

Received 15 July 2022, accepted 21 July 2022, date of publication 27 July 2022, date of current version 3 August 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3194276

## RESEARCH ARTICLE

# Fast Model-Free Learning for Controlling a Quadrotor UAV With Designed Error Trajectory

CHEN AN<sup>1,2</sup>, SHENGDE JIA<sup>3</sup>, JIAXI ZHOU<sup>1,2</sup>, AND CHANG WANG<sup>1,3</sup>

<sup>1</sup>State Key Laboratory of Advanced Design and Manufacturing for Vehicle Body, Hunan University, Changsha 410082, China

<sup>2</sup>College of Mechanical and Vehicle Engineering, Hunan University, Changsha 410082, China

<sup>3</sup>College of Intelligence Science, National University of Defense Technology, Changsha 410073, China

Corresponding author: Shengde Jia (jiashengde08@nudt.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61806217 and Grant 61906203.

**ABSTRACT** Traditional model-based control methods typically require accurate system dynamics. However, when controlling a complex non-linear system such as a quadrotor unmanned aerial vehicle (QUAV), the dynamics are unknown and it is challenging to tune the control parameters manually. This paper proposes a novel model-free learning method that combines the advantages of a model-based method, i.e., sliding mode control (SMC), with the iterative learning control (ILC) method. Specifically, we selected a designed sliding surface to obtain the expected tracking error trajectory as the learning objective, and the system tracking errors of the angles of the QUAV constitute the state space. Then, the policy of converging to the sliding surface is learned by an ILC algorithm. We have provided theoretical proof of the convergence, and validated the proposed method with real-world experiments where the sine wave signals of roll and pitch angles were tracked. The results have demonstrated the effectiveness of the method with less tracking errors as well as faster learning speed compared with a baseline PID controller and a sliding mode controller.


**INDEX TERMS** UAV, sliding mode control, iterative learning control, non-linear control, model-free.

## I. INTRODUCTION

In the past decades, a variety of control methods have been developed for complex non-linear systems, e.g., Lyapunov method for controlling voltage source converter [1] and uncertain fractional-order systems [2], sliding model control for controlling wireless sensor networks [3], hybrid switched systems [4] and stochastic singular semi-Markov jump systems [5], back-stepping for controlling surface vessels [6] and non-smooth non-linear systems [7], and Proportion Integration Differentiation (PID) for controlling DC motors [8] and fluid transportation systems [9]. Many of the above methods are model-based that assume the accurate system dynamics available. However, this is not applicable for controlling many real-time robotic systems in uncertain and dynamic environments. The PID controller has been proved useful for

controlling a quadrotor UAV (QUAV) in a landing task [10] and a transportation task [11]. It is a closed-loop control method that calculates the differences between the desired trajectory and a measured variable, using the proportional, integral and derivative terms of corrections based on real-time tracking errors, without the need of an accurate system model. PID has the advantage of stability while it suffers the problem of manual parameter tuning. Although automatic parameter tuning can be realized by combining PID with other methods such as reinforcement learning [12], [13], it cannot guarantee policy convergence if there exists constant system errors.

Sliding mode control (SMC) [14] is another choice for the control of a QUAV. One important characteristic of SMC is that the desired system trajectory could be designed as a sliding surface [15], so that the system trajectories are forced into a reduced-order subspace and then held there sliding along the surface. Besides, a feedback control law guarantees the system trajectory intersects and stays on the manifold

The associate editor coordinating the review of this manuscript and approving it for publication was Guillermo Valencia-Palomo .

or sliding surface. Once the trajectory reached the sliding surface, the system would switch to the sliding mode. For example, Vazquez-Nicolas *et al.* [16] used SMC to realize robust altitude control. They defined a sliding surface which was the desired altitude error trajectory, and combined SMC with the PD controller to improve the UAV's performance. Chen *et al.* [17] developed a regular SMC controller to control the attitude subsystem (Inner loop) to guarantee fast convergence of Euler angles. They also defined a sliding manifold as the desired attitude error trajectory. They applied the back-stepping technique to the position control loop until the desired attitudes were obtained. Rossomando *et al.* [18] developed a neural sliding mode controller to perform a dynamic compensation for possible parametric uncertainties to realize formation control. They defined a desired error trajectory as the sliding surface. Then, a model-based control law was obtained to force the tracking errors to reach the sliding surface. Yang *et al.* [19], [20] overcame the main obstacle in sliding-mode surfaces analysis and designed a new adaptive controller for the underactuated systems. No doubt such new approaches will be a solid foundation for the expansion of SMC in the follow-up studies. Another work [21] has proposed an algorithm which called model-free sliding mode control, but the approach actually uses an estimated online model to formulate the SMC design. It is still challenging to find a satisfying feedback control law with good stability because many control laws include sort of system dynamics which are not always available or inaccurate.

Iterative learning control (ILC) can be used with SMC to handle unknown system dynamics which can also be called model-free ILC (MFILC). For example, the accuracy of parameter estimation can be improved [22] or the external disturbances [23] can be handled by using ILC. For the translational and rotational motion control of a QUAV, Allahverdy *et al.* [24] used back-stepping integral SMC to track desired trajectories and developed ILC to improve the robustness of the control strategy. However, the above approaches rely on estimated model parameters to construct accurate SMC controllers, which cannot always be guaranteed due to stochastic external disturbance. Different from reference [24], the proposed algorithm does not obtain system parameters to formulate controllers. As for other applications using the model-free ILC method, Riccardo *et al.* [25] used the ILC method to estimate the system parameters to formulate a torque-based ILC scheme. Bogdan and Precup [26] used a hierarchical control system structure in which the optimality of the behavior is formulated as a reference trajectory tracking problem. The learning solutions which are obtained by MFILC are stored in a library and the reference input primitives are optimized by the higher hierarchical level of the control system structure. Xiaodong *et al.* [27] proposed two adaptive ILC algorithms to solve nonlinear control problems with non-parametric uncertainties. The memory space in implementations is reduced. Eddine *et al.* [28] proposed an ILC controller which works under the practical alignment condition and noncyclic desired trajectories. It should be

**TABLE 1.** Comparison of several control algorithms.

Algorithms	Designable Error Behavior	Model-free
LQR	×	×
PID	×	✓
Q-learning	×	✓
SMC	✓	×
SSIL(this paper)	✓	✓

noted that, though our methodology in this work is in a similar spirit as [27], this paper simplifies the existing work and applies the simplified controller to control the QUAV system in the real world.

Control methods such as PID [29], SMC [30], back-stepping [31], Q-learning [32], LQR [33] are either model-based or slow to obtain an appropriate control law to satisfy the desired tracking error trajectory. Compared with the traditional Q-learning method to control the QUAV system, the proposed algorithm of the training process to obtain the control law is faster. The motivation of this paper is to develop a model-free learning method that supports the design of the system error behavior, meanwhile it guarantees fast convergence of system tracking errors. Instead of applying ILC to estimate the model parameters as well as the external disturbance to construct accurate SMC controllers, we use ILC to learn an appropriate control law to force the system tracking error towards the sliding surface.

The main contributions of this paper are as follows.

We propose a Sliding Surface-based iterative learning (SSIL) method that is designable and model-free (see Table 1), which combines the advantages of sliding mode control (SMC) and iterative learning control (ILC). An appropriate control law can be obtained without the need to identify the system dynamics or model parameters which is more convenient than the existing SMC method. Different from designing the adaptive controller, the proposed algorithm uses iteration learning to obtain the final control law. If the environment has changed, the proposed algorithm also can obtain appropriate control law by iteration learning.

We design a novel unknown non-linear system by taking the original system's tracking error as the state of the ILC algorithm, which satisfies the requirement of consistent initial condition for ILC.

Inspired by the work of [27], we simplify the structure of the controller and provide the theoretical proof of convergence of the tracking errors to the target, and we demonstrate fast convergence of the control law through real-world experiments with a quadrotor UAV.

This paper is organized as follows. Section II formulates the problem and briefly introduces the preliminaries of nonlinear control. Section III provides the details of the proposed method with theoretical proof of convergence, followed by simulation experiments in Section IV as well as physical experiments in Section V. Section VI discusses the

proposed algorithm. Finally, Section VII concludes the paper and outlines the future work.

## II. PRELIMINARIES

In the sequel, we introduce the non-linear control problem and sliding mode control (SMC) for solving it. Consider the continuous-time non-linear affine systems described by:

$$\begin{aligned} \dot{\mathbf{x}}_1(t) &= \mathbf{x}_2(t) \\ \dot{\mathbf{x}}_2(t) &= \mathbf{f}(\mathbf{x}(t)) + \mathbf{g}\mathbf{u}(t) + \mathbf{w} \end{aligned} \quad (1)$$

where  $\mathbf{x}(t) \in \mathbb{R}^n$  is the system state vector, and  $\mathbf{u}(t) \in \mathbb{R}^m$  represents the input of the system with physical limitations described as  $\Omega_m = \{\mathbf{u}(t) | \mathbf{u}(t) \in \mathbb{R}^m, |u_i(t)| \leq N, i = 1, \dots, m\}$ . Besides,  $\mathbf{y}(t) \in \mathbb{R}^p$  represents the output vector of the system,  $\mathbf{f}(\mathbf{x}(t)) \in \mathbb{R}^n$  is the unknown drift dynamics,  $\mathbf{g} \in \mathbb{R}^{n \times m}$  is the unknown input matrix.  $\mathbf{w}$  is the external disturbance. For the convenience of further discussion, several assumptions are made as follows:

*Assumption 1:*  $\mathbf{f}(\mathbf{0}) = 0$  and  $\mathbf{f}(\mathbf{x}) + \mathbf{g}(\mathbf{x})\mathbf{u}$  is Lipschitz, and that the non-linear affine system is stabilizable.

*Assumption 2:* There exists an unknown function  $\mathbf{w}_M$  such that  $\|\mathbf{w}\| \leq \mathbf{w}_M$ .

Assumption 1 guarantees that the origin is the equilibrium point of the non-linear system. Assumption 2 guarantees the perturbation of the non-linear system is bounded.

We aim to design a feedback controller to guarantee the globally exponential stability of the closed-loop system. The closed-loop behavior can be achieved by designing a sliding surface as follows:

$$s = \dot{e}_1 + \mu e_1 = 0; \mu > 0. \quad (2)$$

where  $\mathbf{e}$  is the trajectory tracking error, and  $\mu$  is a positive definite constant matrix. Denote the desired trajectory as  $\mathbf{x}_d$ , and then the error can be described as follows:

$$\mathbf{e}_1 = \mathbf{x}_d - \mathbf{x}_1, \mathbf{e}_2 = \dot{\mathbf{x}}_d - \mathbf{x}_2 \quad (3)$$

The sliding surface is designed to approach zero so that the error would also approach zero. The derivative of the error is described as follows:

$$\dot{\mathbf{e}}_1 = \dot{\mathbf{x}}_d - \mathbf{x}_2 \quad (4)$$

We choose the exponential form of the reaching law:

$$\dot{\mathbf{s}} = -\mathbf{k}\mathbf{s} - \varepsilon \text{sgn}(\mathbf{s}) \quad (5)$$

where  $\mathbf{k}$  is a positive definite constant matrix. Finally, the control law can be described as follows:

$$\mathbf{u} = \mathbf{g}^{-1}[\ddot{\mathbf{x}}_d - \mathbf{f} + \mu(\dot{\mathbf{x}}_d - \dot{\mathbf{x}}_1) - \dot{\mathbf{s}}] \quad (6)$$

With an accurate model of the system dynamics  $\mathbf{f}_2$  and the external disturbance  $\mathbf{w}$ , the control law (6) ensures that the sliding mode  $\mathbf{s}$  will approach to 0. If the sliding mode equals to 0 and the desired trajectory satisfies  $\mathbf{x}_d = 0$ , the behavior of the system state  $x_1$  would satisfy  $x_1 = x_1(0)e^{-\mu t}$ . We note that a sliding surface  $\mathbf{s}$  corresponds to a specific error convergence rate. However, due to stochastic dynamics and

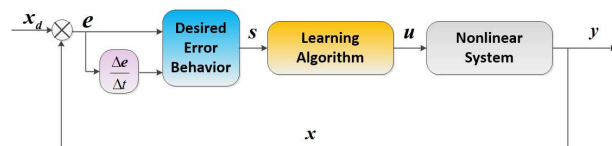


FIGURE 1. The structure of the proposed algorithm.

cost constraints, the model and the external disturbance are difficult or expensive to acquire in some situations. Then, the traditional SMC algorithm can hardly be used to obtain a feasible control policy without the knowledge of system dynamics. In this paper, we develop a novel method to solve this problem by using a learning algorithm to obtain the control policy that guarantees the convergence of the system tracking error.

## III. METHODOLOGY

Specifically, we use an iterative learning algorithm to obtain the control policy. We formulate the error differential equation into a new non-linear system with unknown dynamics. As for the new non-linear system, the error dynamics equation of the primary non-linear system constitutes the new dynamics equation. The output equation is constituted by the sliding surface equation. Then, the original unknown non-linear system control problem is converted into a new non-linear control problem with known desired output. The sliding surface satisfies the design requirement and the system states will approach the sliding surface following an appropriate learning strategy. With the influence of the desired output equation, the novel non-linear system's output  $\mathbf{y}$  will gradually converge to the desired output. Finally, the original system will become stable. The structure of the proposed algorithm is shown in Fig. 1.

### A. SYSTEM DYNAMICS STRUCTURE

In this paper, we assume that the non-linear system dynamics are unknown and continuous. The error equations are shown as follows:

$$\dot{\mathbf{e}}(t) = \dot{\mathbf{x}}_d - \mathbf{f}(\mathbf{x}(t)) - \mathbf{g}(\mathbf{x}(t))\mathbf{u}(t) - \mathbf{w} \quad (7)$$

where  $\mathbf{e}(t) \in \mathbb{R}^n$  is the system state vector available through measurement, and  $\mathbf{u}(t) \in \mathbb{R}^m$  represents the inputs of the non-linear constrained by  $\Omega_m = \{\mathbf{u}(t) | \mathbf{u}(t) \in \mathbb{R}^m, |u_i(t)| \leq N, i = 1, \dots, m\}$ . The states of the new non-linear system are the tracking error of the original non-linear system. The output vector is designed to be the sliding mode of the original non-linear system. The desired output equations are as follows:

$$\begin{aligned} \mathbf{y}(t) &= \mathbf{s} = \dot{e}_1 + \mu \mathbf{e}_1 \\ \mathbf{y}_d(t) &= 0 \end{aligned} \quad (8)$$

The error between the desired output and the actual output of the system (9) are described as follows

$$\Delta \mathbf{y} = \mathbf{y} - \mathbf{y}_d = \mathbf{s} \quad (9)$$

We set the sliding mode  $s$  as the output equation of the novel non-linear system. The main task is to find an appropriate control law acting on the original non-linear system to force the tracking error approach to the sliding surface. Then the new output equation  $\mathbf{y}$  will gradually approach the desired output. When the output of the new non-linear system (7) equals to 0, the tracking error of the original non-linear system (1) will converge exponentially.

$$\lim_{t \rightarrow \infty} \Delta \mathbf{y} = \lim_{t \rightarrow \infty} (\dot{e} + \mu \mathbf{e}) = 0 \quad (10)$$

*Assumption 3:* The control input matrix  $g$  is invertible and is positive definite.  $g^{-1}$  is assumed to be differentiable and bounded.

$$\|g^{-1}\|_{m\infty} \leq l_b$$

where  $l_b$  is an unknown constant.  $\|A_{n \times n}\|_{m\infty} = n \max |a_{ij}|$ .

*Assumption 4:* The alignment condition holds for all iterations.

$$x_{i-1}(T) = x_k(0)$$

*Assumption 5:* The nonlinear function  $\mathbf{f}(\mathbf{x}(t))$  satisfies the linear growth condition

$$\|\mathbf{f}(\mathbf{x}(t))\|_{\infty} \leq c_1 \|\mathbf{x}(t)\|_{\infty} + c_2$$

where  $c_1$  and  $c_2$  are positive unknown constants.  $\|x\|_{\infty} = \max |x_i|$

*Assumption 6:* The desired trajectory  $\mathbf{f}(\mathbf{x}_d)$  and its  $n$ th ( $n = 1, 2$ ) derivations are continuous and bounded.

$$\|\mathbf{x}_d\|_{\infty} \leq l_{xr}, \|\dot{\mathbf{x}}_d\|_{\infty} \leq l_x$$

where  $l_{xr}$  is known and positive constant.

*Assumption 7:* System states  $\mathbf{x}(t)$  is bounded.

$$\|\mathbf{x}\|_{\infty} \leq l_x$$

where  $l_x$  is known positive constant confirmed from the safety flight considerations.

Inspired by [27], we proposed the simplified controller and the theoretical proof of convergence of the tracking error.

*Theorem 1:* Consider the novel non-linear system (9) with the desired trajectory  $x_d$  under assumptions 1-7. The control input satisfies the following equation:

$$u_i(t) = Ks_i(t) + s_i(t)\hat{\eta}_i(t) \quad (11)$$

$$\hat{\eta}_i = \hat{\eta}_{i-1}(t) + s_i^T(t)\Gamma s_i(t) \quad (12)$$

where  $i \in \mathbb{N}$  denotes the number of iterations.  $u_i(t)$  denotes the control input in the  $i$ -th iteration.  $\Gamma$  represents the learning gain matrix. The initial control  $\hat{\eta}(0)$  can be set to zero.

Then, the bounded control law (11) ensures the asymptotic closed-loop stability of the uncertain non-linear system (1),  $\lim_{i \rightarrow \infty} \mathbf{e}_i = 0$ .

*Proof:* During the iterative process, we rewrite the dynamic system (7) at each learning iteration as

$$\dot{\mathbf{e}}_i(t) = \dot{x}_{i,d} - \mathbf{f}_i(\mathbf{x}(t)) - \mathbf{g}_i \mathbf{u}_i(t) - \mathbf{w}_i \quad (13)$$

We define a Lyapunov function as

$$L_i = \frac{1}{2} s_i^T g_i^{-1} s_i + \frac{1}{2} \int_0^t \tilde{\eta}_i^T \Gamma^{-1} \tilde{\eta}_i d\tau \quad (14)$$

where,  $\tilde{\eta}_i = \Lambda - \hat{\eta}_i$ ,  $\Lambda = \text{diag}(\eta, \eta)$ .  $\eta$  is defined as

$$\eta = l_b l_{xr} + l_b c_1 l_x + l_b c_2 + l_b w_M + l_b \lambda_M l_{xr} + l_b \lambda_M l_x \quad (15)$$

Consider the difference of  $L_i$  at the  $i$ th iteration.

$$\begin{aligned} \Delta L_i &= \frac{1}{2} s_i^T g_i^{-1} s_i - \frac{1}{2} s_{i-1}^T g_{i-1}^{-1} s_{i-1} \\ &+ \frac{1}{2} \int_0^t \hat{\eta}_i^T \Gamma^{-1} \tilde{\eta}_i d\tau - \frac{1}{2} \int_0^t \tilde{\eta}_{i-1}^T \Gamma^{-1} \tilde{\eta}_{i-1} d\tau \end{aligned} \quad (16)$$

Then we have

$$\begin{aligned} \frac{1}{2} s_i^T g_i^{-1} (x_i(t)) s_i &= \frac{1}{2} s_i^T (0) g_i^{-1} s_i(0) + \int_0^t s_i^T g_i^{-1} \dot{s}_i d\tau \\ &+ \frac{1}{2} \int_0^t s_i^T \frac{dg_i^{-1}}{dt} s_i d\tau \end{aligned} \quad (17)$$

Then, combining (3), (4), (8) and (17), we obtain

$$\begin{aligned} \frac{1}{2} s_i^T g_i^{-1} (x_i(t)) s_i &= \frac{1}{2} s_i^T (0) g_i^{-1} (x_i(0)) s_i(0) \\ &+ \int_0^t s_i^T g_i^{-1} (\ddot{x}_d - f(\mathbf{x}(t)) \\ &- g_i u(t) - w_i + \mu e_2) \tau \\ &+ \frac{1}{2} \int_0^t s_i^T \frac{dg_i^{-1}}{dt} s_i d\tau \end{aligned} \quad (18)$$

Then, we have

$$\begin{aligned} \frac{1}{2} s_i^T g_i^{-1} (x_i(t)) s_i &= \frac{1}{2} s_i^T (0) g_i^{-1} (x_i(0)) s_i(0) \\ &+ \int_0^t s_i^T (g_i^{-1} \ddot{x}_d - g_i^{-1} f(\mathbf{x}(t)) \\ &- u(t) - g_i^{-1} w_i + \mu g_i^{-1} e_2 \\ &+ \frac{1}{2} \frac{dg_i^{-1}}{dt} s_i) d\tau \end{aligned} \quad (19)$$

The input matrix is a constant matrix, then  $\frac{dg_i^{-1}}{dt} = 0$ . The following inequality is obtained

$$\begin{aligned} &s_i^T (g_i^{-1} \ddot{x}_d - g_i^{-1} f(\mathbf{x}(t)) - g_i^{-1} w_i + \mu g_i^{-1} e_2) \\ &\leq \|s_i^T\|_{\infty} (\|g_i^{-1} \ddot{x}_d - g_i^{-1} f(\mathbf{x}(t)) - g_i^{-1} w_i + \mu g_i^{-1} e_2\|_{\infty}) \\ &\leq \|s_i^T\|_{\infty} (\|g_i^{-1}\|_{m\infty} \|\ddot{x}_d\|_{\infty} + \|g_i^{-1}\|_{m\infty} \|f\|_{\infty} \\ &+ \|g_i^{-1}\|_{m\infty} \|w_i\|_{\infty} + \|\mu g_i^{-1}\|_{m\infty} \|e_2\|_{\infty}) \end{aligned} \quad (20)$$

According to the assumptions (2-7), the following inequality can be obtained

$$\begin{aligned} &s_i^T (g_i^{-1} \ddot{x}_d - g_i^{-1} f(\mathbf{x}(t)) - g_i^{-1} w_i + \mu g_i^{-1} e_2) \\ &\leq \|s_i^T\|_{\infty} (\|g_i^{-1}\|_{m\infty} \|\ddot{x}_d\|_{\infty} + \|g_i^{-1}\|_{m\infty} (c_1 \|x(t)\|_{\infty} + c_2) \\ &+ \|g_i^{-1}\|_{m\infty} \|w_i\|_{\infty} + \|\mu g_i^{-1}\|_{m\infty} \|e_2\|_{\infty}) \end{aligned}$$

$$\begin{aligned} &\leq \|s_i^T\|_\infty(l_b l_{x_r} + l_b c_1 l_x + l_b c_2 + l_b w_M \\ &\quad + l_b \lambda_M l_{x_r} + l_b \lambda_M l_x) \\ &\leq \eta \|s_i^T\|_\infty \leq \eta s_i^T s_i \end{aligned} \quad (21)$$

The last term of the difference of Lyapunov function can also be described

$$\begin{aligned} &\frac{1}{2} \int_0^t \tilde{\eta}_i^T \Gamma^{-1} \tilde{\eta}_i d\tau - \frac{1}{2} \int_0^t \tilde{\eta}_{i-1}^T \Gamma^{-1} \tilde{\eta}_{i-1} d\tau \\ &= \int_0^t (\hat{\eta}_i - \eta)^T \Gamma^{-1} (\hat{\eta}_i - \hat{\eta}_{i-1}) d\tau \\ &\quad - \frac{1}{2} (\hat{\eta}_i - \hat{\eta}_{i-1})^T \Gamma^{-1} (\hat{\eta}_i - \hat{\eta}_{i-1}) d\tau \end{aligned} \quad (22)$$

By substituting the updating law (12) into (22), we get

$$\begin{aligned} &\frac{1}{2} \int_0^t \tilde{\eta}_i^T \Gamma^{-1} \tilde{\eta}_i d\tau - \frac{1}{2} \int_0^t \tilde{\eta}_{i-1}^T \Gamma^{-1} \tilde{\eta}_{i-1} d\tau = \\ &\quad - \int_0^t s_i^T \tilde{\eta}_i s_i - \frac{1}{2} s_i^T s_i^T \Gamma^T s_i s_i d\tau \end{aligned} \quad (23)$$

Then, the difference of  $L_i$  can also be described as

$$\begin{aligned} \Delta L_i &\leq \frac{1}{2} s_i^T(0) g_i^{-1}(x_i(0)) s_i(0) + \int_0^t s_i^T [\eta s_i - u(t) - \tilde{\eta} s_i] d\tau \\ &\quad - \int_0^t \frac{1}{2} s_i^T s_i^T \Gamma^T s_i s_i d\tau - \frac{1}{2} s_{i-1}^T g_{i-1}^{-1} s_{i-1} \end{aligned} \quad (24)$$

Substituting the control law (11) into (24) yields

$$\begin{aligned} \Delta L_i &\leq \frac{1}{2} s_i^T(0) g_i^{-1}(x_i(0)) s_i(0) - \int_0^t s_i^T K s_i d\tau \\ &\quad - \int_0^t \frac{1}{2} s_i^T s_i^T \Gamma^T s_i s_i d\tau - \frac{1}{2} s_{i-1}^T g_{i-1}^{-1} s_{i-1} \end{aligned} \quad (25)$$

Note that  $L_i$  can be represented as follows

$$L_k = L_1 + \sum_{i=2}^k \Delta L_i \quad (26)$$

Hence, substituting (25) into (26) yields

$$\begin{aligned} L_k &\leq L_1 + \sum_{i=2}^k \left[ \frac{1}{2} s_i^T(0) g_i^{-1}(x_i(0)) s_i(0) \right. \\ &\quad + \int_0^t s_i^T [\eta s_i - u(t) - \tilde{\eta} s_i] d\tau \\ &\quad \left. - \int_0^t \frac{1}{2} s_i^T s_i^T \Gamma^T s_i s_i d\tau - \frac{1}{2} s_{i-1}^T g_{i-1}^{-1} s_{i-1} \right] \end{aligned} \quad (27)$$

Based on Assumption 4, we can get

$$\frac{1}{2} s_i(0)^T g_i^{-1} s_i(0) = \frac{1}{2} s_{i-1}(T)^T g_{i-1}^{-1} s_{i-1}(T) \quad (28)$$

Then, inequality (27) can be rewritten as

$$L_k \leq L_1 - \sum_{i=2}^k K_m \int_0^t \|s_i\|^2 d\tau - \sum_{i=2}^k \frac{1}{2} \Gamma_m \int_0^t \|s_i^T s_i\| d\tau \quad (29)$$

where  $K_m$  and  $\Gamma_m$  represent the minimum eigenvalue of the matrix  $K$  and  $\Gamma$  respectively.

According to the definition of Lyapunov function  $L_i$ ,  $L_1$  can be described as

$$L_1 = \frac{1}{2} s_1^T g_1^{-1} s_1 + \frac{1}{2} \int_0^t \tilde{\eta}_1^T \Gamma^{-1} \tilde{\eta}_1 d\tau \quad (30)$$

The time derivative of  $L_1$  is represented by

$$\dot{L}_1 = \frac{1}{2} s_1^T g_1^{-1} \dot{s}_1 + \frac{1}{2} s_1^T \frac{dg_1^{-1}}{dt} s_1 + \frac{1}{2} \tilde{\eta}_1^T \Gamma^{-1} \dot{\tilde{\eta}}_1 \quad (31)$$

According to (22), (23) and  $\hat{\eta} = 0$ , we can obtain

$$\begin{aligned} \frac{1}{2} \tilde{\eta}_1^T \Gamma^{-1} \dot{\tilde{\eta}}_1 &= \frac{1}{2} (\tilde{\eta}_1^T \Gamma^{-1} \dot{\tilde{\eta}}_1 - \tilde{\eta}_0^T \Gamma^{-1} \dot{\tilde{\eta}}_0) + \frac{1}{2} \tilde{\eta}_0^T \Gamma^{-1} \dot{\tilde{\eta}}_0 \\ &= -s_1^T s_1 \tilde{\eta}_1 - \frac{1}{2} s_1^T s_1^T \Gamma^T s_1 s_1 + \frac{1}{2} \eta^T \Gamma^{-1} \dot{\eta} \end{aligned} \quad (32)$$

Therefore

$$\dot{L}_1 \leq -s_1^T s_1 \tilde{\eta}_1 - \frac{1}{2} s_1^T s_1^T \Gamma^T s_1 s_1 + \frac{1}{2} \eta^T \Gamma^{-1} \dot{\eta} \leq \frac{1}{2} \eta^T \Gamma^{-1} \dot{\eta} \quad (33)$$

Because  $\eta^T \Gamma^{-1} \dot{\eta}$  is positive in the iteration  $[0, T]$ . Thus, we have

$$L_1 = L_1(0) + \int_0^t \dot{L}_1 d\tau \leq L_1(0) + \int_0^t \frac{1}{2} \eta^T \Gamma^{-1} \dot{\eta} d\tau < \infty \quad (34)$$

Since  $L_1(0) = \frac{1}{2} \mu(x_{d,1}(0) - x_{1,1}(0))^T g \mu(x_{d,1}(0) - x_{1,1}(0))$  is bounded. Thus,  $L_1(t)$  is bounded over  $t \in [0, T]$ . As the  $L_i$  is positive and  $L_1$  is bounded, we can obtain

$$\|s_i\|^2 \rightarrow 0 \quad \text{as } i \rightarrow \infty \quad (35)$$

Then, we can obtain

$$\dot{e}_{1,i} + \mu e_{1,i} \rightarrow 0 \quad \text{as } i \rightarrow \infty \quad (36)$$

Finally, we can obtain

$$e_{1,i} \rightarrow 0 \quad \text{and} \quad e_{2,i} \rightarrow 0 \quad \text{as } i \rightarrow \infty \quad (37)$$

*Remark 2:* Applying the traditional SMC algorithm needs the identification of the system dynamics. To deal with this problem, we have developed a novel unknown non-linear system that the error equation of the original system is used by the iterative learning algorithm to force the error of the original system towards the sliding surface. When the state error of the original system reaches the sliding mode surface, the original system will become stable. The proposed control structure does not need to identify the system dynamics, and the appropriate control law can still be obtained.

## B. SLIDING SURFACE BASED ITERATIVE LEARNING

To implement the SMC algorithm, the prior knowledge of system dynamics is necessary. However, with stochastic dynamics and cost constraints, the complete or partial system dynamics are difficult to acquire. Besides, when the number of system state increases, the dynamics become even more difficult to obtain. In the sequel, we propose an ILC-based

algorithm to solve this problem.  $\delta$  denotes the designed threshold of the tracking error.  $\Gamma$  denotes iterative learning parameters which are obtained by trial and error.  $\mu$  is a designed parameter that decides the error trajectory.

**Algorithm 1** Sliding Surface Based Iterative Learning (SSIL)

**Require:**  $\mu, \Gamma, \delta, K$

**Ensure:**  $\mathbf{u}^*(\mathbf{x})$

- 1: Design  $\mu$  to formulate the sliding surface

$$s_i = \dot{e}_i + \mu e_i; \mu > 0$$

- 2: Obtain the desired output equation and the output of the new non-linear system.

$$\begin{aligned} \mathbf{x}' &= e_i \\ y' &= \dot{e} + \mu e; \mathbf{y}_d = 0; \end{aligned}$$

- 3: Obtain the iterative learning law:

$$\begin{aligned} u_i(t) &= Ks_i(t) + s_i(t)\hat{\eta}_i(t) \\ \hat{\eta}_i &= \hat{\eta}_{i-1}(t) + s_i^T(t)\Gamma s_i(t) \end{aligned}$$

- 4: **if**  $\|e_i\| \leq \delta$  **then**

5:  $\mathbf{u}^*(t) = u_i(t);$

6: **break;**

7: **else**

8:  $i \leftarrow i + 1;$

9: **Go to Step 2;**

10: **end if**

*Remark 3:* Instead of obtaining the accurate system dynamics, Algorithm 1 uses the iterative learning algorithm to solve the non-linear control problem. Specifically, we convert the original non-linear control problem into a novel non-linear control problem with known desired output formalized as the output equation. In other words, the sliding surface  $s$  which satisfies the design criteria is the desired output. We apply the error of the original system equation to be the novel non-linear system. We use iterative learning algorithm to drive the novel state, i.e. the error of the original system, towards the desired output. As a result, the error of the original system can reach the sliding surface. Finally, the system will be stable and the performance of the system will satisfy the designed criteria.

**IV. NUMERICAL SIMULATION**

In this paper, we deal with the angular control of the continuous-time non-linear QUAV system described as follows. The position of the QUAV is denoted by a vector  $\mathbf{x}_1 = [x, y, z]^T$ , measured with respect to the inertial frame. The linear velocities  $\mathbf{x}_2 = [u, v, w]^T$  and angular velocities  $\mathbf{x}_4 = [p, q, r]^T$  are defined with respect to the body frame. The attitude angles are described by a vector  $\mathbf{x}_3 = [\phi, \theta, \psi]^T$ . The model of the QUAV consists of kinematics and dynamics.

$$\dot{\mathbf{x}}_2 = \mathbf{f}_1(\mathbf{x}_2, \mathbf{x}_4) + \mathbf{g}_1(\dot{\mathbf{x}}_1, T(\mathbf{x}_5)) + \mathbf{d}_1 \quad (38)$$

with unknown non-linear term  $\mathbf{d}_1 = [d_{11}, d_{12}, d_{13}]^T$ , including model uncertainties and unmodeled dynamics.

The function  $\mathbf{f}_1(\mathbf{x}_2, \mathbf{x}_4)$  and  $\mathbf{g}_1(\dot{\mathbf{x}}_1, T(\mathbf{x}_5))$  are denoted as follows:

$$\mathbf{f}_1(\mathbf{x}_2, \mathbf{x}_4) = \begin{bmatrix} rv - qw \\ pw - ru \\ qu - pv \end{bmatrix} \quad (39)$$

$$\mathbf{g}_1(\dot{\mathbf{x}}_1, T(\mathbf{x}_5)) = -\frac{1}{m} \begin{bmatrix} D_x \dot{x}^2 \\ D_y \dot{y}^2 \\ D_z \dot{z}^2 + T \end{bmatrix} + \mathbf{R}_1 \mathbf{g} \quad (40)$$

$$\mathbf{R}_1 = \begin{bmatrix} -\sin \theta \\ \cos \theta \sin \phi \\ \cos \theta \cos \phi \end{bmatrix} \quad (41)$$

where  $\mathbf{x}_5 = [w_1, w_2, w_3, w_4]^T$  is the vector of propeller rotation speeds.  $D_i (i = x, y, z)$  represents the air resistance coefficients.  $m$  is the mass of the QUAV.  $T$  is the total thrust with respect to the body frame described by

$$T(\mathbf{x}_5) = \sum_{i=1}^4 b w_i^2 \quad (42)$$

where  $b$  is the thrust factor. The rotational motion dynamics of the QUAV can be described as follows:

$$\dot{\mathbf{x}}_4 = \mathbf{f}_2(\mathbf{x}_4) + \mathbf{f}_3 \mathbf{g}_2(\mathbf{x}_5) + \mathbf{d}_2 \quad (43)$$

with the diagonal matrix  $J_i (i = x, y, z)$  where  $\mathbf{f}_3 = \text{diag}(1/J_x, 1/J_y, 1/J_z)$  is the moment of inertia of the QUAV.  $\mathbf{d}_2 = [d_{21}, d_{22}, d_{23}]^T$  denotes the model uncertainties and measurement noises, and

$$\mathbf{f}_2(\mathbf{x}_4) = \begin{bmatrix} \frac{J_y - J_z}{J_x} qr \\ \frac{J_z - J_x}{J_y} pr \\ \frac{J_x - J_y}{J_z} pq \end{bmatrix} \quad (44)$$

$$\mathbf{g}_2(\mathbf{x}_5) = \begin{bmatrix} lb(-w_2^2 + w_4^2) \\ lb(w_1^2 - w_3^2) \\ d(w_1^2 - w_2^2 + w_3^2 - w_4^2) \end{bmatrix} \quad (45)$$

where  $\mathbf{g}_2(\mathbf{x}_5)$  is the additional moment acted on the QUAV,  $l$  is the distance from the gravity center of QUAV to the gravity center of each propeller rotor,  $b$  is the thrust factor, and  $d$  is the torque coefficient of the propeller.

The kinematics of the QUAV in body-fixed coordinate is given as follows:

$$\dot{\mathbf{x}}_1 = \mathbf{f}_5(\mathbf{x}_3)\mathbf{x}_2 + \mathbf{d}_3 \quad (46)$$

$$\dot{\mathbf{x}}_3 = \mathbf{f}_4(\mathbf{x}_3)\mathbf{x}_4 + \mathbf{d}_4 \quad (47)$$

where  $\mathbf{d}_3$  and  $\mathbf{d}_4$  denote the model uncertainties, i.e., the measurement noises and external disturbances, with

$$\mathbf{f}_4(\mathbf{x}_3) = \begin{bmatrix} 1 & \sin \phi \tan \theta & \cos \phi \tan \theta \\ 0 & \cos \phi & -\sin \phi \\ 0 & \sin \phi / \cos \theta & \cos \phi / \cos \theta \end{bmatrix} \quad (48)$$

$$\mathbf{f}_5(\mathbf{x}_3) = \begin{bmatrix} c_2 c_3 s_1 s_2 c_1 - c_1 c_3 c_1 s_2 c_3 + s_1 s_3 \\ c_2 c_3 s_1 s_2 s_3 + c_1 c_3 c_1 c_2 s_3 - s_1 c_3 \\ -s_2 & s_1 c_2 & c_1 c_2 \end{bmatrix} \quad (49)$$

TABLE 2. Main parameters of the QUAUV.

Parameters	Value	Units	Parameters	Value	Units
$g$	9.806	$m/s^2$	$m$	1.5	$kg$
$b$	1.105e-5	—	$J_x$	1.745e-2	$kg \cdot m^2$
$d$	1.489e-7	—	$J_y$	1.745e-2	$kg \cdot m^2$
$l$	0.225	$m$	$J_z$	3.175e-2	$kg \cdot m^2$
$D_x$	6.579e-2	$N/(m/s)^2$	$u^*$	0	$m/s$
$D_y$	6.579e-2	$N/(m/s)^2$	$v^*$	0	$m/s$
$D_z$	6.579e-2	$N/(m/s)^2$	$w^*$	0	$m/s$
$\phi^*$	0	$rad$	$\Delta u$	10	$m/s$
$\theta^*$	0	$rad$	$\Delta v$	10	$m/s$
$\psi^*$	0	$rad$	$\Delta w$	10	$m/s$
$\Delta\phi$	$\pi/4$	$rad$	$\Delta w_i$	9000	$rpm$
$\Delta\theta$	$\pi/4$	$rad$			
$\Delta\psi$	$\pi/2$	$rad$			

TABLE 3. Parameter settings in numerical simulation.

	P	I	D
Angular Velocity Controller(p)	0.15	0	0
Angular Velocity Controller(q)	0.15	0.002	0
Angular Velocity Controller(r)	0.15	0.001	0

the parameter to be designed. We set  $\mu = 5$ . The error dynamics of angles and angular velocities were reformulated to be the new non-linear system. The output of the new non-linear system was the sliding surface equation.

$$\mathbf{y}(t) = CE = \mu(\mathbf{x}_d - \mathbf{x}_1) + \dot{\mathbf{x}}_d - \mathbf{x}_2 = \mu\mathbf{e}_1 + \mathbf{e}_2 \quad (51)$$

The inner loop controller was designed as follows:

$$\begin{aligned} u_i(t) &= Ks_i(t) + s_i(t)\hat{\eta}_i(t); \\ \hat{\eta}_i &= \hat{\eta}_{i-1}(t) + s_i^T(t)\Gamma s_i(t) \end{aligned} \quad (52)$$

Set  $\Gamma = \text{diag}(0.6, 0.6, 1.5)$ ,  $K = \text{diag}(0.5, 0.5, 0.7)$  and  $\mathbf{u}_0 = [0, 0, 0]^T$ .

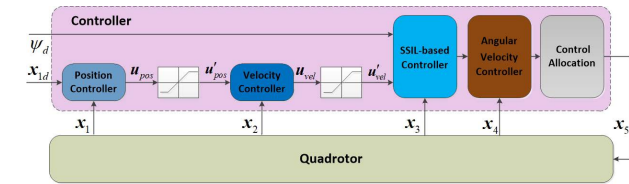


FIGURE 2. The cascaded control framework of a QUAUV, where the output of each loop controller is a function of the state vector of the next loop, and the desired input is bounded to ensure flight safety. Algorithm 1 is used as the angle controller to reduce the tracking error.

where  $c_1 = \cos \phi$ ,  $c_2 = \cos \theta$ ,  $c_3 = \cos \psi$ ,  $s_1 = \sin \phi$ ,  $s_2 = \sin \theta$ , and  $s_3 = \sin \psi$ .

In order to verify the effectiveness of Algorithm 1, sine waves were selected as the desired angle trajectory. The desired angle trajectory was governed by  $\phi_d = \frac{\pi}{12} \sin(\pi t/10)$ ,  $\theta_d = \frac{\pi}{12} \sin(\pi t/10)$ ,  $\psi_d = \frac{\pi}{12} \sin(\pi t/10)$ . The initial condition were as follows:  $\mathbf{x}_1(0) = [0, 0, -100]^T$ ,  $\mathbf{x}_2(0) = [0, 0, 0]^T$ ,  $\mathbf{x}_3(0) = [0, 0, 0]^T$  and  $\mathbf{x}_4(0) = [0, 0, 0]^T$ . The main parameters of QUAUV for simulation are shown in Table 2.

### A. DESIGN OF THE SLIDING SURFACE

We used Algorithm 1 and SMC algorithm to control the inner loop of the QUAUV. The position controller of QUAUV was a proportional controller and the parameters of the controller were adjusted by trial and error. The velocity controller was a PID controller and the parameters of the controller were also adjusted by trial and errors. The main parameters of the angular velocity controllers are shown in Table 3. As for the inner loop controller, the states that needed to be stabilized were the angles and the angular velocities. The structure of the flight controller is shown in Fig. 2. In other words, we designed the sliding surface as follows:

$$\mathbf{s} = \mu\mathbf{e}_1 + \mathbf{e}_2; \mu > 0 \quad (50)$$

where  $\mathbf{e}_1 = [e_\phi, e_\theta, e_\psi]^T$  was the error of angles,  $\mathbf{e}_2 = [e_p, e_q, e_r]^T$  was the error of angular velocities, and  $\mu$  was

### B. SIMULATION RESULTS

The intermediate tracking results for Euler angles are shown in Fig. 3 to Fig. 5. The red line denotes the desired trajectories, and the blue line shows the Euler angles of the QUAUV system at the first iteration. The mean tracking error of Euler angles in each iteration is shown in Fig. 6, which demonstrates the effectiveness of Algorithm 1 that the stability of the inner control loop of the QUAUV can be guaranteed. As the iteration goes on, the tracking error is gradually reduced. The changes of the sliding mode are shown in Fig. 7. The sliding mode  $\mathbf{s}$  gradually approaches 0 that satisfies the equation (11). It was in accordance with the theoretical proof of the convergence of Algorithm 1 in Section III. The tracking performance of the proposed algorithm in the tenth iteration and sliding mode control is shown in Fig. 8 to Fig. 10. The proposed algorithm can make the system stable without applying system parameters and the tracking error is less than that of the sliding mode control algorithm.

### V. REAL-WORLD EXPERIMENTS

#### A. EXPERIMENTAL SETTINGS

In order to verify the proposed method to control a real-world non-linear system, we chose to control the Euler angles of a real QUAUV. We note that the physical parameters of the QUAUV were unknown. Fig. 11 shows the real-world experimental environment.

It consisted of a ground station, a QUAUV platform, and a remote controller. The ground station was responsible for managing the information of the states of the QUAUV as well as for tuning the control parameters. The remote controller was responsible for providing the input signals to initiate Algorithm 1. The hardware of the QUAUV was a four-motor

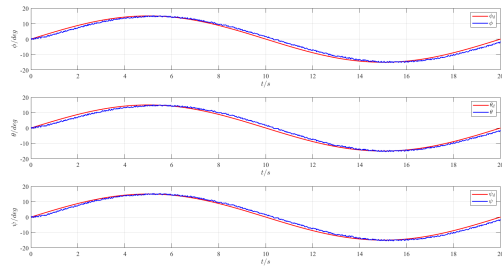


FIGURE 3. Desired and actual Euler angles  $\phi$ ,  $\theta$ , and  $\psi$  in the first iteration.

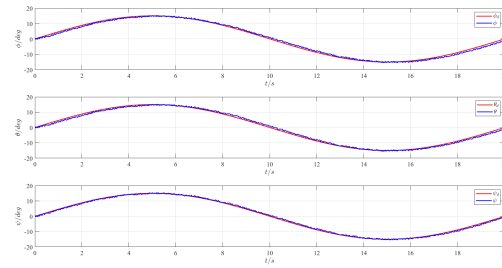


FIGURE 4. Desired and actual Euler angles  $\phi$ ,  $\theta$ , and  $\psi$  in the fifth iteration.

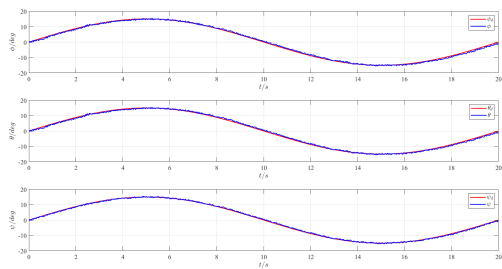


FIGURE 5. Desired and actual Euler angles  $\phi$ ,  $\theta$ , and  $\psi$  in the tenth iteration.

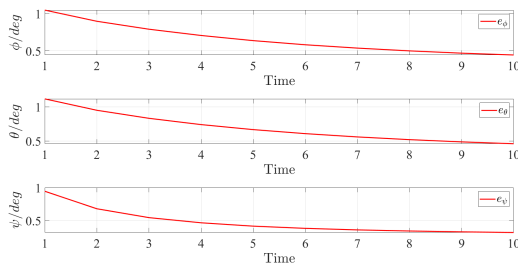


FIGURE 6. Learning performance (as measured by the MAE Euler angles tracking error) for tracking a sine wave.

driven DJI-F330,<sup>1</sup> which had three degrees of freedom. The on-board flight controller was an FMUV5+.<sup>2</sup> The main FMU processor was STM32F765 and the gyroscope in the flight

<sup>1</sup>DJI. F330. Accessed: Aug. 12, 2021. [Online]. Available: <https://www.dji.com/flame-wheel-arf/spec>

<sup>2</sup>PX4 Autopilot. CUAV V5. Accessed: Jan 30, 2020. [Online]. Available: <https://px4.io/cuav-v5-and-v5-nano-now-shipping-with-the-latest-stable-px4-v1-9-preinstalled/>

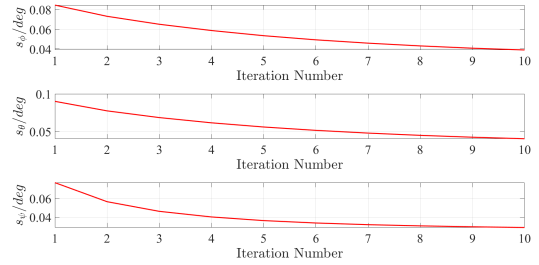


FIGURE 7. Learning performance (as measured by the sliding mode  $s$ ) for tracking a sine wave.

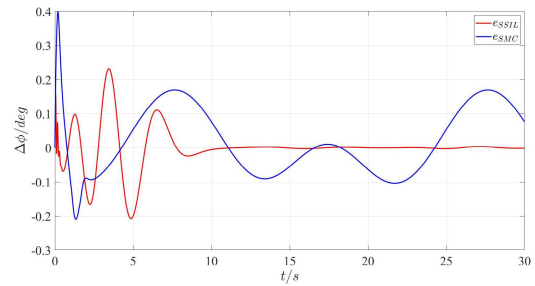


FIGURE 8. Tracking errors of  $\phi$  of the proposed algorithm (SSIL) and sliding mode control (SMC).

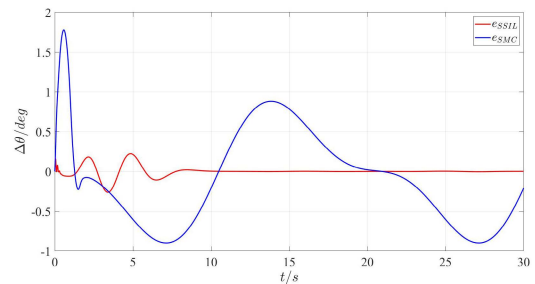


FIGURE 9. Tracking errors of  $\theta$  of the proposed algorithm (SSIL) and sliding mode control (SMC).

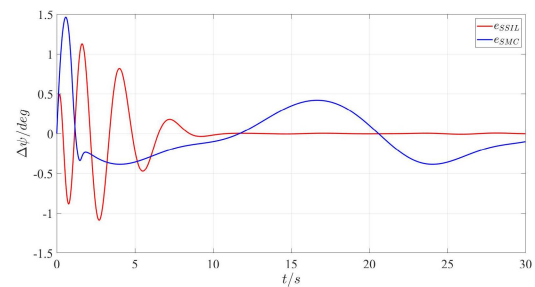


FIGURE 10. Tracking errors of  $\psi$  of the proposed algorithm (SSIL) and sliding mode control (SMC).

controller was ICM-20602. Algorithm 1 ran on the flight controller. The software was implemented using C++.

### B. SLIDING SURFACE DESIGN FOR THE QUAV

A sine curve was provided as the desired roll angle  $\phi_d$  and desired pitch angle  $\theta_d$  that were sent to Algorithm 1. In each



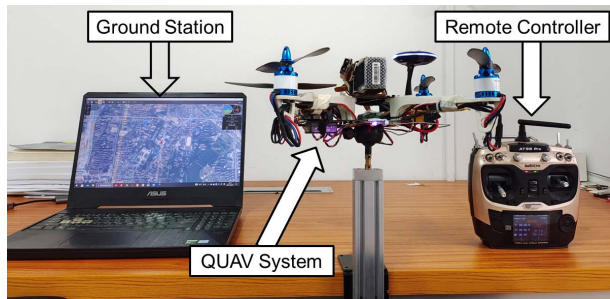


FIGURE 11. Experiment Setup of practical experiment.

TABLE 4. Parameter settings in real-world experiment.

Constant	values
$\mu_1$	260
$\mu_2$	245
$\Gamma_1$	0.02
$\Gamma_2$	0.02
control cycle	10s

iteration, the desired roll angle and pitch angle trajectory were governed by  $\phi_d = -\frac{\pi}{18} \sin(\pi t/5)$  and  $\theta_d = -\frac{\pi}{18} \sin(\pi t/5)$ . After 10 seconds, the desired angle would be 0 until toggling the switch on the remote controller. Considering the safety and reliability of the experiment, the output of the motors would also be set to 0 after 10 seconds. We designed the sliding surface as follows:

$$y = s = \mu e_1 + e_2; \mu > 0 \quad (53)$$

where  $e_1 = [e_\phi, e_\theta]^T$  was the error vector of the roll angle and pitch angles, and  $e_2 = \dot{e}_1$  was the derivative vector accordingly. Considering the adjustment capacity of the QUAV system,  $\mu$  was designed to be  $\mu = \text{diag}(\mu_1, \mu_2)$ . Table 4 shows the numerical settings of the constants used in tests.

The angle controller was designed as follows:

$$\begin{aligned} u_i(t) &= Ks_i(t) + s_i(t)\hat{\eta}_i(t); \\ \hat{\eta}_i &= \hat{\eta}_{i-1}(t) + s_i^T(t)\Gamma s_i(t) \end{aligned} \quad (54)$$

where  $i \in \mathbb{N}$  was the number of iterations,  $\Gamma = \text{diag}(\Gamma_1, \Gamma_2)$ , and the initial control input was  $u_0 = [0, 0]^T$ . Set  $K = \text{diag}(0.5, 0.8)$ .

### C. EXPERIMENTAL RESULTS

The tracking results for the roll and pitch angles are shown in Fig. 14 and Fig. 15. As can be seen, the initial errors were not equal to 0. The attitude of the QUAV system began to respond about 0.5 seconds after the desired signals were provided. Different from the simulation results, the actual system needed a response time to handle the desired signals. This explains why the attitude tracking was not effective at the beginning of the real-world experiment. As shown in Fig. 12, the trend of the desired signals could be roughly tracked, but there existed significant tracking errors in the first iteration.

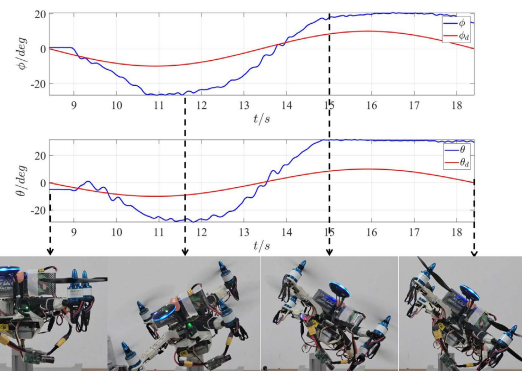


FIGURE 12. Desired and actual roll angle  $\phi$  and pitch angle  $\theta$  in the first iteration.

The tracking results for the roll and pitch angles in the fifth iteration are shown in the fifth subgraph of Fig. 14 and Fig. 15. The tracking errors were obviously reduced, but not converged. In the tenth iteration, as shown in the tenth subgraph of Fig. 14 and Fig. 15, the angles could almost capture the trends of the desired signals and tracking errors were significantly reduced. The mean absolute errors of the roll and pitch angles have been reduced to 5 degrees after 10 iterations.

Due to the sensory noises and the vibration of the experimental apparatus, the tracking errors could not be completely eliminated. The measured Mean Absolute Error (MAE) in the real-world experiment is shown in Fig. 16, and MAE gradually approached to 0. Besides, the mean of the tracking error might slightly increase between the iterations. Overall, the mean of tracking error shows the trend of decreasing towards 0. In other words, we demonstrated that Algorithm 1 could reduce the tracking errors with an appropriate sliding surface. Fig. 17 shows the change of the learning compensation  $\Gamma s_i(t)$  in each iteration. The trend is similar to the variation trend of the sliding surface. This trend is also satisfied the equation (11). This means that the Algorithm 1 can satisfy equation (11) in real-world applications. As we can see from the tracking results, Algorithm 1 can effectively reduce the tracking error.

## VI. DISCUSSION

### A. ROBUSTNESS OF THE PROPOSED SSIL ALGORITHM

Comparing flight data in Fig. 14 and Fig. 15, the initial attitude angle error was different due to the difference of the QUAV placement at the beginning of each experiment. Initial errors in the real-world experiment were shown in Table 5. The initial error was bounded because the QUAV placement did not vary significantly at the beginning of the experiment. It did not satisfy Assumption 5 but experimental results show that the mean of the tracking error generally decreased to 0. We note that the proposed SSIL algorithm can also make the tracking errors of states approach 0 when the initial errors of states are bounded.

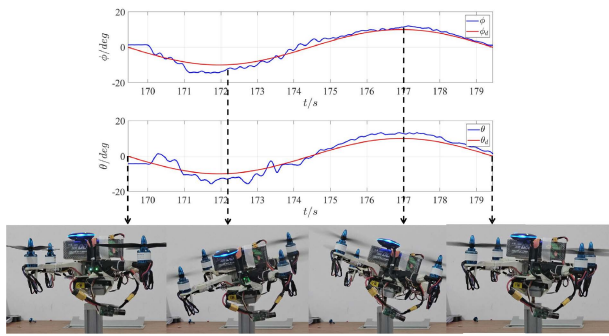


FIGURE 13. Desired and actual roll angle  $\phi$  and pitch angle  $\theta$  in the tenth iteration.

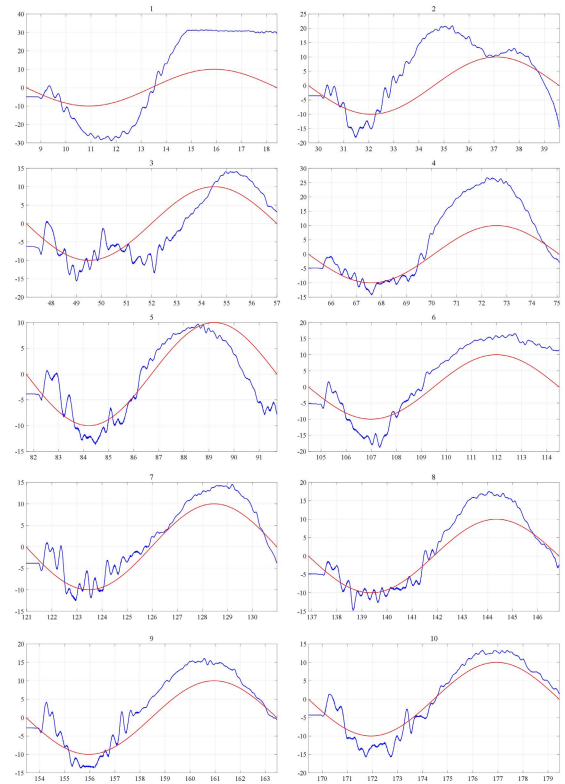


FIGURE 15. Desired and actual pitch angle  $\theta$  in the real-world experiment. The red line denotes the desired roll angle  $\theta_d$ . The blue line denotes the actual roll angle  $\theta$ . The subtitle of this picture  $i$  ( $i = 1, \dots, 10$ ) denotes the number of iterations. The abscissa axis is the time of the real-world experiment and the unit is second. The vertical coordinate represents the pitch angle  $\theta$  and the unit of this coordinate is degree.

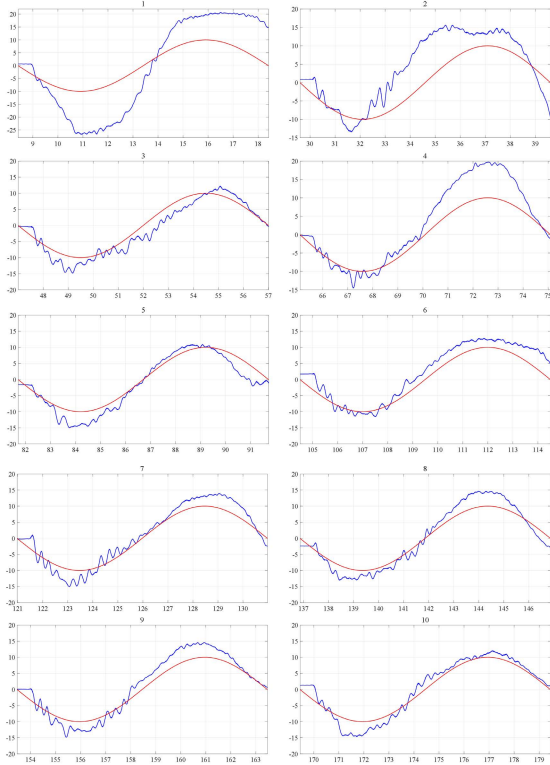


FIGURE 14. Desired and actual roll angle  $\phi$  in the real-world experiment. The red line denotes the desired roll angle  $\phi_d$ . The blue line denotes the actual roll angle  $\phi$ . The subtitle of this picture  $i$  ( $i=1, \dots, 10$ ) denotes the number of iterations. The abscissa axis is the time of the real-world experiment and the unit is second. The vertical coordinate represents the roll angle  $\phi$  and the unit of this coordinate is degree.

**B. THE CHOICE OF DERIVATIVES IN REAL-WORLD EXPERIMENTS**

The changes of the actual roll angular rate in the real-world experiment in the tenth iteration are shown in Fig. 18. Due to the performance of the sensor and external noises, angular velocity changed rapidly. In the real-world experiment,  $e_2 = \dot{e}_1$  was the derivative vector which was used to constitute the sliding surface. Considering the safety of the QUAV system, the angular error was the main influence factor in the sliding surface. Unlike the design of sliding surface with angular

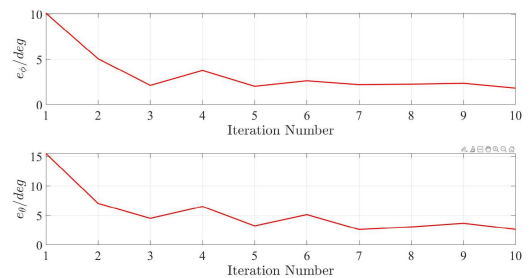


FIGURE 16. Learning performance (as measured by the MAE angles tracking error) for tracking sine wave.

velocity in the numerical simulation, we used derivatives of the error of roll and pitch angles to constitute the sliding surface.

**C. PERFORMANCE OF THE PROPOSED SSIL ALGORITHM**

Compared with PID, Fig. 19 shows that the tracking errors have been significantly reduced after 10 iterations. It demonstrates that the proposed SSIL algorithm can further improve the performance of the PD control law.

When assigning a specific closed-loop behavior, the PD method requires parameter tuning to satisfy the design

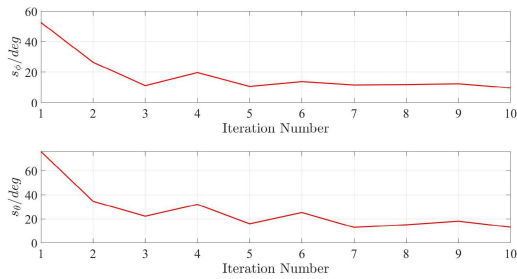


FIGURE 17. Learning compensation values  $\Gamma s_j(t)$  in each iterations.

TABLE 5. Initial error in our ten tests.

	Roll Angle Error/ $^{\circ}$	Pitch Angle Error/ $^{\circ}$
1	0.6546	-4.9597
2	0.8704	-3.5243
3	-0.3222	-6.1958
4	-0.2755	-4.8086
5	-1.5979	-3.8553
6	1.7157	-5.2208
7	-0.2296	-3.8398
8	-2.3844	-4.8437
9	0.0877	-2.7946
10	1.2988	-4.3181

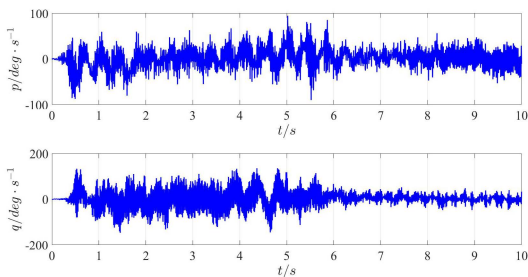


FIGURE 18. Measured angular velocity  $p$  and  $q$  in the tenth iteration.

requirements which can be time consuming. Compared with the PD method, the SSIL designs the sliding surface which satisfies the assigned requirement. With the influence of ILC, the sliding mode  $s$  will approach 0. When the sliding mode  $s$  equals to 0, the tracking errors will converge exponentially to 0. This phenomenon can be seen in Fig. 7 and Fig. 17. Fig. 7 shows that the iterative learning law makes the sliding mode  $s$  approach 0 in the simulation experiment, and Fig. 17 shows that the iterative learning law makes the sliding mode  $s$  approach 0 in the real-world experiment. Due to sensor noises and external disturbances, the change of the sliding mode  $s$  sometimes increases slightly in the real-world experiment. The convergence process of the sliding mode is realized by ILC. This process is approximately diagrammed in Fig. 19. Red arrows in Fig. 19 represent the process of tuning PID parameters. Blue arrows in Fig. 19 represent the process of the iterative learning. In the iterative learning process, the

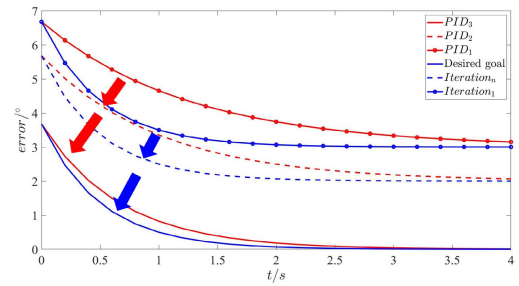


FIGURE 19. Schematic diagram of the error convergence.

sliding mode  $s$  gradually approaches 0, and the tracking error approaches 0.

### VII. CONCLUSION

In this paper, we have proposed a designable and model-free method to solve non-linear control problems. The proposed Sliding Surface based Iterative Learning (SSIL) algorithm combines the advantages of sliding mode control (SMC) and iterative learning control (ILC). The desired tracking error trajectory is defined as the learning objective according to the actual task requirements, and then the tracking error is regarded as the state space of a new non-linear system. A learning algorithm is applied to obtain the control law to force the tracking error to converge to the desired error trajectory. One advantage of the proposed method is that no effort is needed to model the system dynamics or to identify the model parameters. The obtained control law can force the tracking error to converge to the designed desired error trajectory. We use ILC as the learning algorithm and the convergence of the proposed control framework has been proved. The trend of the tracking errors has been shown approaching 0 in both the simulation and the real-world experiments. For future work, we will try other online learning algorithms to enhance the anti-interference ability of the proposed algorithm. Besides, we will also test the proposed algorithm to control a QUAUV in more difficult real-world tasks.

### REFERENCES

- [1] B. Shakerighadi, E. Ebrahimzadeh, and F. Blaabjerg, "Large-signal stability modeling for the grid-connected VSC based on the Lyapunov method," *Energies*, vol. 11, no. 10, pp. 360–365, Aug. 2018.
- [2] E. S. A. Shahri, A. Alfi, and J. A. T. Machado, "Lyapunov method for the stability analysis of uncertain fractional-order systems under input saturation," *Appl. Math. Model.*, vol. 81, pp. 663–672, May 2020.
- [3] S. Qu, L. Zhao, and Z. Xiong, "Cross-layer congestion control of wireless sensor networks based on fuzzy sliding mode control," *Neural Comput. Appl.*, vol. 32, no. 17, pp. 13505–13520, Sep. 2020.
- [4] X. Su, X. Liu, P. Shi, and Y.-D. Song, "Sliding mode control of hybrid switched systems via an event-triggered mechanism," *Automatica*, vol. 90, pp. 294–303, Apr. 2018.
- [5] Q. Wenhai, Z. Guangdeng, and K. Hamidreza, "Sliding mode control for nonlinear stochastic singular semi-Markov jump systems," *IEEE Trans. Autom. Control*, vol. 1, no. 1, pp. 1–8, Jul. 2019.
- [6] G. Wen, S. S. Ge, C. L. P. Chen, F. Tu, and S. Wang, "Adaptive tracking control of surface vessel using optimized backstepping technique," *IEEE Trans. Cybern.*, vol. 49, no. 9, pp. 3420–3431, Sep. 2019.
- [7] J. Yu, P. Shi, and L. Zhao, "Finite-time command filtered backstepping control for a class of nonlinear systems," *Automatica*, vol. 92, pp. 173–180, Jun. 2018.

- [8] Z. Qi, Q. Shi, and H. Zhang, "Tuning of digital PID controllers using particle swarm optimization algorithm for a CAN-based DC motor subject to stochastic delays," *IEEE Trans. Ind. Electron.*, vol. 67, no. 7, pp. 5637–5646, Jul. 2020.
- [9] E. B. Priyanka, C. Maheswaria, S. Thangavela, and M. Ponni Bala, "Integrating IoT with LQR-PID controller for online surveillance and control of flow and pressure in fluid transportation system," *J. Ind. Inf. Integr.*, vol. 17, no. 4, pp. 563–570, Jan. 2020.
- [10] C. Rosales, C. M. Soria, and F. G. Rossomando, "Identification and adaptive PID control of a hexacopter UAV based on neural networks," *Int. J. Adapt. Control Signal Process.*, vol. 33, no. 1, pp. 74–91, Nov. 2018.
- [11] P. E. I. Pounds, D. R. Bersak, and A. M. Dollar, "Stability of small-scale UAV helicopters and quadrotors with added payload mass under PID control," *Auton. Robots.*, vol. 33, nos. 1–2, pp. 129–142, 2012.
- [12] I. Carlucho, M. D. Paula, and G. G. Acosta, "An adaptive deep reinforcement learning approach for MIMO PID control of mobile robots," *ISA Trans.*, vol. 102, pp. 280–294, Jul. 2020.
- [13] M. Gheisamejad and M. H. Khooban, "An intelligent non-integer PID controller-based deep reinforcement learning: Implementation and experimental results," *IEEE Trans. Ind. Electron.*, vol. 68, no. 4, pp. 3609–3618, Apr. 2021.
- [14] X. Yu and O. Kaynak, "Sliding-mode control with soft computing: A survey," *IEEE Trans. Ind. Electron.*, vol. 56, no. 9, pp. 3275–3285, Jul. 2009.
- [15] H. K. Khalil, *Nonlinear Systems*, 3rd ed. Upper Saddle River, NJ, USA: Prentice-Hall, 2002.
- [16] J. M. Vazquez-Nicolas, E. Zamora, I. González-Hernández, R. Lozano, and H. Sossa, "PD+SMC quadrotor control for altitude and crack recognition using deep learning," *Int. J. Control, Autom. Syst.*, vol. 18, no. 4, pp. 834–844, Apr. 2020.
- [17] F. Chen, R. Jiang, K. Zhang, B. Jiang, and G. Tao, "Robust backstepping sliding-mode control and observer-based fault estimation for a quadrotor UAV," *IEEE Trans. Ind. Electron.*, vol. 63, no. 8, pp. 5044–5056, Aug. 2016.
- [18] F. Rossomando, C. Rosales, J. Gimenez, L. Salinas, C. Soria, M. Sarcinelli-Filho, and R. Carelli, "Aerial load transportation with multiple quadrotors based on a kinematic controller and a neural SMC dynamic compensation," *J. Intell. Robot. Syst.*, vol. 100, no. 2, pp. 519–530, Nov. 2020.
- [19] T. Yang, N. Sun, and Y. Fang, "Adaptive fuzzy control for a class of MIMO underactuated systems with plant uncertainties and actuator dead-zones: Design and experiments," *IEEE Trans. Cybern.*, vol. 52, no. 8, pp. 8213–8226, Aug. 2022.
- [20] T. Yang, N. Sun, and Y. Fang, "Neuroadaptive control for complicated underactuated systems with simultaneous output and velocity constraints exerted on both actuated and unactuated states," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Oct. 8, 2021, doi: 10.1109/TNNLS.2021.3115960.
- [21] N. Ebrahimi, S. Ozgoli, and A. Ramezani, "Model-free sliding mode control, theory and application," *Proc. Inst. Mech. Eng., I, J. Syst. Control Eng.*, vol. 232, no. 10, pp. 1292–1301, Oct. 2018.
- [22] E. G. Razmjou, S. K. H. Sani, and J. Sadati, "Robust adaptive sliding mode control combination with iterative learning technique to output tracking of fractional-order systems," *Trans. Inst. Meas. Control*, vol. 40, no. 6, pp. 1808–1818, Apr. 2018.
- [23] J.-S. Lu, M.-Y. Cheng, K.-H. Su, and M.-C. Tsai, "Wire tension control of an automatic motor winding machine—An iterative learning sliding mode control approach," *Robot. Comput.-Integr. Manuf.*, vol. 50, pp. 50–62, Apr. 2018.
- [24] D. Allahverdy, A. Fakharian, and M. B. Menhaj, "Back-stepping integral sliding mode control with iterative learning control algorithm for quadrotor UAVs," *J. Electr. Eng. Technol.*, vol. 14, no. 6, pp. 2539–2547, Aug. 2019.
- [25] R. Mengacci, F. Angelini, M. G. Catalano, G. Grioli, A. Bicchi, and M. Garabini, "On the motion/stiffness decoupling property of articulated soft robots with application to model-free torque iterative learning control," *Int. J. Robot. Res.*, vol. 40, no. 1, pp. 348–374, Aug. 2020.
- [26] M.-B. Radac and R.-E. Precup, "Optimal behaviour prediction using a primitive-based data-driven model-free iterative learning control approach," *Comput. Ind.*, vol. 74, pp. 95–109, Dec. 2015.
- [27] X.-D. Li, M.-M. Lv, and J. K. L. Ho, "Adaptive ILC algorithms of nonlinear continuous systems with non-parametric uncertainties for non-repetitive trajectory tracking," *Int. J. Syst. Sci.*, vol. 47, no. 10, pp. 2279–2289, 2016.
- [28] C. E. Boudjedir, M. Bouri, and D. Boukhetala, "Model-free iterative learning control with nonrepetitive trajectories for second-order MIMO nonlinear systems—Application to a delta robot," *IEEE Trans. Ind. Electron.*, vol. 68, no. 8, pp. 7433–7443, Aug. 2021.
- [29] K. H. Ang, G. Chong, and Y. Li, "PID control system analysis, design, and technology," *IEEE Trans. Control Syst. Technol.*, vol. 13, no. 4, pp. 559–576, Jul. 2005.
- [30] Y. Shtessel et al., *Sliding Mode Control and Observation*. New York, NY, USA: Springer, 2014.
- [31] J. Zhou and C. Wen, *Adaptive Backstepping Control of Uncertain Systems: Nonsmooth Nonlinearities, Interactions or Time-Variations*. Cham, Switzerland: Springer, 2008.
- [32] C. J. C. H. Watkins and P. Dayan, "Q-learning," *Mach. Learn.*, vol. 8, nos. 3–4, pp. 279–292, 1992.
- [33] G. C. Chow, *Analysis and Control of Dynamic Economic Systems*. Hoboken, NJ, USA: Wiley, 1975.



**CHEN AN** received the B.S. degree in automatic control from Central South University, Changsha, China, in 2014, and the M.S. degree from the National University of Defense Technology, Changsha, in 2017. He is currently pursuing the Ph.D. degree with the College of Mechanical and Vehicle Engineering, Hunan University.

His current research interests include reinforcement learning and non-linear control.



**SHENGDE JIA** received the B.S. degree in automatic control from the Harbin Institute of Technology, Harbin, China, in 2008, and the M.S. and Ph.D. degrees from the National University of Defense Technology, Changsha, China, in 2011 and 2015, respectively.

He is currently an Associate Professor with the College of Mechatronic Engineering and Automation, National University of Defense Technology. His current research interests include coordination and control of multiple unmanned aerial vehicles, and machine learning.



**JIAXI ZHOU** received the B.S. degree in civil engineering and the Ph.D. degree in mechanics from Northwestern Polytechnical University, Xian, China, in 2005 and 2010, respectively.

He is currently an Associate Professor with the College of Mechanical and Vehicle Engineering, Hunan University, Changsha, China. He has published over 60 refereed journals in the areas of mechanical vibration and control. His current research interests include vibration isolation, vibration control, non-linear dynamics, and metamaterials.



**CHANG WANG** received the B.S. degree in mathematics from the University of Science and Technology of China (USTC), the M.Sc. degree in applied mathematics from the National University of Defense Technology, China, and the Ph.D. degree in robot learning from the Delft University of Technology, The Netherlands. He is currently an Assistant Professor at the College of Intelligence Science, National University of Defense Technology. His research interests include reinforcement learning, developmental robotics, and human–robot teamwork.

...