

Realtime Brain-Inspired Adaptive Learning Control for Nonlinear Systems With Configuration Uncertainties

Yanhui Zhang[✉], *Member, IEEE*, Yuxin Yang, Weifang Chen, *Member, IEEE*, and Hua Yang[✉]

Abstract—This paper investigates the problem of adaptive tracking control for quadcopter in the presence of nonlinear configuration uncertainties. It utilizes a real-time brain-inspired learning control (RbiLC) method to address the challenges posed by nonlinear time-varying uncertain instructions. To address the issue of flight control law reconfiguration caused by unknown changes in the fuselage configuration (e.g., propellers or motors), this paper introduces an online learning-evaluation-optimization reconstruction mechanism based on RbiLC. The proposed adaptive learning controller mitigates the need for extensive human resources and reduces the time required for flight controller design. The Lyapunov-Krasovskii function is introduced as a compensatory measure to address the impact of parameter uncertainty on system stability. Furthermore, this paper proposes a signed sinusoidal function perturbation estimate to guide the direction and magnitude throughout the online learning process. The approach conducts a theoretical stability analysis on a quadcopter vehicle considering uncertainties in UAV dynamics modeling. The results demonstrate that the proposed scheme achieves superior control and faster adaptation, enabling the system to ultimately converge to a compact set within a limited time domain. Finally, software-in-the-loop (SITL) simulations and flight verification results are presented to validate the proposed control strategy.

Note to Practitioners—In various UAV application scenarios, UAVs often necessitate multiple motor and size configurations to accommodate diverse mission load requirements. This paper examines the self-learning reconfiguration control of a nonlinear UAV controller, focusing on achieving stable control despite uncertain propeller configurations. The proposed self-learning control method is inspired by the self-learning mechanism observed in the human brain. Without any manual intervention, the traditional proportional-integral-derivative (PID) control serves as the benchmark and is intricately combined with an evolutionary strategy for iterative learning. The learning process is realized during flight without relying on a precise model through real-time evaluation, nonlinear command track-

ing response, and iterative reconstruction using a design-based symbolic sine function optimal controller. The practice has proved that when the UAV model cannot be identified in real-time, the UAV controller can be reconfigured in real-time through the model-free control method based on brain-inspired learning iterations to achieve stable control of the UAV. The notable feature of this paper is the establishment of the RbiLC framework, the completion of the theoretical proof of stability, and the real-world flight experiment on the quadcopter

Index Terms—Real-time adaptive learning control, nonlinear quadcopter control, configuration uncertainties.

I. INTRODUCTION

ADAPTIVE control has been extensively researched for linear system control tasks, utilizing historical response information to improve the control performance of the current controller [1]. However, nonlinear uncertain strict feedback systems are prevalent in practical engineering, as seen in cross-domain flight rocket control systems [2] and unmanned aerial vehicle (UAV) control systems [3], [4]. In such control systems, the real-time uncertain configuration of the plant models can increase control difficulty, leading to system oscillation or divergence. Therefore, this paper aims to study the real-time adaptive learning control problem of nonlinear strict-feedback systems with uncertain configurations to realize a point-by-point response to the desired command.

Achieving stable control of multirotor UAVs under different working conditions typically requires redesigning the UAV controllers with different loads and propeller sizes, which is a time-consuming and laborious task. For instance, in [5], a small aerial photography drone with a load of 200g is equipped with 3-inch propellers. While in [6], a swarm drone with a load of 400g is equipped with 6-inch propellers. The pesticide spraying drone has a load of 8kg and matches 20-inch propellers [7], and the unmanned aerial vehicle in [8] has a load of 220kg with a blade size of approximately 48 inches. Traditional methods for adapting flight controllers to such different aircraft require searching for the optimal controller based on years of controller design experience and long-term actual flight tests.

Most recent studies can be mainly categorized into two trends for the control system reconstruction problem. One of them combines model-base method and artificial experience to reconstruct the controller parameters offline, such as model reference adaptive control (MRAC) [9], fault-tolerant control systems (FTCS) [10], retrospective cost adaptive control

Manuscript received 7 April 2023; revised 6 June 2023; accepted 27 June 2023. Date of publication 7 August 2023; date of current version 8 August 2024. This article was recommended for publication by Associate Editor L. Zeng and Editor M. Dotoli upon evaluation of the reviewers' comments. This work was supported in part by the National Natural Science Foundation of China for the Near Space Research Program under Grant U20B2007 and Grant 12002306. (Corresponding author: Hua Yang.)

Yanhui Zhang is with the School of Aeronautics and Astronautics, Zhejiang University, Hangzhou 310027, China, and also with the Huanjiang Laboratory Advanced Aircraft Research Center, Zhuji, Zhejiang 311800, China (e-mail: anhuizhang91@gmail.com).

Yuxin Yang, Weifang Chen, and Hua Yang are with the School of Aeronautics and Astronautics, Zhejiang University, Hangzhou 310027, China (e-mail: yhsaa@zju.edu.cn).

Color versions of one or more figures in this article are available at <https://doi.org/10.1109/TASE.2023.3292170>.

Digital Object Identifier 10.1109/TASE.2023.3292170

1545-5955 © 2023 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See <https://www.ieee.org/publications/rights/index.html> for more information.

(RCAC) [11], model predictive control (MPC) [12] and H_∞ control [13].

The other trend in control is to employ a model-free approach to address these problems. Model-free control methods are designed to achieve control of a system without requiring the mathematical model or structural parameters of the plant. Instead, they rely on limited input and output response data to design closed-loop controllers. The earliest model-free control methods can be traced back to proportional-integral-derivative (PID) control [14], [15]. Since then, several model-free control approaches have been developed, including extremum seeking (ES) [16], Ziegler-Nichols-PID (ZN-PID) [17], PID-ES [18], model-free adaptation control (MFAC) [19], event-triggered adaptive control [20], [21], T-S Fuzzy [22] and iterative learning control (ILC) [23]. Our previous work has also explored the nonlinear parameter strict feedback system control method of ILC for a class of unknown state delays in [24]. In addition, ILC was utilized to learn the controller of a closed-loop UAV system in [25].

On this basis, some model-free intelligent learning control methods have been proposed to be applied to UAVs. To name just a few, in [26] and [27], robust transform-based real-time adaptive intelligent control is used to offset the parametric uncertainty of the quadcopter and improve the stability under external disturbances. Reinforcement learning (RL) is implemented in [28], enabling the controller to learn when multiple actuators fail.

Later, a domain adversarially invariant meta-learning (DAIML) method is proposed to solve this problem in [29]. Their approach combines adaptive flight and reinforcement learning-based robot control. As mentioned in [30] and [31], future spacecraft also need to quickly and accurately deal with the control problems caused by spacecraft configuration changes, such as aerodynamics and loads changes. Typically, experts learn to migrate controller parameters based on experience in threefold, instruction tracking response, performance evaluation, and controller iterative refactoring.

In this paper, a real-time brain-inspired learning control (RbiLC) scheme is proposed, which consists of three steps: iterative instruction response, performance evaluation, and control parameter reconstruction. Moreover, we establish an appropriate Lyapunov-Krasovskii function to ensure that the controller structure converges to a compact set through iterative evolution during learning. Compared to existing work, our work's main contributions are:

- 1) A weak model-dependent brain-inspired adaptive learning reconstruction control algorithm is proposed which is based on self-supervised extremum-seeking. Unlike deep learning schemes, this learning process is interpretable and real-time.
- 2) A real-time cost evaluation function is developed to guide the controller parameters toward the optimal values. To mitigate the impact of wind disturbance, a digital wind direction measurement and anemometer hardware are connected to the ground station. They evaluate the external wind disturbance and incorporate it into the wind disturbance control compensation.

- 3) An adaptive extremum seeking method incorporating deep learning momentum acceleration is introduced into the RbiLC framework to achieve interpretability of the learning process. Furthermore, a stability theory proof is achieved based on the established Lyapunov-Krasovskii function.
- 4) A software-in-the-loop (STIL) simulations and real-world experiments are conducted to evaluate wind field disturbance cancellation in the learning process, considering disturbed wind fields and wind field measurements.
- 5) Finally, the proposed RbiLC algorithm is employed for online adaptive learning during real-world quadcopter flight, enabling the system to adapt to unknown configuration changes in the propellers or motors.

The rest of this paper is organized as follows. Section II formalizes the adaptive learning tracking control problem for a class of nonlinear systems with uncertain configurations, and this section reviews some necessary assumptions and mathematical tools. Section III presents the detailed design process of the brain-inspired adaptive learning controller and rigorously proves the stability of the control system. In Section IV, the effectiveness of the RbiLC is confirmed by SITL simulation, and real-world experiment in Section V. Section VI summarizes this work and gives an outlook for future research.

II. PRELIMINARIES

In this section, some basic concepts and results of UAV adaptive control systems are introduced, and basic problem descriptions are given. Consider a class of nonlinear dynamic systems with configuration uncertainties as follows:

$$\begin{cases} \dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t)) + \mathbf{g}(\mathbf{x}(t))\mathbf{u}(t) + \mathbf{w}(t) \\ \mathbf{y}(t) = \mathbf{h}(\mathbf{x}(t)) \end{cases} \quad (1)$$

where $\mathbf{u}(t) \in \mathbb{R}^n$, $\mathbf{y}(t) \in \mathbb{R}^n$, $\mathbf{x}(t) \in \mathbb{R}^m$, $\mathbf{w}(t) \in \mathbb{R}^m$ respectively represent the system control vector, system output vector, system state vector and unknown external disturbance mapping in the Euclidean space. $\mathbf{x}^n(t)$ denotes the n order derivative, $\mathbf{f}(\mathbf{x}(t)) : \mathbb{R}^n \rightarrow \mathbb{R}^n$ and $\mathbf{g}(\mathbf{x}(t)) : \mathbb{R}^n \rightarrow \mathbb{R}^n$ are nonlinear uncertain Lipschitz continuous functions.

In general, the multi-channel coupling control problem of UAV is always decoupled into multiple single-channel control problems as shown in [14] and [32]. The common fixed structure PID controller is defined as follows:

$$\mathbf{u}(t)_j = K_p^j \mathbf{e}(t) + K_i^j \int_0^t \mathbf{e}(\tau) d\tau + K_d^j \dot{\mathbf{e}}(t) \quad (2)$$

where K_p^j , K_i^j , K_d^j present proportional, integral and derivative terms of different channels controller gains, and $j \in [\phi, \theta, \psi]$. The performance function of nonlinear system controller with uncertain configurations can be defined as

$$J^*(\mathbf{x}(t)) = \arg \min_{\mathbf{u} \in \mathcal{A}(\Omega)} \int_t^\infty (\mathcal{S}(\mathbf{x}(\tau)) + U^*(\mathbf{x}(\tau), \mathbf{u}(\tau))) d\tau. \quad (3)$$

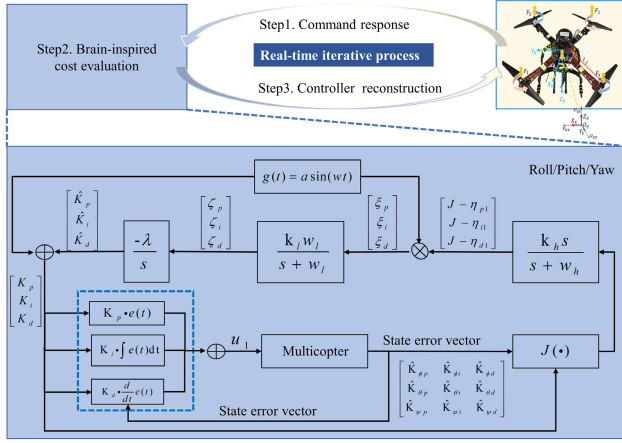


Fig. 1. Scheme of adaptive controller for multicopter. Our study achieves real-time reconfiguration of the controller by estimating the controller's optimal structural parameter matrix $\mathbf{K}^*$. A time-dependent sinusoidal signal with direction $g(t)$ is superimposed in each iteration step based on the controller estimate $\hat{\mathbf{K}}$ from the previous step as the next controller structural parameter $\mathbf{K}$, and u_1 is the output of flight controller and the $J(\cdot)$ is the performance function as show in (4).

The purpose of this paper is to get the system output $y(x)$ tending to the expected output $y^*(x)$ by updating the optimal cost corresponding $J^*(x)$ to the controller structure $U^*(x)$.

Model-free controllers, unlike traditional methods that rely on a prior mathematical model, rely on a data-driven feedback response from the system, such as extremum seeking (ES) and proportional-integral-derivative (PID) control. These methods can be used not only for linearized systems but also for nonlinear ones. In particular, the model-free PID control method [33], which is used by more than 95% of industry control systems, is suitable as the adaptive control [34]. Therefore, a self-learning controller framework that can reconfigure in real-time has significant research value. In this article, we describe a quadcopter control learning system in Figure 1.

The UAV control system has a problem of control channel coupling, which is usually addressed by decoupling it into single-channel processing.

Assuming that the optimal cost of the system for different dynamical model configurations can be formulated as a second-order Taylor expansion of J at the optimal compensation factor $\mathbf{K}^*$ with

$$\lim_{u \rightarrow u^*} J(\mathbf{K}^*) = 0. \quad (4)$$

The error of estimation defined as $\tilde{\mathbf{K}} = \mathbf{K}^* - \hat{\mathbf{K}}$, and the optimal performance function can be described as $f(\mathbf{K})^*$.

III. REAL-TIME BRAIN-INSPIRED ADAPTIVE LEARNING CONTROLLER

In this section, a real-time brain-inspired adaptive learning controller is presented for nonlinear system by following the design procedure of the adaptive learning method, where the Lyapunov-Krasovskii functions are employed to compensate for the uncertain configuration of the system. On this basis, the stability of the control system and the convergence of the command tracking error are proved by setting a suitable performance cost function.

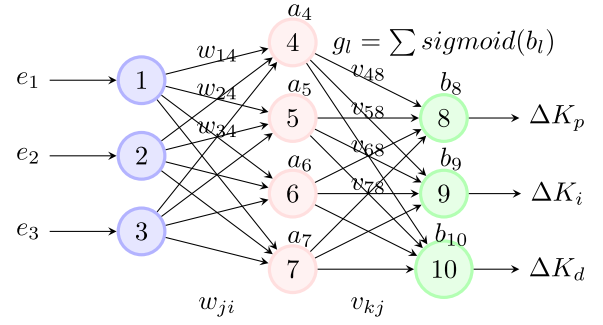


Fig. 2. Schematic diagram of the human brain learning controller parameters iteratively in real time. e_i , ($i = 1, 2, 3$) denotes the regulation error of three attitude channels, and ΔK present the system control parameters increment.

A. RBiLC Design

In this work, the brain-inspired process is utilized to calculate the controller's globally stabilized structure $\mathbf{K}(t) = [\mathbf{K}_p(t), \mathbf{K}_i(t), \mathbf{K}_d(t)]$ for unknown multirotor as shown in Fig. 1.

The step command error sequence under wind compensation is used as the system input, and the controller structure parameter $\mathbf{K}(t)$ is updated in each step iterative response as the system output.

$$\mathbf{K}(t) = \begin{bmatrix} K_{\phi p}(t) & K_{\phi i}(t) & K_{\phi d}(t) \\ K_{\theta p}(t) & K_{\theta i}(t) & K_{\theta d}(t) \\ K_{\psi p}(t) & K_{\psi i}(t) & K_{\psi d}(t) \end{bmatrix} \quad (5)$$

Consequently, combining (2), (5) with (36), one can define the adaptive basic controller as

$$\mathbf{U}(\mathbf{K}, t)_j = \mathbf{K}(t)[\mathbf{E}(t), \int_0^t \mathbf{E}(\tau) d\tau, \dot{\mathbf{E}}(t)]^T \quad (6)$$

where $K_{\phi p}(t)$, $K_{\phi i}(t)$, $K_{\phi d}(t)$ present adaptive proportional, integral and derivative terms of roll channel controller parameter, and the same as pitch and yaw channels. The error matrix $\mathbf{E}(t) = [e_\phi, e_\theta, e_\psi]^T$ denotes different channels errors.

Nevertheless, reconstructing the controller with optimal parameters in real-time is a challenging task. Inspired by the human brain searching for the optimal search strategy, this paper presents that the essence of deep neural networks is to iterate the gradient, as shown in Fig. 2. Besides, the Nesterov's Momentum [35] update rule in deep learning is given by

$$\begin{cases} V_{i+1} = \mu V_i - \lambda \nabla \hat{J}_i(K_i + \mu V_i) \\ K_{i+1} = K_i + V_{i+1} \end{cases} \quad (7)$$

where V_i is the incrementation of i th gradient, $\lambda, \mu \in (0, 1)$ denote the learning rate and monmenta parameter respectively, one can rewrite the controller updates as

$$\Delta U(t) = \Delta K_p^j e(t) + \Delta K_i^j \int_0^t e(\tau) d\tau + \Delta K_d^j \dot{e}(t). \quad (8)$$

The parameter gain of the controller will be nonlinear with the error, and the adaptive search for optimal performance controller structure is essentially a problem of finding an optimal solution as $\mathbf{K}^* = [\mathbf{K}_p^*, \mathbf{K}_i^*, \mathbf{K}_d^*]$, in a multidimensional manifold. For k th learning, we have $\mathbf{K}(k) = [K_\phi(k), K_\theta(k), K_\psi(k)]^T$, and the basic cost function

of nonlinear system performance of a whole learning cycle $t \in [t_0, t_0 + T]$ can be represented as

$$\mathcal{J} = \eta\sigma + (1 - \eta) \int_{t_0}^{t_0+T} e^2 dt \quad (9)$$

where $\eta \in [0, 1]$ presents the overshoot coefficient, here we set $\eta = 0.35$. The response overshoot can be caculated as

$$\sigma = \frac{y_{\max}(t) - y_d(t)}{y_d(t)} \times 100\%, t \in (t_0, t_0 + T) \quad (10)$$

and $y_{\max}$, y_d denote the max of measure value and the desire value, t_0 is the initial time of each learing process and T is the cycle time.

It is worth mentioning that the calculation methods of systems performance cost function are different from traditional methods such as integrated square error (ISE) or integrated absolute error (IAE) in [36], the calculation method is derived from the fact that human experts rely on the output response curve of the system to perform performance evaluation during the controller reconfiguration design process. The action-critic used in reinforcement learning also uses the similar design ethos. Nonetheless, manual methods that rely on expert experience are time-consuming and labor-intensive. Besides, the reinforcement learning neural networks are unexplainable black boxes. Therefore, a learning method based on interpretable gradient descent is urgently needed to make the system achieve local asymptotic stability after multiple response iterations. According to the brain-inspired cost evaluation funtion shown in (9), the cost function discrete form is made as follows:

$$\mathcal{J}(k) = \eta\sigma + (1 - \eta) \sum_{i=1}^{T_k/\Delta t} e_k^2(i) \cdot \Delta t \quad (11)$$

where $e_k(i)$ denotes the error in following the target trajectory with initial disturbance using the 1st iteration parameters $\mathbf{K}(1)$, and t_0 presents the start time of the step command. Suppose the gradient of the cost function $\nabla \mathcal{J}(k)$ is available and it could iteratively optimize the parameter of RBiLC. The parameter update rule can be defined as

$$\begin{cases} \mathbf{K}_p(k+1) = \mathbf{K}_p(k) - a_p \nabla \mathcal{J}_p(k), \\ \mathbf{K}_i(k+1) = \mathbf{K}_i(k) - a_i \nabla \mathcal{J}_i(k), \\ \mathbf{K}_d(k+1) = \mathbf{K}_d(k) - a_d \nabla \mathcal{J}_d(k). \end{cases} \quad (12)$$

where $\mathbf{K}(k)$ denotes the parameter vector after k th iterations and a_p, a_i, a_d denote the step size of the parameters of quadcopter at each step of the iteration. As $t \rightarrow \infty$, the cost function $\mathcal{J}(\mathbf{K}(k))$ will approach a local minimum in the manifold space of PID controller gains. The question transfer to how we can estimate $\nabla \mathcal{J}(\mathbf{K}(k))$. That is

$$\nabla \mathcal{J}(\mathbf{K}(k)) = \left(\frac{\partial}{\partial k_p} \mathcal{J}_p(k), \frac{\partial}{\partial k_i} \mathcal{J}_i(k), \frac{\partial}{\partial k_d} \mathcal{J}_d(k) \right) \quad (13)$$

and the derivative to the controller parameter evaluated error $\tilde{K} = K^* - \hat{K}$ can be computed as

$$\tilde{K}(k+1) = \left(-\frac{\lambda a^2}{2} \mathcal{J}'' \right) \tilde{K}(k) \quad (14)$$

where a is the brain inspire learning rate, and the λ is the adaptive gain for optimize the local controller parameters.

Hence, combin (1), (6) and (11), one have the adaptive learning controller of nonlinear control system can be rewritten as multichannel matrix form

$$\begin{cases} \mathbf{X}^n(t) = \mathbf{F}(\mathbf{X}(t)) + \mathbf{W}(t) \\ \quad + \mathbf{G}(\mathbf{X}(t))\mathbf{K}(t)[\mathbf{E}(t), \int_0^t \mathbf{E}(\tau)d\tau, \dot{\mathbf{E}}(t)]^T \\ \mathbf{Y}(t) = \mathbf{H}(\mathbf{X}(t)). \end{cases} \quad (15)$$

These parameters are randomly initialized to $\mathbf{K}(0)$, and then the system control cost function $\mathcal{J}(\mathbf{K}(k))$ is evaluated by periodically inputting perturbation step signals to update the parameter changes $\mathbf{K}(k)$. The initial value of the optimization value should be reset randomly several times and compared to find the global best results. For the PID parameters are randomly initialized, each iteration is optimized based on gradient-direction disturbances, which are guided by calculating the cost and gradient of the average response. By drawing the function relationship between cost function and iteration number, the convergence of the system can be guaranteed, and the working principle of this adjustment method is verified.

B. Theoretical Analysis

Before conducting the stability analysis of the online control of the nonlinear configuration uncertain system, we first need to make some assumptions.

Assumption 1: Define the transforms function as $\mathcal{M} = M(x)$, and one can rewrite the nonlinear control system (1) as follows

$$\begin{cases} \dot{z}_i = z_{i+1}, & i = 0, \dots, n-1 \\ \dot{z}_n = f(z) + g(z)u(t) + w(t), \\ y = z_1. \end{cases} \quad (16)$$

where $z^* \triangleq (y^*, 0, \dots, 0)^T \in \mathbb{R}^n$, and $m \in \mathbb{R}^n$, $J_M(\cdot)$ presents the Jacobian matrix of map $M(x) : \mathbb{R}^n \rightarrow \mathbb{R}^n$.

$$M(x) = (h(x), \sum_{i=1}^n \frac{\partial h}{\partial x_i}(x) f_i(x), \dots, \sum_{i=1}^n (\frac{\partial h}{\partial x_i}(x))^n f_i(x))^T. \quad (17)$$

For any setpoint $y^*(t)$, there always exists increasing function f_1, f_2 from $\mathbb{R}^+ \rightarrow \mathbb{R}^+$ with $\limsup_{x \rightarrow 0} \frac{f_1(x)}{x} < \infty$ and $\limsup_{x \rightarrow 0} \frac{f_2(x)}{x} < \infty$.

Remark 1: The bottom bound of the known constant $\underline{c}$ and the upper bound of the function f_1, f_2 are important in constructed the controller parameters. Thus, the configuration uncertain system can be written as

$$\begin{cases} \dot{x}_i = x_{i+1}, & i = 0, \dots, n-1 \\ \dot{x}_n = f(x) + g(x)u(t) + w(t), \\ y = x_1. \end{cases} \quad (18)$$

where $f(x), g(x)$ denote the uncertain function of system. Suppose

$$\begin{cases} g(x) \geq \underline{c}, \\ |f(x)| + |g(x)| + \|J_f(x)\| + \|J_g(x)\| \leq r(\|x\|). \end{cases} \quad (19)$$

where $J_f(x), J_g(x)$ denote the gradient of function $f(x), g(x)$, respectively. $r: \mathbb{R}^+ \rightarrow \mathbb{R}^+$ is continuous and increasing function.

Assumption 2: There exist constants $c_1, c_2 \in (0, +\infty)$, then the inverse of the system Jacobian Matrix $J_M(x)$ have

$$\frac{\|J_M^{-1}(x)\| - c_2}{\|x\|} \leq c_1 \cdot \max(0, \log(x)), \forall x \in \mathbb{R}^n \quad (20)$$

Remark 2: The super-linear rate of $\|J_M(x)\|$ in Assumption 2 cannot be relaxed essentially. Since the transformation function M defined by (17) is identify, this assumption is automatically satisfied.

Definition 1: It should be pointed out that, based on System (1) and Assumptions 1, 2, we define a class of nonlinear structural parameter uncertain systems as $\mathcal{N}$, then the configuration uncertain systems can be described as following:

$$\mathcal{N}(\underline{c}, \mathcal{F}_1, \mathcal{F}_2) = (f, g, h), \quad (21)$$

Lemma 1: (see [37]:) Define $\bar{Y}(t) = (y_0(t), \dots, y_n(t))^T$, and suppose that there exists two bounded functions f_a and g_a which are defined on $t \in [0, T]$, it has

$$\begin{cases} \dot{y}_i = y_{i+1}, & i = 0, \dots, n-1 \\ \dot{y}_n = f_a + g_a u_i + w(i). \end{cases} \quad (22)$$

where $f_a = F(x(t)) - F(x^*)G(x(t))/G(x^*)$ and $g_a = G(x(t))$. For arbitrary constant $\epsilon > 0$ and any initial parameters $\mathbf{K} \in \mathbb{R}^{3 \times 3}$, if $b_0 = f_1(c\epsilon)$, and $\alpha, w, \lambda \in (0, \infty)$, then we have $\|\bar{Y}(t)\| \leq ce^{-\alpha t}\epsilon$, when $t \in (t_0, t_0 + T)$.

Theorem 1: Consider a class of nonlinear uncertain systems (21) with adaptive reconfiguration learning control rules (14). For any given $(\underline{c}, \mathcal{F}_1, \mathcal{F}_2)$, there is an adaptive structural parameter $\mathbf{K}$ in the manifold space $\mathcal{P} \in \Omega^{3 \times 3}$, such that any closed-loop control system $(f, g, h) \in \mathcal{N}(\underline{c}, \mathcal{F}_1, \mathcal{F}_2)$ of the adjustment error $e(t) = y^* - y(t)$ can converge to zero on $[t_0, t_f] \in [0, \infty]$.

$$\lim_{t \rightarrow \infty} (y(x(t))^* - y(x(t))) = 0. \quad (23)$$

Proof: Defining the Lyapunov-Krasovskii function as follows:

$$V_{i,k} = \frac{1}{2} v_{i,k}^2 + \frac{1}{2} \sum_{j=1}^i \int_{t-d_j}^t \psi_{i,j}^2(x_{j,k}(\tau)) d\tau \quad (24)$$

To confirm the stability of the control system, one designated the composite performance evaluation function at the k th learning as follows:

$$\mathcal{J}(k) = V_{i,k} + \eta(k)\sigma(k) + (1 - \eta(k)) \sum_{i=1}^{T_k/\Delta t} e_k^2(i) \cdot \Delta t. \quad (25)$$

The whole proof is composed of three parts:

Step 1: For any admissible control law $u(t) \in \mathcal{A}(\Omega)$, reshape the continuously differentiable performance function

$\mathcal{J}(k)$ to $J(x)$, then the nonlinear Lyapunov equation can be defined as

$$\begin{aligned} 0 = & \mathcal{S}(x) + U(x(\tau), u(\tau)) \\ & + (\nabla J(x))^T (f(x) + g(x)u(x) + w) \end{aligned} \quad (26)$$

and $\nabla J(x) = 0$, let Hamilton form be

$$\begin{aligned} H(x, u(x), \nabla J(x)) = & \mathcal{S}(x) + U(x, u(x)) \\ & + (\nabla J(x))^T (f(x) + g(x)u(x) + w). \end{aligned} \quad (27)$$

From (3), Hamilton-Jacobi-Bellman (HJB) equation is build as

$$\min_{u \in \mathcal{H}(\Omega)} H(x, u(x), \nabla J^*(x)) = 0 \quad (28)$$

then, the optimal control problem can be established as

$$u^*(x) = \arg \min_{u \in \mathcal{H}(\Omega)} H(x, u(x), \nabla J^*(x)). \quad (29)$$

Besides, combining the quadratic utility term $u^T R u$ in $U(x, u) = x^T Q x + u^T R u$, the controller U^* is also written as

$$U^*(x, u) = -\frac{1}{2} R^{-1} g^T(x) \nabla J^*(x) \quad (30)$$

Combining (26),(28) with (42), the HJB equation is also written as

$$\begin{aligned} 0 = & \mathcal{S}(x) + U(x, u^*(x)) \\ & + (\nabla J^*(x))^T (f(x) + g(x)u^*(x) + w) \end{aligned} \quad (31)$$

with $J^*(K^*) \rightarrow 0$, then $H(x, u^*(x), \nabla J^*(x)) = 0$, exciting utility term $\mathcal{S}(x)$ as

$$\mathcal{S}(x) = \frac{1}{4} (\nabla J^*(x))^T \nabla J^*(x) + \Gamma^2(x) \quad (32)$$

we have

$$\begin{aligned} \frac{dJ^*(x)}{dt} = & (\nabla J^*(x))^T [f(x(t)) + g(x(t))u(t) + w]_{\dot{x}} \\ = & -\frac{1}{4} (\nabla J^*(x))^T \nabla J^*(x) - \Gamma^2(x) \\ & - U(x, u^*(x)) + (\nabla J^*(x))^T w(x) \end{aligned} \quad (33)$$

therefore, we can obtain

$$\begin{aligned} \dot{J}^*(x) = & -U(x, u^*(x)) - \Gamma^2(x) + (w(x))^T w(x) \\ & - \left[\frac{1}{2} \nabla J^*(x) - w(x) \right]^T \left[\frac{1}{2} \nabla J^*(x) - w(x) \right] \\ \leq & -U(x, u^*(x)) - [\Gamma^2(x) - (w(x))^T w(x)] \end{aligned} \quad (34)$$

According to the boundary conditions $\|\Delta f(x)\| - \Gamma(x) \leq 0$, (34) is equivalent to

$$\dot{J}^*(x) \leq -U(x, u^*(x)) < 0, \forall x \neq 0. \quad (35)$$

Hence, the control system has optimal parameters to satisfy Lyapunov local asymptotic stabilization.

Step 2: Next, it is also necessary to prove that the learning method based on cost evaluation can find the optimal control configuration. According to Fig. 1, the system performance

evaluation cost function (single channel) can be abstracted into the following

$$J(K) = \mathcal{N}^*(\underline{c}, \mathcal{F}_1, \mathcal{F}_2) + \frac{\mathcal{N}''}{2}(K - K^*)^2 \quad (36)$$

it could be obtained from (36) that the estimate parameters $\hat{K} = K - a \sin \omega t$ and $\tilde{K} 2 \sin^2 \omega t = 1 - \cos 2\omega t$

$$\begin{aligned} J(K) &= \mathcal{N}^* + \frac{\mathcal{N}''}{2}(\tilde{K} - a \sin \omega t)^2 \\ &= \mathcal{N}^* + \frac{\mathcal{N}''}{2} \left(\tilde{K}^2 - 2a\tilde{K} \sin \omega t + \frac{a^2}{2} - \frac{a^2}{2} \cos 2\omega t \right) \\ &= \mathcal{N}^* + \frac{\mathcal{N}''}{2} \tilde{K}^2 - a\mathcal{N}'' \tilde{K} \sin \omega t + \frac{a^2 \mathcal{N}''}{2} \sin^2 \omega t \\ &= \mathcal{N}^* + \frac{a^2 \mathcal{N}''}{4} + \frac{\mathcal{N}''}{2} \tilde{K}^2 - a\mathcal{N}'' \tilde{K} \sin \omega t \\ &\quad + \frac{a^2 \mathcal{N}''}{4} \cos 2\omega t \end{aligned} \quad (37)$$

where ω_i presents modulation frequency, direct current (DC) signal in (37) can be eliminated after high-pass filtering, that is

$$\frac{s}{s+h}[J] = \frac{\mathcal{N}''}{2} \tilde{K}^2 - a\mathcal{N}'' \tilde{K} \sin \omega t + \frac{a^2 \mathcal{N}''}{4} \cos 2\omega t. \quad (38)$$

Further,

$$\xi = \frac{s}{s+h}[J] \cdot a \sin \omega t. \quad (39)$$

Submitting $2 \sin^2 \omega t = 1 - \cos 2\omega t$ into (39), it yields

$$\begin{aligned} \xi &= -\frac{a\mathcal{N}''}{2} \tilde{K} + \frac{a\mathcal{N}''}{2} \tilde{K} \cos 2\omega t \\ &\quad + \frac{a^2 \mathcal{N}''}{8} (\sin \omega t - \sin 3\omega t) + \frac{\mathcal{N}''}{2} \tilde{K}^2 \sin \omega t \end{aligned} \quad (40)$$

and

$$\tilde{K} = \frac{\lambda}{s} \xi. \quad (41)$$

Feeding the result of combine singal (40) and (41), we obtain

$$\tilde{K}(k+1) = \left(-\frac{\lambda a^2}{2} \mathcal{N}'' \right) \tilde{K}(k). \quad (42)$$

Since $\lambda a^2 > 0$ and $\mathcal{N}'' > 0$, it can be seen that the estimation error is monotonically decreasing, so the system performance function can take the optimal value $J(K) \rightarrow J^*(K^*)$ in the finite time domain, and the corresponding control parameters are the optimal controller parameters $\mathbf{K}^*$.

Step 3: Based on the proof of the local asymptotic stability of the system in step 1 and the proof of the convergence of the learning rate in step 2, we also need to prove the convergence of the stabilization error at the optimal point to ensure that the asymptotic stability of the reconfigured controller adjustment error is close to zero. By Assumptions 1, 2 and Lemma 1, we have

$$\begin{aligned} |f_a| &= |F(x(t)) - F(x^*)G(x(t))/G(x^*)| \\ &\leq |F(x(t)) - F(x^*)| + f_1(|y^*|)|G(x(t)) - G(x^*)|/\underline{c} \\ &\leq (\underline{c} + f_1(|y^*|))\|H(x(t)) - H(x^*)\|/\underline{c} \\ &\leq (\underline{c} + f_1(|y^*|))g_1(\|M(x(t)) - M(x^*)\|)/\underline{c} \end{aligned}$$

$$\begin{aligned} &= (\underline{c} + f_1(|y^*|))g_1(\|Y(t)\|)/\underline{c} \\ &\leq (\underline{c} + f_1(|y^*|))g_1(\|\bar{Y}(t)\|)/\underline{c} \end{aligned} \quad (43)$$

and

$$\begin{aligned} \|Y(0)\| &= \|M(x(0)) - z^*\| \\ &\leq f_1(\|x(0)\|) + \|z^*\| \\ &\leq f_1(\epsilon) + |y^*|. \end{aligned} \quad (44)$$

Accordingly,

$$\begin{aligned} \lim_{k \rightarrow k^*} |e(k)| &= \lim_{k \rightarrow k^*} |y(x(k))^* - y(x(k))| \\ &= \lim_{k \rightarrow k^*} |y_1(k)| \\ &\leq \lim_{k \rightarrow k^*} \|\bar{Y}(k)\| \\ &\leq \lim_{k \rightarrow k^*} \underline{c} e^{-a \frac{1}{|k-k^*|}} \\ &= 0. \end{aligned} \quad (45)$$

This completes the proof of Theorem 1. $\square$

IV. SIMULATION VERIFICATION

Software in the loop (SITL) simulation is performed to validate the effectiveness of the proposed brain-inspired learning control scheme in this section. The simulation platform was built on Ubuntu18.04.6 LTS system, with the hardware configuration as 11th Gen Intel Core i7-11800H@2.30GHz*16 and disk as 151.0G. The quadcopter model is simulated with X-type as shown in Fig. 6. The ground control station (GCS) uses the open source ground station Mission Planer running on Windows 10 and connecte with a quadcopter using MAVProxy by Local Area Network (LAN).

In order to ensure the stability of the position outer loop control, the stability of the inner attitude loop should be considered first. This paper study the angular controller to verify the effect of the proposed RBiLC on the cascade controller (angle-rate-loop). Attitude control includes three channels, roll, pitch, and yaw. The proposed single input single output (SISO) method RBiLC is used to control this channel individually, as shown in Fig. 1. The parameter values based on measurements of the copter used in this simulator are set as shown in Table I. The random wind disturbance speed is set to $v_{wind} = 1.3$ m/s.

The simulation process is shown in Fig. 3. At the beginning of the simulation, we first use the default initialization PID controller parameters, climb 10 meters in the air in position mode, hover for 3 seconds, and then switch the flight mode to self-learning (SL) mode. Then the UAV performs the ϕ^+ maneuver, restores the horizontal attitude, and remains horizontal until an action evaluation calculation is completed. The iterative parameter estimate $K(1)$ is given. Then the controller reconstructed based on $K(1)$ is used, $U(K(1), x)$ controls the next roll hostile maneuver of the system. Similarly, the iterative update controller is $U(K(2), x)$. Finally, after reaching the maximum number of iterations or setting the cost threshold, the learning will be stopped and the learned parameters will be recorded. The pilot will decide whether to use the learned parameters. The same as pitch and yaw

Algorithm 1 RBILC Algorithm for Attitude Control**Input:****Initialization:** attitude controller parameters:

$$\mathbf{K}_p(0) = [K_{\phi,p}(0), K_{\theta,p}(0), K_{\psi,p}(0)]^\top;$$

$$\mathbf{K}_i(0) = [K_{\phi,i}(0), K_{\theta,i}(0), K_{\psi,i}(0)]^\top;$$

$$\mathbf{K}_d(0) = [K_{\phi,d}(0), K_{\theta,d}(0), K_{\psi,d}(0)]^\top;$$

 $J(k)$: cost function value in time k , $J(0) = 0$;**Hyperparameters:** $\eta_0 = 0.5$: overshoot coefficient; $\alpha_0 = 0.2$: brain inspire learning rate; $\lambda = 0.6$: adaptive gain; δ_0 : sample time; k_{max} : the maximum cycle of learning;**Output:**

$$U_{\phi,\theta,\psi}(K(k)) = [E((k-1)), \sum_{i=0}^{m_k} E(i) \cdot \Delta t, \frac{E(k)-E(k-1)}{\Delta t}] \cdot \mathbf{K}(k-1).$$

- 1: Adaptive learning process
- 2: **for** channels number $j = 1 : 3$ **do**
- 3: **for** iter number $k = 1 : k_{(j,max)}$ **do**
- 4: Check attitude level;
- 5: **if** attitude is level & $J_j(k) > J_j^*$ **then**
- 6: $J_j(k) \leftarrow$ cost value as shown in (11);
- 7: $\nabla \mathbf{K}_j(k) \leftarrow$ parameters change estimator from (14);
- 8: $\mathbf{Y}_j(k) \leftarrow$ do action $(-1)^k [\phi_d, \theta_d, \psi_d]_j^\top$;
- 9: back to level;
- 10: $k++$;
- 11: Update $U_j(K(k))$;
- 12: **else**
- 13: Wait back to level();
- 14: Update $U_j(K(k)) = U_j(K(0))$;
- 15: **while** The learning process is interrupted **do**
- 16: Baseline controller takeover;
- 17: Update $U_j(K(k))$;
- 18: Logging $\mathbf{K}_j^*$;
- 19: **return** $U_{\phi,\theta,\psi}(K(k))$.



Fig. 3. Snapshot of SITL for quadcopter attitude controller adaptive learning. There are three main parameters $[K_p, K_i, K_d]$ for a single channel. Previous research shows that the proportional term coefficient K_p is the primary influence. Therefore, in this paper, we give the initial values $\mathbf{K}_i(0)$ and $\mathbf{K}_d(0)$ of ZN adjustment in the SITL simulation environment and real test flight environment in advance, mainly for the $\mathbf{K}_p$ term, which has more influence on stability.

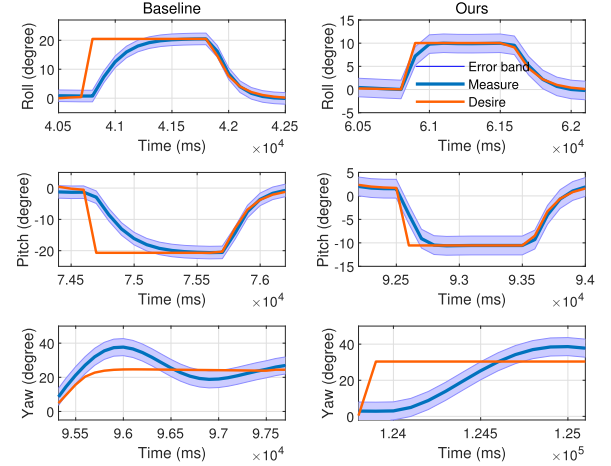


Fig. 4. Compare the control response of quadcopter drone systems before and after adaptive learning in SITL.

TABLE I

PARAMETERS OF QUADCOPTER PLATFORM

Symbol	Quantity	Value	Unit
m	Mass of quadcopter	1.673	kg
l	Diagonal motor shaft distance	0.45	m
b	Motor lift coefficient	0.63	—
g	Gravity coefficient	9.81	m/s ²
I_x	x-axis moment of inertia	0.0071	kg·m ²
I_y	y-axis moment of inertia	0.0071	kg·m ²
I_z	z-axis moment of inertia	0.0127	kg·m ²

channels. The process logic for designing the self-learning flight control is shown in Fig. 1. The $\mathcal{J}$ denotes the error cost over a period defined in (9).

V. EXPERIMENTATION

In order to verify the effectiveness of the proposed method, we built an X-type quadcopter flight platform based on the

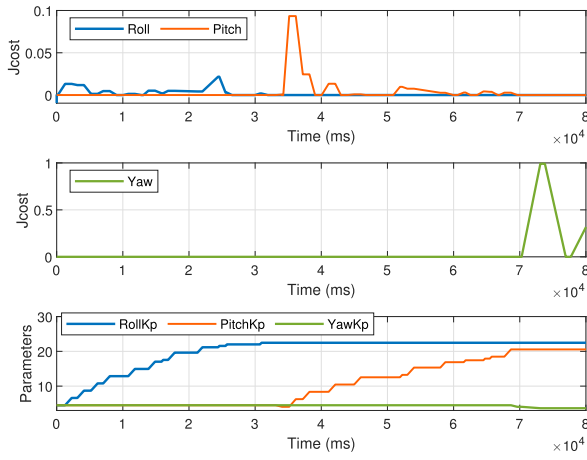


Fig. 5. Cost value and structure parameters of quadcopter controller during RiBLC learning in SITL.

TABLE II
HARDWARE OF QUADCOPTER PLATFORM

Name	Description	Number	Function
Motor	2312KV brushless DC motor	4	Power plant
Propeller	9450*2-blade / 9455*3-blade	4	Force
ESC	Hobbywing 20A	4	Motor driving
Li-Po	2200mAh	1	Battery
Servo	MG 996R	1	Package deliver

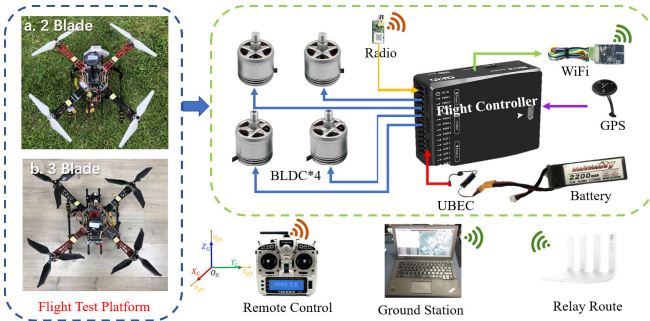


Fig. 6. Hardware of quadcopter experimental platform.

F450 frame with the following specific parameters: 2312kv brushless motor, Hobbywing 20A brushless electronic speed controller (ESC), flight controller hardware, UAV body frame, battery, 9450 self-locking propeller and 9455 noise-reducing folding three-blade propellers. The specific parameters are given in Table II.

1) *UAV Design*: All of our experiments were conducted at the NearSpace Vehicle Research Center of Zhejiang University. The overall UAV hardware architecture of the flight platform is shown in Fig. 6. We designed two aerodynamic characteristics of the propeller (2-blade and 3-blade) to simulate an uncertain variation of the UAV power system model. Without relying on a priori knowledge of the UAV, real-time controller reconfiguration can be achieved for different models using only real-time aerial learning data.



Fig. 7. The fully automatic realtime aerovane.

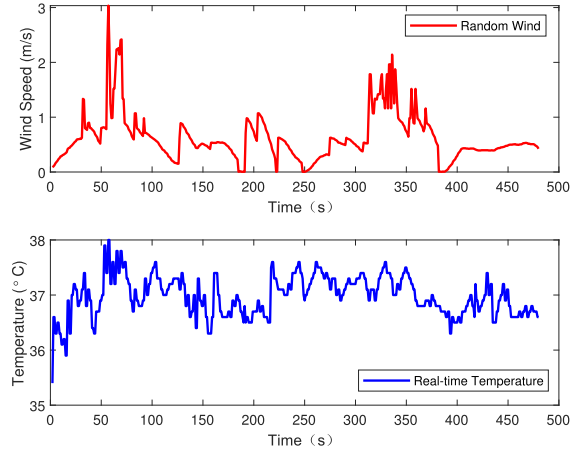


Fig. 8. Outdoor random wind and temperature.

The 2-blade propeller (APC material) radius is $r_1 = 119.38\text{mm}$, pitch is $c_1 = 127\text{mm}$ and weight is $w_1 = 11.3\text{g}$. The 3-blade propeller (carbon fiber material) radius is $r_2 = 119.38\text{mm}$, pitch is $c_2 = 139.7\text{mm}$, weight is $w_2 = 35\text{g}$, and the 3-blade propeller tip is optimized for noise reduction. For external random wind disturbance, the wind speed and direction of the flight site need to be measured in real-time. However, the existing equipment in the market can not realize the automatic cooperative measurement of wind direction and speed. Then we designed and built a universal real-time wind direction anemometer based on a normal anemometer. It can automatically follow the maximum wind direction steering to measure the test flight environment. This aerovane, as shown in Fig. 7 comprises of digital aerovane model (Aicevoos-AH-10) and an automatic changeover, was designed to measure the random speed and direction of wind disturbance automatically in Fig. 8.

All experiments were conducted outdoors as shown in Fig. 9, the aircraft positioning system based on a Global Positioning System (GPS) module, the data transmission link based on a WIFI module, and the remote control link based on the open source OPEN TX system with SBUS protocol.

A. Case A: Learning Under 2-Blade Propeller Configuration

The 2-blade propeller adopts a 9450 self-locking plastic paddle. The control effect of manual experience for parameter adjustment is used as the baseline. Then the RBiLC is intervened for learning control. Finally, the control effect is compared, and the quantitative indexes are selected as rising time overshoot and average cumulative error.

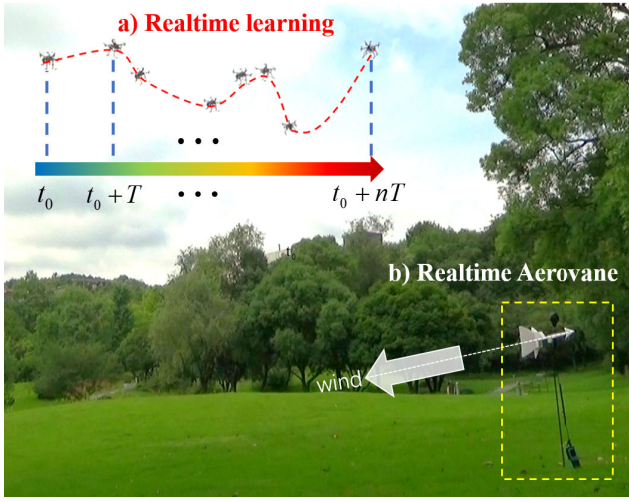


Fig. 9. Outdoor flight test of RBiLC for quadcopter uncertain model platform. In a): t_0 denotes the time of learning start time, and T denotes the periodic time of random step signal, n denotes the max number of iteration learning. In b): In real-time, the wind vane can record the relative air speed in the direction of the maximum wind speed in the outdoor random wind field.



Fig. 10. The real-time snapshots of the UAV's periodic command steps response process under the condition of unknown power configuration in real-world.

We have installed a miniature camera on board to record environmental information during the learning process. The on-board screen, which displays different learning instructions, is presented in Fig. 10. The excitation response process of the UAV under various unknown configuration model conditions was recorded by the on-board camera. It was observed that at different heights with the same horizontal position, high-altitude obstructions are reduced while wind disturbance is more significant, and the low-altitude wind is relatively weak. Therefore, high-altitude aircraft require a particular wind compensation inclination command to maintain a horizontal position. Additionally, when the periodic command returns to zero, it is necessary to superimpose the wind disturbance compensation command based on the horizontal attitude.

$$[\phi, \theta, \psi]_{cmd} = [\phi, \theta, \psi]_{level} + [\phi, \theta, \psi]_{desire}, \quad (46)$$

where $[\phi, \theta, \psi]_{cmd}$ represent the real desire command angles send to attitude controller, $[\phi, \theta, \psi]_{level}$ represent measure-level angles, and $[\phi, \theta, \psi]_{desire}$ represent the relative desire angles.

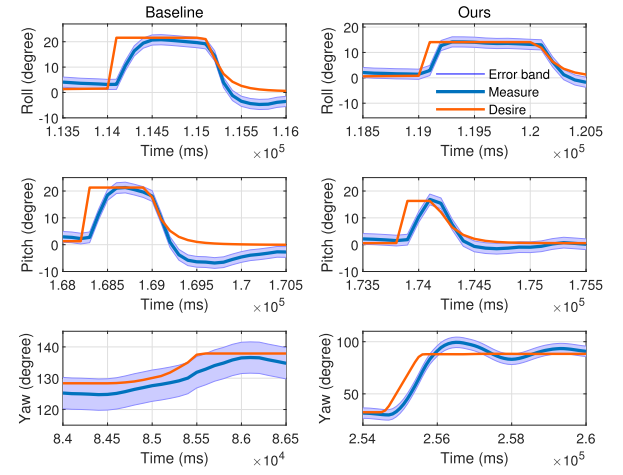


Fig. 11. 2-blade in real-world: attitude tracking response of self-learning. The left output curve are the step responses of $[\phi, \theta, \psi]$ for the model-free controller designed based on ZN rules, and the right are the step response used RBiLC method.

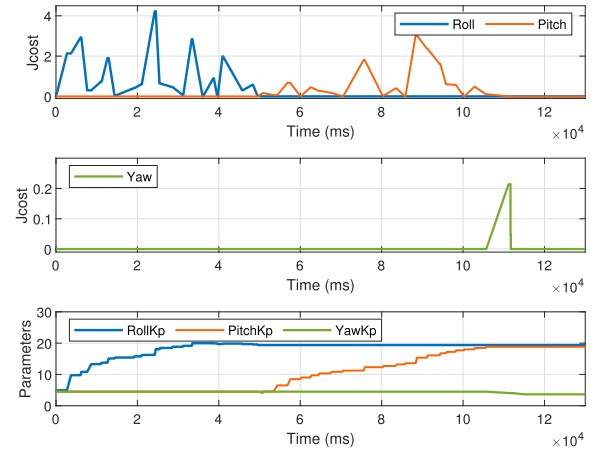


Fig. 12. 2-blade in real-world: the cost value and the controller parameters of roll, pitch, yaw (ϕ, θ, ψ) during the attitude controller self-learning.

The disadvantage of the ZN method is that the optimal controller reconstruction design needs to be carried out through multiple manual offline trials and errors for unknown control models. In contrast, the RBiLC method can significantly reduce the rise time and the overshoot of the roll and pitch channels' responses. It should be noted, however, that due to wind compensation, the yaw channel may experience partial overshoot when tracking large orders exceeding 50° , but the rise time has been significantly improved.

Comparing the attitude tracking learning effect of the RBiLC controller with that based on the ZN method using the data presented in Figure 11, we can observe that RBiLC reduces the rise time of all three attitude channels by at least 38.46%. Additionally, overshoot is also reduced in this experimentation.

To downscale the 9-dimensional space solution problem to a 3-dimensional one and reduce the complexity of channel coupling, independent learning is carried out in the order of roll, pitch, and yaw. This is achieved by updating the controller parameters through each iteration of the cost function, as shown in Figure 12.

TABLE III
3 AXIS ATTITUDE (ϕ, θ, ψ) PERFORMANCE COMPARISON

Case	Method	Rise time(Second)	Overshoot(%)	Controller Gains
SITL	Baseline1-ZN-PID	[0.75, 0.42, 0.9]	[0, 2.01, 0]	[4.53, 4.53, 4.41]
	Ours-RBiLC	[0.42, 0.21, 0.35]	[0, 1.53, 3.12]	[22.84, 21.53, 4.04]
	$\sigma_p(\%)$	[56.67↓, 52.32↓, 61.11↓]	[0.23.88↓, 3.12↑]	[↑, ↑, ↓]
Two-blade RW	Baseline2-ZN-PID	[0.52, 0.26, 0.47]	[5.12, 8.23, 0]	[4.53, 4.53, 4.41]
	Ours-RBiLC	[0.31, 0.16, 0.21]	[2.01, 3.32, 10.26]	[19.8, 19.6, 3.8]
	$\sigma_p(\%)$	[40.38↓, 38.46↓, 55.31↓]	[60↓, 62.5↓, 10.26↑]	[↑, ↑, ↓]
Three-blade RW	Baseline3-ZN-PID	[0.36, 0.42, 0.23]	[3.68, 6.15, 39.12]	[4.24, 4.24, 4.50]
	Ours-RBiLC	[0.25, 0.28, 0.11]	[1.71, 3.03, 20.78]	[20.13, 22.82, 4.10]
	$\sigma_p(\%)$	[30.56↓, 33.33↓, 52.17↓]	[53.53↓, 50.73↓, 46.89↓]	[↑, ↑, ↓]

1. RW in *Two-blade RW* and *Three-blade RW* represents *Real World* experiments, ZN represents the Ziegler-Nichols method.

2. Relative change σ_p between ZN-PID and RBiLC was defined as $\sigma_p = (V_{ZN-PID} - V_{RBiLC})/V_{RBiLC}$, and V denotes as the value of rise time and the overshoot.

3. To simplify the learning process, we set the integral terms k_i and differential terms k_d in the whole process to remain immovable for the time being, taking a value of 0.01.

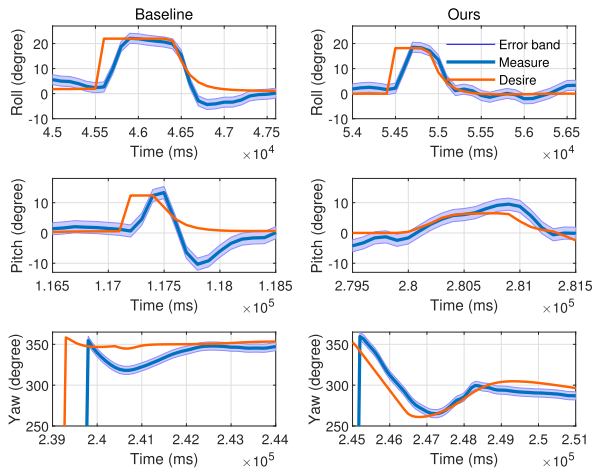


Fig. 13. 3-blade in real-world: attitude tracking response of self-learning. The left output curve are the step responses of $[\phi, \theta, \psi]$ for the model-free controller designed based on ZN rules, and the right are the step response used RBiLC method.

B. Case B: Learning Under 3-Blade Propeller Configuration

The learning experiment of the 2-blade propeller can fully demonstrate the effectiveness of the learning process. However, to more effectively illustrate that this method is also effective when dealing with UAVs with unknown configurations or uncertain physical models, we designed a 3-blade carbon fiber propeller with an unknown model to conduct control reconstruction experiments on the same body with learning while flying.

As can be seen from the Fig. 13, the attitude control system roll and yaw channel overshoot after learning have been significantly lower. At the same time, in order to resist gusts, the aircraft equilibrium state needs a roll angle to counteract the wind disturbance. Then there will be a problem of small final equivalent attitude instructions and slow iteration of controller parameters, we solve by expanding the maximum number of

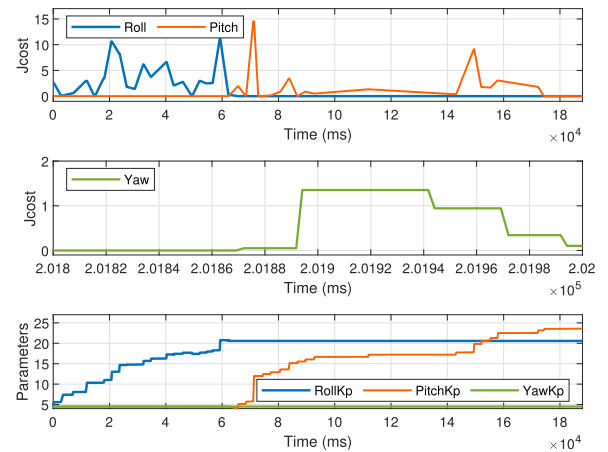


Fig. 14. 3-blade in real-world: the cost value and the controller parameters of roll, pitch, yaw (ϕ, θ, ψ) during the attitude controller self-learning.

iterations and accelerating the historical momentum. In the Fig. 14, after a sufficient number of iterations, the performance cost function of the system may fluctuate or stagnate in gusty conditions but may eventually be J^* .

Based on *CaseA* in above, we only replace the propeller of the unknown dynamic model and simulate the controller adaptive reconstruction learning process under the unknown configuration. Fig. 14 shows that after about 3~5 minutes of learning, the overall performance cost tends to be zero. Combined with the automatic heading adjustment compensation mechanism to the wind direction, the standard step response is excessively soft variable step instructions, thereby reducing wind field interference and realizing pilot test flight debugging without manual intervention.

We also conducted flight experiments to compare the attitude and motor output changes before and after the UAV dropped the package with or without the quadcopter controller turning on the self-learning function. The results show that the

learning controller can easily cope with the random system perturbations caused by the parcel drop and can quickly and stably restore a smooth flight.

VI. CONCLUSION

In this paper, a real-time brain-inspired learning control (RbiLC) algorithm is developed to solve the multi-rotor control problem in a class of nonlinear systems with unknown configurations. A cost function, similar to the control effect evaluation of the human brain, is used to guide the adaptive reconstruction of the controller parameters in real-time to obtain the UAV controller fitting for the unknown configuration. The algorithm's logic and effectiveness are verified in the SITL environment and deployed on a quadcopter in a natural wind disturbance environment for actual flight. The results demonstrate that the RbiLC can reduce the rise time by at least 30%, decrease the overshoot response by 20%, and, most importantly, reduce the full process controller reconfiguration time. It can also learn controller parameters for $[\phi, \theta, \psi]$ in 3~5 minutes. Future work will incorporate tiltable vertical takeoff and landing vehicles (VTOL) into the adaptive learning framework to minimize the design of flight controllers with varying model parameters.

ACKNOWLEDGMENT

The authors would like to thank the valuable contributions of Dr. Wenwen Zhao, Dr. Zhongzheng Jiang, and Yesi Chen during the experimental stage, as well as the contributions of Prof. Chao Xu and Dr. Yong Chen during the revision of the article. They would also grateful for their efforts and support.

REFERENCES

- [1] H. Anfinsen and O. M. Aamo, "Adaptive control of linear 2×2 hyperbolic systems," *Automatica*, vol. 87, pp. 69–82, Jan. 2018.
- [2] Z. Wang, W. Bao, and H. Li, "Second-order dynamic sliding-mode control for nonminimum phase underactuated hypersonic vehicles," *IEEE Trans. Ind. Electron.*, vol. 64, no. 4, pp. 3105–3112, Apr. 2017.
- [3] X. Dong, Y. Hua, Y. Zhou, Z. Ren, and Y. Zhong, "Theory and experiment on formation-containment control of multiple multirotor unmanned aerial vehicle systems," *IEEE Trans. Autom. Sci. Eng.*, vol. 16, no. 1, pp. 229–240, Jan. 2019.
- [4] Y. Zhang, H. Yang, Y. Chen, and W. Chen, "Adaptive extremum seeking controller via nonlinear variable gain for uncertainty model multirotor," in *Proc. 41st Chin. Control Conf. (CCC)*, Jul. 2022, pp. 2308–2314.
- [5] H. Kang, H. Li, J. Zhang, X. Lu, and B. Benes, "FlyCam: Multitouch gesture controlled drone gimbal photography," *IEEE Robot. Autom. Lett.*, vol. 3, no. 4, pp. 3717–3724, Oct. 2018.
- [6] M. Doostmohammadian, A. Taghieh, and H. Zarrabi, "Distributed estimation approach for tracking a mobile target via formation of UAVs," *IEEE Trans. Autom. Sci. Eng.*, vol. 19, no. 4, pp. 3765–3776, Oct. 2022.
- [7] H. K. Tran, J.-S. Chiou, and V.-H. Dang, "New fusion algorithm-reinforced pilot control for an agricultural tricopter UAV," *Mathematics*, vol. 8, no. 9, p. 1499, Sep. 2020.
- [8] *Electric Passenger-Grade Autonomous Aerial Vehicle*. Accessed: Jun. 1, 2023. [Online]. Available: <https://www.ehang.com/cn/ehangaav/>
- [9] M. Jin, J. Lee, and N. G. Tsagarakis, "Model-free robust adaptive control of humanoid robots with flexible joints," *IEEE Trans. Ind. Electron.*, vol. 64, no. 2, pp. 1706–1715, Feb. 2017.
- [10] Y. Zhang and J. Jiang, "Bibliographical review on reconfigurable fault-tolerant control systems," *Annu. Rev. Control*, vol. 32, no. 2, pp. 229–252, Dec. 2008.
- [11] S. A. U. Islam, T. W. Nguyen, I. V. Kolmanovsky, and D. S. Bernstein, "Data-driven retrospective cost adaptive control for flight control applications," *J. Guid., Control, Dyn.*, vol. 44, no. 10, pp. 1732–1758, Oct. 2021.
- [12] A. T. Hafez, A. J. Marasco, S. N. Givigi, M. Iskandarani, S. Yousefi, and C. A. Rabbath, "Solving multi-UAV dynamic encirclement via model predictive control," *IEEE Trans. Control Syst. Technol.*, vol. 23, no. 6, pp. 2251–2265, Nov. 2015.
- [13] H. Wang, Z. Li, H. Xiong, and X. Nian, "Robust H_∞ attitude tracking control of a quadrotor UAV on $SO(3)$ via variation-based linearization and interval matrix approach," *ISA Trans.*, vol. 87, pp. 10–16, Apr. 2019.
- [14] C. Zhao and L. Guo, "Control of nonlinear uncertain systems by extended PID," *IEEE Trans. Autom. Control*, vol. 66, no. 8, pp. 3840–3847, Aug. 2021.
- [15] Y. Zhang, T. Chen, S. Chen, and Z. Wang, "Assess-ranging-fuse: Obstacle detection based on stereo vision threat assessment and ultrasonic ranging," in *Proc. Chin. Autom. Congr. (CAC)*, Nov. 2020, pp. 6186–6191.
- [16] D. Zhou, A. Al-Durra, I. Matraji, A. Ravey, and F. Gao, "Online energy management strategy of fuel cell hybrid electric vehicles: A fractional-order extremum seeking method," *IEEE Trans. Ind. Electron.*, vol. 65, no. 8, pp. 6787–6799, Aug. 2018.
- [17] J. G. Ziegler and N. B. Nichols, "Optimum settings for automatic controllers," *J. Dyn. Syst., Meas., Control*, vol. 115, no. 2B, pp. 220–222, Jun. 1993.
- [18] N. J. Killingsworth and M. Krstić, "PID tuning using extremum seeking: Online, model-free performance optimization," *IEEE Control Syst. Mag.*, vol. 26, no. 1, pp. 70–79, Feb. 2006.
- [19] Q. Wei, L. Zhu, R. Song, P. Zhang, D. Liu, and J. Xiao, "Model-free adaptive optimal control for unknown nonlinear multiplayer nonzero-sum game," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 2, pp. 879–892, Feb. 2022.
- [20] H. Wang, K. Xu, and J. Qiu, "Event-triggered adaptive fuzzy fixed-time tracking control for a class of nonstrict-feedback nonlinear systems," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 68, no. 7, pp. 3058–3068, Jul. 2021.
- [21] Q. Hou and J. Dong, "Cooperative fault-tolerant output regulation of linear heterogeneous multiagent systems via an adaptive dynamic event-triggered mechanism," *IEEE Trans. Cybern.*, vol. 53, no. 8, pp. 5299–5310, Aug. 2023.
- [22] J. Dong, Y. Wu, and G.-H. Yang, "A new sensor fault isolation method for T-S fuzzy systems," *IEEE Trans. Cybern.*, vol. 47, no. 9, pp. 2437–2447, Sep. 2017.
- [23] Z. Cao, H.-B. Dürr, C. Ebenbauer, F. Allgöwer, and F. Gao, "Iterative learning and extremum seeking for repetitive time-varying mappings," *IEEE Trans. Autom. Control*, vol. 62, no. 7, pp. 3339–3353, Jul. 2017.
- [24] Y. Chen, D. Huang, N. Qin, and Y. Zhang, "Adaptive iterative learning control for a class of nonlinear strict-feedback systems with unknown state delays," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Dec. 31, 2021, doi: [10.1109/TNNLS.2021.3136644](https://doi.org/10.1109/TNNLS.2021.3136644).
- [25] W. He, T. Meng, X. He, and C. Sun, "Iterative learning control for a flapping wing micro aerial vehicle under distributed disturbances," *IEEE Trans. Cybern.*, vol. 49, no. 4, pp. 1524–1535, Apr. 2019.
- [26] P. K. Muthusamy, M. Garratt, H. Pota, and R. Muthusamy, "Real-time adaptive intelligent control system for quadcopter unmanned aerial vehicles with payload uncertainties," *IEEE Trans. Ind. Electron.*, vol. 69, no. 2, pp. 1641–1653, Feb. 2022.
- [27] J. Ma, H. Wang, and J. Qiao, "Adaptive neural fixed-time tracking control for high-order nonlinear systems," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Jun. 6, 2022, doi: [10.1109/TNNLS.2022.3176625](https://doi.org/10.1109/TNNLS.2022.3176625).
- [28] C. Liu, C. Dong, Z. Zhou, and Z. Wang, "Barrier Lyapunov function based reinforcement learning control for air-breathing hypersonic vehicle with variable geometry inlet," *Aerosp. Sci. Technol.*, vol. 96, Jan. 2020, Art. no. 105537.
- [29] M. O'Connell et al., "Neural-fly enables rapid learning for agile flight in strong winds," *Sci. Robot.*, vol. 7, no. 66, May 2022, Art. no. eabm6597.
- [30] W. Bao, Y. Zhang, and Z. Qi, "Thoughts on the development of intelligent control technology," *Scientia Sinica Informationis*, vol. 50, no. 8, pp. 1267–1272, Aug. 2020.
- [31] W. Bao, "Present situation and development tendency of aerospace control techniques," *Acta Automatica Sinica*, vol. 39, pp. 697–702, Jan. 2014.
- [32] R. Mahony, V. Kumar, and P. Corke, "Multirotor aerial vehicles: Modeling, estimation, and control of quadrotor," *IEEE Robot. Autom. Mag.*, vol. 19, no. 3, pp. 20–32, Sep. 2012.
- [33] L. Guo, "Feedback and uncertainty: Some basic problems and results," *Annu. Rev. Control*, vol. 49, no. 2, pp. 27–36, May 2020.

- [34] A. Safaei, "Cooperative adaptive model-free control with model-free estimation and online gain tuning," *IEEE Trans. Cybern.*, vol. 52, no. 9, pp. 8642–8654, Sep. 2022.
- [35] Y. Bengio, N. Boulanger-Lewandowski, and R. Pascanu, "Advances in optimizing recurrent networks," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, May 2013, pp. 8624–8628.
- [36] O. A. Somefun, K. Akingbade, and F. Dahunsi, "The dilemma of PID tuning," *Annu. Rev. Control*, vol. 52, pp. 65–74, Jun. 2021.
- [37] C. Zhao and L. Guo, "PID controller design for second order nonlinear uncertain systems," *Sci. China Inf. Sci.*, vol. 60, no. 2, Jan. 2017, Art. no. 022201.



Yanhui Zhang (Member, IEEE) was born in Anhui, China, in 1991. He received the M.S. degree in control engineering from the Nanjing University of Aeronautics and Astronautics University, Nanjing, China, in 2017. He is currently pursuing the Ph.D. degree in aerospace science and technology with the School of Aeronautics and Astronautics, Zhejiang University, Hangzhou, China. His research interests include adaptive learning control, bionic flow control, and near-space hypersonic vehicle motion planning and control. He serves as a reviewer for the

China Automation Conference (CAC), the China Control Conference (CCC), and the Data Driven and Learning Control Conference (CCDLS).



Yuxin Yang was born in Liaoning, China, in 1998. She received the B.S. degree in new energy science and engineering from North China Electric Power University, Beijing, China, in 2020. She is currently pursuing the Ph.D. degree in aerospace science and technology with the School of Aeronautics and Astronautics, Zhejiang University, Hangzhou, China. Her research interests include aircraft design, parametric design, and fast aerodynamic prediction.



He is the Winner of the Second Prize of the National Defense Science and Technology Award of the Science and Technology Commission of the Chinese Military Commission.

Weifang Chen (Member, IEEE) was born in Hunan, China, in 1970. He received the B.S. and Ph.D. degrees in aerodynamics from the National Defense University of Science, Hunan, China, in 1991 and 2008, respectively, under Mr. Hesun Cao (member of the International Academy of Astronautics). Since 2010, he has been the Center Director of the NearSpace Vehicle Research Center, Zhejiang University. He is currently a Chinese Military Expert Group Member and the Director of the School of Aeronautics and Astronautics, Zhejiang University.



Hua Yang was born in Ningbo, China, in 1987. He received the B.S. degree in aircraft design from Beihang University, Beijing, China, in 2010, and the Ph.D. degree in aircraft design from Beihang University, Beijing, in 2015. He is currently an Associate Researcher with the School of Aeronautics and Astronautics, Zhejiang University, Hangzhou, China. His current research interests include aircraft conceptual design, flight dynamics simulation, and aerodynamic optimization.