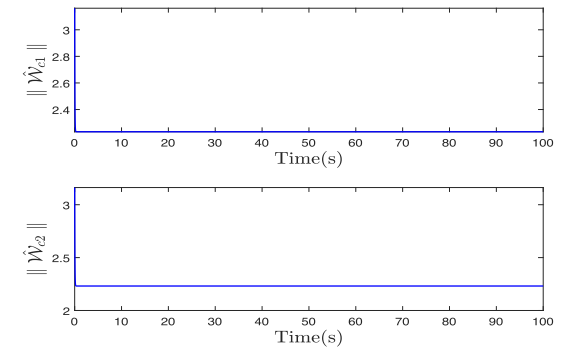

Fig. 7. Weights of critic FLSs.

TABLE II
PARAMETERS OF OTHER METHODS

Method [39]	Values	Traditional method	Values
k_1	40	k_1	40
k_2	30	k_2	30
β_{a1}	3.2	σ_1	4
β_{c1}	3.4	σ_2	4
β_{a2}	2.6	Γ_1	$2I_{10}$
β_{c2}	2.8	Γ_2	$2I_{10}$

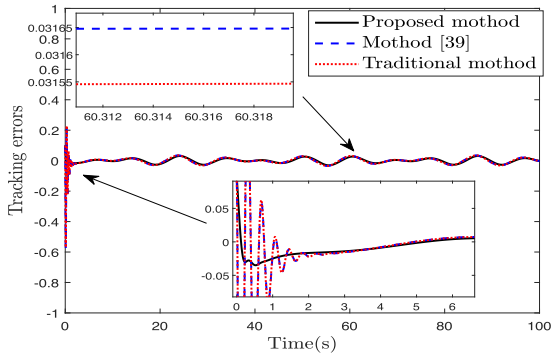

Fig. 8. Comparison of tracking errors for three different methods.

Case 2: Verify the advantages of the proposed algorithm by comparing different methods.

In this case, to verify the advantages of the algorithm, we will compare it with the existing optimized backstepping method [39] and the traditional backstepping method (the fuzzy state observers used and their parameters are identical). In addition, to ensure the fairness of the comparison, the sampling steps used are all $\Delta T = 0.01$ and the total simulation time is 100 s. The FLSs used are all ten-fuzzy rule FLSs and all other parameters are shown in Table II. (The parameters used in method [39] refer to the original text and the parameters used in the proposed algorithm are the same as in Case 1.) The controllers of optimized backstepping method [39] are

$$\alpha_1 = -k_1\hat{z}_1 - \frac{1}{2}\hat{\mathcal{W}}_{a1}^T\phi_1(\hat{z}_1), \quad u = -k_2\hat{z}_2 - \frac{1}{2}\hat{\mathcal{W}}_{a2}^T\phi_2(\hat{z}_2) \tag{84}$$

and the weight adaptive laws are $\dot{\hat{\mathcal{W}}}_{ai} = -\beta_{ai}\phi_i\phi_i^T(\hat{\mathcal{W}}_{ai} - \hat{\mathcal{W}}_{ci})$ and $\dot{\hat{\mathcal{W}}}_{ci} = -1/2\phi_i\hat{z}_i - \beta_{ci}\phi_i\phi_i^T\hat{\mathcal{W}}_{ci}, i = 1, 2$. The initial weight vectors are $\hat{\mathcal{W}}_{ai}(0) = \hat{\mathcal{W}}_{ci}(0) = \text{ones}(10, 1)$.

The controllers of traditional backstepping method are

$$\alpha_1 = -k_1\hat{z}_1 - \hat{\mathcal{W}}_1^T\phi_1(\mathcal{X}_1), \quad u = -k_2\hat{z}_2 - \hat{\mathcal{W}}_2^T\phi_2(\mathcal{X}_2). \tag{85}$$

The FLSs inputs are $\mathcal{X}_1 = [\hat{z}_1; \dot{x}_r]$ and $\mathcal{X}_2 = [\hat{z}_2; \dot{\alpha}_1]$, the weight adaptive laws are $\dot{\hat{\mathcal{W}}}_i = \Gamma_i(\phi_i\hat{z}_i - \sigma_i\hat{\mathcal{W}}_i)$ and the initial weight vectors are $\hat{\mathcal{W}}_i(0) = \text{ones}(10, 1)$, where $i = 1, 2$. In addition, two error performance criteria, integral of absolute error (IAE) and time integral of absolute error (TIAE), are given as

$$\text{IAE} = \int_0^{t_f} \mid z_1(t) \mid dt, \quad \text{TIAE} = \int_0^{t_f} t \mid z_1(t) \mid dt \tag{86}$$

Authorized licensed use limited to: TIANJIN UNIVERSITY. Downloaded on November 21,2023 at 11:34:39 UTC from IEEE Xplore. Restrictions apply.

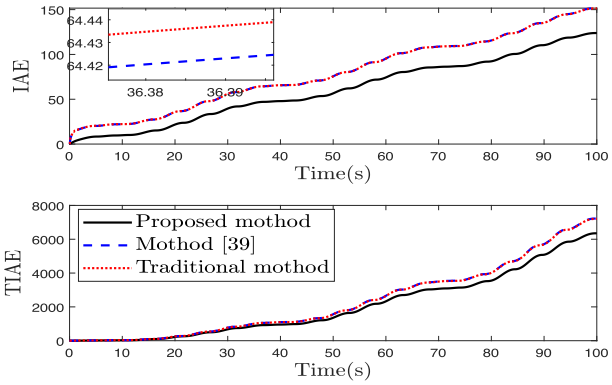

Fig. 9. Comparison for IAE and TIAE of three different methods.

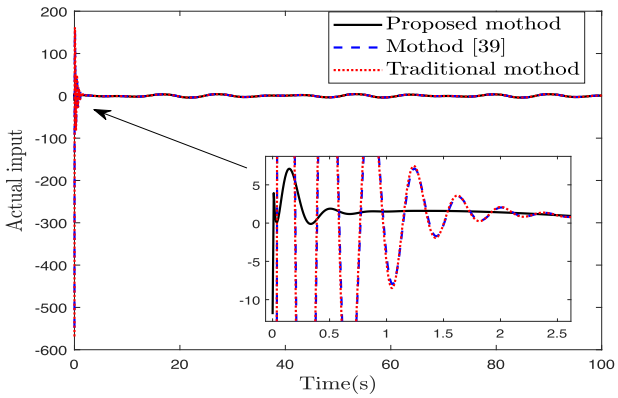

Fig. 10. Comparison for the inputs of three different methods.

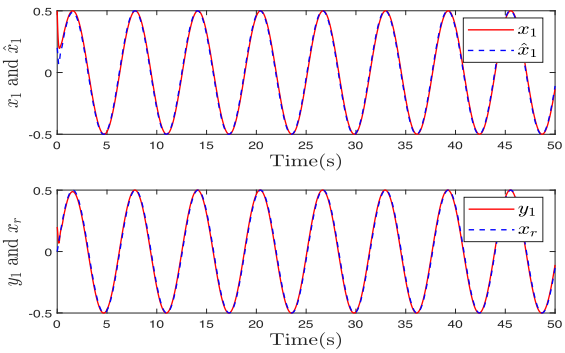

Fig. 11. System state x_1, observed state $\hat{x}_1$, system output y_1, and reference trajectory x_r.

where t_f is final time of the simulation. If the two criteria are smaller, it means that the tracking error and long-term tracking error are smaller.

Remark 7: Equations (84) and (85) appear to be formally cleaner than the proposed algorithm in this article, however, these controllers are designed based on the same observer. That is, all three methods use the state observer, the proposed algorithm uses the structural information of the observer, while the other algorithms do not. The structural information

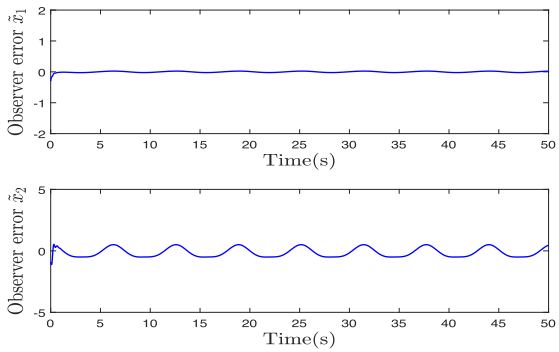

Fig. 12. State errors $\tilde{x}_1$ and $\tilde{x}_2$ of the observer.

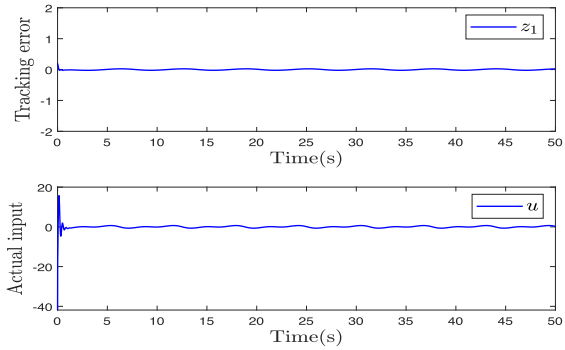

Fig. 13. Tracking error and actual input.

$\delta_1\tilde{x}_1(t) + \hat{\varpi}_1^T\varphi_1(\underline{x}_1(t))$ is already computed in the observer part, so this does not introduce excessive computational effort.

Fig. 8 shows a comparison of the tracking errors for the three methods. Since the proposed algorithm uses a cost function based on the error derivative, it has a significant effect on the optimization for the error and its rate of change. Because the errors are optimized, overshoot and jitters are reduced. And all three controllers have error terms $-k_1\hat{z}_1$, thus jitters and overshoot of the control inputs are also reduced. This can be seen in Fig. 10. The smaller values of IAE and TIAE indicate better tracking error performance, and it can be seen from Fig. 9 that the proposed algorithm has a better tracking performance.

Example 2: Consider the following numerical example:

$$\begin{cases} \dot{x}_1(t) = x_2 + x_1^2\sin(x_1x_2) \\ \dot{x}_2(t) = u + x_1x_2\cos^2(x_1) \end{cases} \tag{87}$$

$y_1(t) = x_1(t)$ and the reference trajectory is $x_r = 0.5\sin(t)$. For simplicity, all other parameters are chosen exactly as in *Case 1* in *Example 1*. The simulation results are shown in Figs. 11–13. It can be seen that the proposed algorithm still has good results for a complex system.

VI. Conclusion

In this article, a novel optimized adaptive control method based on subsystems error derivatives is proposed for strict-feedback systems with unmeasurable states. A cost function based on the error derivative is used and its advantages are

Authorized licensed use limited to: TIANJIN UNIVERSITY. Downloaded on November 21,2023 at 11:34:39 UTC from IEEE Xplore. Restrictions apply.

demonstrated in detail. It solves two problems caused by the classical cost functions. Considering the case where the states are unmeasurable, a fuzzy state observer is designed that avoids the requirement that the design gain constants satisfy the Hurwitz equation. Based on RL, the observer, and error derivative cost function, an improved optimized control algorithm is given. Simulation results show that the algorithm can significantly reduce tracking errors and suppress overshoot and jitters. Our future work will focus on using the improved cost function to study switched systems or systems with failures.

REFERENCES

[1] R. Bellman, "Dynamic programming," *Science*, vol. 153, no. 3731, pp. 34–37, 1966.

[2] D. Wang, J. Qiao, and L. Cheng, "An approximate neuro-optimal solution of discounted guaranteed cost control design," *IEEE Trans. Cybern.*, vol. 52, no. 1, pp. 77–86, Jan. 2022.

[3] B. Luo, Y. Yang, D. Liu, and H.-N. Wu, "Event-triggered optimal control with performance guarantees using adaptive dynamic programming," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 1, pp. 76–88, Jan. 2020.

[4] L. An and G.-H. Yang, "Optimal transmission power scheduling of networked control systems via fuzzy adaptive dynamic programming," *IEEE Trans. Fuzzy Syst.*, vol. 29, no. 6, pp. 1629–1639, Jun. 2021.

[5] L. Kong, W. He, C. Yang, and C. Sun, "Robust neurooptimal control for a robot via adaptive dynamic programming," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 6, pp. 2584–2594, Jun. 2021.

[6] N. Wang, Y. Gao, and X. Zhang, "Data-driven performance-prescribed reinforcement learning control of an unmanned surface vehicle," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 12, pp. 5456–5467, Dec. 2021.

[7] G. Wen, C. L. P. Chen, J. Feng, and N. Zhou, "Optimized multi-agent formation control based on an identifier cactor ccritic reinforcement learning algorithm," *IEEE Trans. Fuzzy Syst.*, vol. 26, no. 5, pp. 2719–2731, May 2018.

[8] Q. Wei, Z. Liao, Z. Yang, B. Li, and D. Liu, "Continuous-time time-varying policy iteration," *IEEE Trans. Cybern.*, vol. 50, no. 12, pp. 4958–4971, Dec. 2020.

[9] H. Zhang, H. Su, K. Zhang, and Y. Luo, "Event-triggered adaptive dynamic programming for non-zero-sum games of unknown nonlinear systems via generalized fuzzy hyperbolic models," *IEEE Trans. Fuzzy Syst.*, vol. 27, no. 11, pp. 2202–2214, Nov. 2019.

[10] Y. Yang, W. Gao, H. Modares, and C.-Z. Xu, "Robust actor-critic learning for continuous-time nonlinear systems with unmodeled dynamics," *IEEE Trans. Fuzzy Syst.*, vol. 30, no. 6, pp. 2101–2112, Jun. 2022.

[11] W. Gao, Y. Jiang, Z.-P. Jiang, and T. Chai, "Output-feedback adaptive optimal control of interconnected systems based on robust adaptive dynamic programming," *Automatica*, vol. 72, pp. 37–45, 2016.

[12] Z. Peng, R. Luo, J. Hu, K. Shi, S. K. Nguang, and B. K. Ghosh, "Optimal tracking control of nonlinear multiagent systems using internal reinforce Q-learning," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 8, pp. 4043–4055, Aug. 2022.

[13] L. Ding, S. Li, H. Gao, Y.-J. Liu, L. Huang, and Z. Deng, "Adaptive neural network-based finite-time online optimal tracking control of the nonlinear system with dead zone," *IEEE Trans. Cybern.*, vol. 51, no. 1, pp. 382–392, Jan. 2021.

[14] T. Wang, Y. Wang, X. Yang, and J. Yang, "Further results on optimal tracking control for nonlinear systems with nonzero equilibrium via adaptive dynamic programming," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Aug. 24, 2021, doi: 10.1109/TNNLS.2021.3105646.

[15] Y. Fu, C. Hong, J. Fu, and T. Chai, "Approximate optimal tracking control of nondifferentiable signals for a class of continuous-time nonlinear systems," *IEEE Trans. Cybern.*, vol. 52, no. 6, pp. 4441–4450, Jun. 2022.

[16] F. L. Lewis, D. Vrabie, and V. L. Syrmos, *Optimal Control*, 3rd ed. Hoboken, NJ, USA: Wiley, 2012.

[17] H. Modares and F. L. Lewis, "Optimal tracking control of nonlinear partially-unknown constrained-input systems using integral reinforcement learning," *Automatica*, vol. 50, no. 7, pp. 1780–1792, 2014.

[18] H. Modares, F. L. Lewis, and Z.-P. Jiang, "H_∞ tracking control of completely unknown continuous-time systems via off-policy reinforcement learning," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 10, pp. 2550–2562, Oct. 2015.

[19] C. Li, J. Ding, F. L. Lewis, and T. Chai, "A novel adaptive dynamic programming based on tracking error for nonlinear discrete-time systems," *Automatica*, vol. 129, 2021, Art. no. 109687.

[20] G. Wen, C. L. P. Chen, and S. S. Ge, "Simplified optimized backstepping control for a class of nonlinear strict-feedback systems with unknown dynamic functions," *IEEE Trans. Cybern.*, vol. 51, no. 9, pp. 4567–4580, Sep. 2021.

[21] Y. Li, Y. Fan, K. Li, W. Liu, and S. Tong, "Adaptive optimized backstepping control-based RL algorithm for stochastic nonlinear systems with state constraints and its application," *IEEE Trans. Cybern.*, vol. 52, no. 10, pp. 10542–10555, Oct. 2022.

[22] G. Wen, S. S. Ge, and F. Tu, "Optimized backstepping for tracking control of strict-feedback systems," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 8, pp. 3850–3862, Aug. 2018.

[23] S.-C. Tong, Y.-M. Li, G. Feng, and T.-S. Li, "Observer-based adaptive fuzzy backstepping dynamic surface control for a class of MIMO nonlinear systems," *IEEE Trans. Syst, Man, Cybern., B*, vol. 41, no. 4, pp. 1124–1135, Aug. 2011.

[24] L. Wang, H. Wang, P. X. Liu, S. Ling, and S. Liu, "Fuzzy finite-time command filtering output feedback control of nonlinear systems," *IEEE Trans. Fuzzy Syst.*, vol. 30, no. 1, pp. 97–107, Jan. 2022.

[25] S. Li, C. K. Ahn, and Z. Xiang, "Command-filter-based adaptive fuzzy finite-time control for switched nonlinear systems using state-dependent switching method," *IEEE Trans. Fuzzy Syst.*, vol. 29, no. 4, pp. 833–845, Apr. 2021.

[26] S. Li, C. K. Ahn, and Z. Xiang, "Sampled-data adaptive output feedback fuzzy stabilization for switched nonlinear systems with asynchronous switching," *IEEE Trans. Fuzzy Syst.*, vol. 27, no. 1, pp. 200–205, Jan. 2019.

[27] K. Li and Y. Li, "Adaptive NN optimal consensus fault-tolerant control for stochastic nonlinear multiagent systems," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Aug. 25, 2021, doi: 10.1109/TNNLS.2021.3104839.

[28] G. Wen and C. L. P. Chen, "Optimized backstepping consensus control using reinforcement learning for a class of nonlinear strict-feedback-dynamic multi-agent systems," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Aug. 30, 2021, doi: 10.1109/TNNLS.2021.3105548.

[29] G. Feng, "A survey on analysis and design of model-based fuzzy control systems," *IEEE Trans. Fuzzy Syst.*, vol. 14, no. 5, pp. 676–697, May 2006.

[30] L. Sun and W. Huo, "Adaptive fuzzy control of spacecraft proximity operations using hierarchical fuzzy systems," *IEEE/ASME Trans. Mechatronics*, vol. 21, no. 3, pp. 1629–1640, Mar. 2016.

[31] C. Wu, J. Liu, X. Jing, H. Li, and L. Wu, "Adaptive fuzzy control for nonlinear networked control systems," *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 47, no. 8, pp. 2420–2430, Aug. 2017.

[32] Y.-X. Li and G.-H. Yang, "Graph-theory-based decentralized adaptive output-feedback control for a class of nonlinear interconnected systems," *IEEE Trans. Cybern.*, vol. 49, no. 7, pp. 2444–2453, Jul. 2019.

[33] G. Liu, J. H. Park, H. Xu, and C. Hua, "Reduced-order observer-based output-feedback tracking control for nonlinear time-delay systems with global prescribed performance," *IEEE Trans. Cybern.*, early access, Mar. 25, 2022, doi: 10.1109/TCYB.2022.3158932.

[34] H. Wang, P. X. Liu, and P. Shi, "Observer-based fuzzy adaptive output-feedback control of stochastic nonlinear multiple time-delay systems," *IEEE Trans. Cybern.*, vol. 47, no. 9, pp. 2568–2578, Sep. 2017.

[35] W. Shi, "Adaptive fuzzy output-feedback control for nonaffine mimo nonlinear systems with prescribed performance," *IEEE Trans. Fuzzy Syst.*, vol. 29, no. 5, pp. 1107–1120, May 2021.

[36] Y. Cheng, J. Zhang, H. Du, G. Wen, and X. Lin, "Global event-triggered output feedback stabilization of a class of nonlinear systems," *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 51, no. 7, pp. 4040–4047, Jul. 2021.

[37] F. Li, Y. Shen, L. Wang, and Y.-W. Wang, "Consensus of upper-triangular multiagent systems with sampled and delayed measurements via output feedback," *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 50, no. 2, pp. 600–608, Feb. 2020.

[38] Y. Wei, Y. Wang, C. K. Ahn, and D. Duan, "IBLF-based finite-time adaptive fuzzy output-feedback control for uncertain MIMO nonlinear state-constrained systems," *IEEE Trans. Fuzzy Syst.*, vol. 29, no. 11, pp. 3389–3400, Nov. 2021.

[39] G. Wen, B. Li, and B. Niu, "Optimized backstepping control using reinforcement learning of observer-critic-actor architecture based on fuzzy system for a class of nonlinear strict-feedback systems," *IEEE Trans. Fuzzy Syst.*, vol. 30, no. 10, pp. 4322–4335, Oct. 2022.

[40] Y.-m. Li, K. Li, and S. Tong, "An observer-based fuzzy adaptive consensus control method for nonlinear multi-agent systems," *IEEE Trans. Fuzzy Syst.*, vol. 30, no. 11, pp. 4667–4678, Nov. 2022.

Authorized licensed use limited to: TIANJIN UNIVERSITY. Downloaded on November 21,2023 at 11:34:39 UTC from IEEE Xplore. Restrictions apply.

[41] S. Tong, X. Min, and Y. Li, "Observer-based adaptive fuzzy tracking control for strict-feedback nonlinear systems with unknown control gain functions," *IEEE Trans. Cybern.*, vol. 50, no. 9, pp. 3903–3913, Sep. 2020.
[42] L. Wang and J. Dong, "Adaptive fuzzy consensus tracking control for uncertain fractional-order multi-agent systems with event-triggered input," *IEEE Trans. Fuzzy Syst.*, vol. 30, no. 2, pp. 310–320, Feb. 2022.
[43] B. Liu, M. Hou, C. Wu, W. Wang, Z. Wu, and B. Huang, "Predefined-time backstepping control for a nonlinear strict-feedback system," *Int. J. Robust Nonlinear Control*, vol. 31, no. 8, pp. 3354–3372, 2021.
[44] Y.-J. Liu and S. Tong, "Barrier Lyapunov functions-based adaptive control for a class of nonlinear pure-feedback systems with full state constraints," *Automatica*, vol. 64, pp. 70–75, 2016.

Dongdong Li received the B.S. degree in automation from the Anhui University of Science and Technology, Huainan, China, in 2020. He is currently working toward the M.S. degree in control science and engineering with the College of Information Science and Engineering, Northeastern University, Shenyang, China.

His research interests include fuzzy adaptive control, reinforcement learning, distributed optimization, and cyber-physical systems.

Jiuxiang Dong (Member, IEEE) received the B.S. degree in mathematics and applied mathematics and the M.S. degree in applied mathematics from Liaoning Normal University, Dalian, China, in 2001 and 2004, respectively, and the Ph.D. degree in navigation guidance and control from Northeastern University, Shenyang, China, in 2009.

He is currently a Professor with the College of Information Science and Engineering, Northeastern University. His research interests include fuzzy control, robust control, and reliable control.

Dr. Dong is an Associate Editor for *International Journal of Control, Automation, and Systems*.

Authorized licensed use limited to: TIANJIN UNIVERSITY. Downloaded on November 21,2023 at 11:34:39 UTC from IEEE Xplore. Restrictions apply.