*Appl. Math. Mech. -Engl. Ed.*, **45**(4), 603–618 (2024)

**APPLIED MATHEMATICS AND MECHANICS (ENGLISH EDITION)**

https://doi.org/10.1007/s10483-024-3103-8

# Adaptive state-constrained/model-free iterative sliding mode control for aerial robot trajectory tracking*

Chen AN[1,†], Jiaxi ZHOU[1], Kai WANG[1,2]

1. College of Mechanical and Vehicle Engineering, Hunan University, Changsha 410082, China;
2. Department of Mechanical Engineering, The Hong Kong Polytechnic University, Hong Kong, China

(Received Dec. 18, 2023 / Revised Feb. 19, 2024)

**Abstract** This paper develops a novel hierarchical control strategy for improving the trajectory tracking capability of aerial robots under parameter uncertainties. The hierarchical control strategy is composed of an adaptive sliding mode controller and a model-free iterative sliding mode controller (MFISMC). A position controller is designed based on adaptive sliding mode control (SMC) to safely drive the aerial robot and ensure fast state convergence under external disturbances. Additionally, the MFISMC acts as an attitude controller to estimate the unmodeled dynamics without detailed knowledge of aerial robots. Then, the adaption laws are derived with the Lyapunov theory to guarantee the asymptotic tracking of the system state. Finally, to demonstrate the performance and robustness of the proposed control strategy, numerical simulations are carried out, which are also compared with other conventional strategies, such as proportional-integral-derivative (PID), backstepping (BS), and SMC. The simulation results indicate that the proposed hierarchical control strategy can fulfill zero steady-state error and achieve faster convergence compared with conventional strategies.

**Key words** aerial robot, hierarchical control strategy, model-free iterative sliding mode controller (MFISMC), trajectory tracking, reinforcement learning

**Chinese Library Classification** V24
**2010 Mathematics Subject Classification** 93C10, 93C85

## 1 Introduction

A multirotor aerial robot is a typical kind of unmanned aerial vehicle (UAV) that has been widely applied in military and civil applications like surveillance[1], patrolling for power cables[2], aerial cinematography[3], and ground mapping[4–7]. This kind of aerial robot offers significant advantages of the ability of vertical takeoff and landing, and hovering flight. However, the excessive maneuvering ability of the aerial robot may cause structural damage and cracking. It is necessary to resolve this opposite characteristic in the design of flight controllers. Due

* Citation: AN, C., ZHOU, J. X., and WANG, K. Adaptive state-constrained/model-free iterative sliding mode control for aerial robot trajectory tracking. *Applied Mathematics and Mechanics* (*English Edition*), **45**(4), 603–618 (2024) https://doi.org/10.1007/s10483-024-3103-8

† Corresponding author, E-mail: anchen@hnu.edu.cn

to these characteristics, numerous studies have been conducted to improve the efficiency and safety of flight[8–10].

The multirotor aerial robot is a nonlinear system with constrained control inputs and strong coupling between system states, resulting in a significant challenge in controller design. Many control strategies have been investigated. A survey considering the architectural design and control algorithms of a quadrotor was proposed[11–12]. With regard to quadrotors, flight control methods are mainly divided into two major groups: linear control methods and nonlinear control methods[13–15]. Linear control methods applied in quadrotor flight control include the proportional-integral-derivative (PID) controller[16], linear quadratic controller, and $H_\infty$ controller[17–19]. The PID controller is widely applied in commercial quadrotor systems due to its robustness and effectiveness. Bouabdallah et al.[20] designed the PID controller for a quadrotor, and derived the dynamic model of the quadrotor utilizing the Euler-Lagrange method. A high performance and robustness of the linear quadratic regulator controller was proposed by Elkhatem and Engin[21] using the adjusted weighting matrices automatically through the state variables matrix of the quadrotor.

Considering the nonlinear model of the aerial robot, Vazquez-Nicolas et al.[22] proposed a real-time robust altitude control strategy that combines a proportional-derivative controller and a sliding mode controller. The combination of these control actions allows for improving the performance of the aerial robot system. Many extended sliding mode control (SMC) methods have been presented for the flight controller design[23]. Benallegue et al.[24] proposed a sliding mode observer with a feedback linearization controller to avoid the influence of external disturbances and parameter uncertainties and to stabilize the attitude of the quadrotor. Wang et al.[25] proposed an incremental SMC assisted with a designed sliding mode observer. Instead of the traditional model-based nonlinear dynamic inversion, they used the proposed control structure to reduce the model dependency and uncertainties. A gain-scheduling-based SMC law integrated with a novel nonlinear disturbance was proposed for compensating the system uncertainties existing in the actual system[26]. The adaptive nonsingular fast terminal SMC algorithm[27] was proposed to ensure fast convergence and to provide robustness against unknown external disturbances. An adaptive fuzzy gain scheduling method combined with a sliding mode controller was proposed to reduce the chattering phenomenon, so as to stabilize the attitude of the quadrotor system. The above literature mainly discusses the control strategy for external disturbance compensation and gain scheduling. Hua et al.[28] proposed a robust fractional-order sliding mode-based state constrained control scheme to control the aerial robot in the presence of the model uncertainties and wind gust disturbances. An adaptive generalized regression neural network[29] was used to estimate the disturbances and unmodeled dynamics to solve the path tracking problem.

### 1.1 Motivation and contribution

Despite efforts to improve the trajectory tracking capacity of aerial robots in the presence of external disturbances and parameter uncertainties, less attention has been paid to the controller design that takes into account dynamic limitations for safe operations. In this paper, a nonlinear control law for driving below the allowable maximum state errors is considered in order to avoid possible structural failures and to ensure the safety of flight. The main contributions of this paper are briefly summarized as follows.

A novel hierarchical control strategy consisting of an adaptive sliding mode controller and a model-free iterative sliding mode controller (MFISMC) is presented. A neural network-based controller is developed as the position controller of the aerial robot by combining sliding surfaces consisting of constrained states with radial basis function neural networks (RBFNNs), on the basis of which the output weight matrix as well as the center and standard deviation are updated. The convergence of the proposed position controller is proved.

An MFISMC is proposed as the attitude controller of the aerial robot. The proposed controller utilizes a linear approximate model as part of the controller to stabilize both the Euler

angles and angular rates of the aerial robot. The convergence of the proposed algorithm is proved.

New state and action sets and reward functions are proposed to construct reinforcement learning environments to improve the performance of the attitude controller.

### 1.2 Organization

This paper is organized as follows. Section 2 describes the aerial robot dynamics in detail. Section 3 establishes the adaptive position controller and the MFISMC attitude controller. The simulation results are presented in Section 4. The discussion and conclusions are presented in Sections 5 and 6, respectively.

## 2 Aerial robot dynamic model

An aerial robot is illustrated in Fig. 1. Its dynamic model is set up by the body-frame $B\,(O_{\mathrm{b}}x_{\mathrm{b}}y_{\mathrm{b}}z_{\mathrm{b}})$ and the earth-frame $E\,(OX_0Y_0Z_0)$. The body-fixed coordinate $B\,(O_{\mathrm{b}}x_{\mathrm{b}}y_{\mathrm{b}}z_{\mathrm{b}})$ is considered with the origin coinciding with the center of gravity of the aerial robot[30–32].

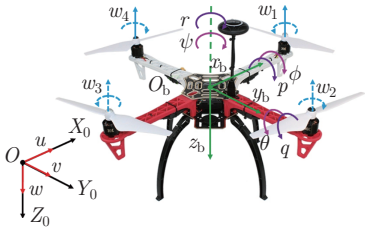


**Fig. 1** Configuration of an aerial robot. The aerial robot consists of a cross rigid frame with four rotors that produce driving forces (color online)

The equation of translational motion of the aerial robot can be described as

$$\dot{x}_2 = f_1(x_2, x_4) + g_1(\dot{x}_1, T) + d_1, \tag{1}$$

where $d_1\,(=[d_{11}, d_{12}, d_{13}]^{\mathrm{T}})$ is the unknown term, including model uncertainties, external disturbances, and unmodeled dynamics. $x_1\,(=[x, y, z]^{\mathrm{T}})$ is the position of the aerial robot, which is measured with respect to the inertial frame. $x_2\,(=[u, v, w]^{\mathrm{T}})$ is the linear velocity of the aerial robot. $x_4\,(=[p, q, r]^{\mathrm{T}})$ is the angular velocity of the aerial robot. The functions $f_1(x_2, x_4)$ and $g_1(\dot{x}_1, T)$ are denoted as follows:

$$f_1(x_2, x_4) = \begin{bmatrix} rv - qw \\ pw - ru \\ qu - pv \end{bmatrix}, \quad g_1(\dot{x}_1, T) = -\frac{1}{m}\begin{bmatrix} D_x\dot{x}^2 \\ D_y\dot{y}^2 \\ D_z\dot{z}^2 + T \end{bmatrix} + \begin{bmatrix} -\sin\theta \\ \cos\theta\sin\phi \\ \cos\theta\cos\phi \end{bmatrix} g, \tag{2}$$

where $D_i\,(i = x, y, z)$ represents the air resistance coefficient and may change as a consequence of weather conditions. $g$ is the gravitational acceleration. $m$ is the mass of the aerial robot. $T\left(=\sum_{i=1}^{4} bw_i^2\right)$ is the total thrust with respect to the body frame. $b\,(>0)$ is the thrust factor. $T$ is one of the control inputs to control the height of the aerial robot. The reactive torque caused by the $i$th rotor in free air is $M_i = -kw_i^2, i = 1, 2, 3, 4$, and $k > 0$. $k$ is a parameter depending on the radius of the propeller, the density of air, the lift coefficients, and the drag coefficients of the blade. The equation of rotational motion of the aerial robot can be described as

$$\dot{x}_4 = f_2(x_4) + Jg_2(x_5) + d_2, \tag{3}$$

where $J\,(=\mathrm{diag}(1/J_x, 1/J_y, 1/J_z))$ is a diagonal matrix, and $J_i\,(i = x, y, z)$ is the moment of inertia of the aerial robot. $d_2\,(=[d_{21}, d_{22}, d_{23}]^{\mathrm{T}})$ denotes the model uncertainties and measure-

ment noises, and

$$f_2(x_4) = \begin{bmatrix} \dfrac{J_y - J_z}{J_x} qr \\ \dfrac{J_z - J_x}{J_y} pr \\ \dfrac{J_x - J_y}{J_z} pq \end{bmatrix}, \quad g_2(x_5) = \begin{bmatrix} lb(-w_2^2 + w_4^2) \\ lb(w_1^2 - w_3^2) \\ -k(w_1^2 - w_2^2 + w_3^2 - w_4^2) \end{bmatrix}, \tag{4}$$

where $x_5\,(=[w_1, w_2, w_3, w_4]^{\mathrm{T}})$ is the vector of propeller rotation speeds. $g_2(x_5)\,(=[M_x, M_y, M_z]^{\mathrm{T}})$ is the additional moment acting on the aerial robot. $g_2(x_5)$ is also the control input to control the torque of the aerial robot. $l$ is the distance from the gravity center of the aerial robot to the gravity center of each propeller rotor.

The kinematic equations of rotational and translational motions are obtained utilizing the rotation matrix. The translational kinematics can be given by $\dot{x}_1 = f_4(x_3)x_2 + d_3$, where $d_3$ denotes the model uncertainties, measurement noises, and external disturbances, $x_3\,(=[\phi, \theta, \psi]^{\mathrm{T}})$ denotes the attitude angle of the aerial robot, and

$$f_4(x_3) = \begin{bmatrix} \cos\theta\cos\psi & \sin\phi\sin\theta\cos\phi - \cos\phi\sin\psi & \cos\phi\sin\theta\cos\psi + \sin\phi\sin\psi \\ \cos\theta\cos\psi & \sin\phi\sin\theta\sin\psi + \cos\phi\cos\psi & \cos\phi\cos\theta\sin\psi - \sin\phi\cos\psi \\ -\sin\theta & \sin\phi\cos\theta & \cos\phi\cos\theta \end{bmatrix}. \tag{5}$$

## 3 Flight controller design

The designed flight controller of an aerial robot is shown in Fig. 2. The control objective for the aerial robot is to track a designed trajectory $x_{1\mathrm{d}} = [x_{\mathrm{d}}, y_{\mathrm{d}}, z_{\mathrm{d}}]^{\mathrm{T}}$ and a desired yaw angle $\psi_{\mathrm{d}}$. In the translational subsystem, the proposed position controller is composed of an adaptive RBFNN. Specifically, the proposed position controller is designed to calculate $T$, $\phi_{\mathrm{d}}$, and $\theta_{\mathrm{d}}$. With the desired yaw angle $\psi_{\mathrm{d}}$, in the rotational subsystem, an MFISMC is used to obtain proper values of torques $M = [M_x, M_y, M_z]^{\mathrm{T}}$, to actuate the aerial robot and track the desired Euler angles $\phi_{\mathrm{d}}$, $\theta_{\mathrm{d}}$, and $\psi_{\mathrm{d}}$. Then, a reinforcement learning algorithm is implemented to improve the performance of the controller. Finally, the Lyapunov theory is applied to the stability analysis.

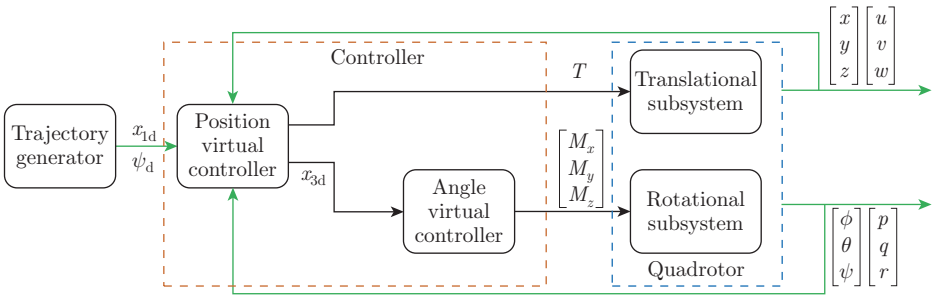


**Fig. 2** Control scheme of the aerial robot (color online)

### 3.1 Controller design for position tracking

Consider $x_{1\mathrm{d}} = [x_{\mathrm{d}}, y_{\mathrm{d}}, z_{\mathrm{d}}]^{\mathrm{T}}$ as the desired position. Then, the tracking error is denoted as $e_1 = x_{1\mathrm{d}} - x_1$.

Assume that the aerial robot has limited maneuverability considering flight safety and structural failure. The position of the aerial robot could not change rapidly. Then, a sliding surface $s_1$ for the aerial robot can be given by $s_1 = \dot{e}_1 + \sigma \mathrm{sat}_{\mathrm{L}}(e_1)$, where $s_1 = [s_x, s_y, s_z]^{\mathrm{T}}$.

$\sigma\,(\in \mathbb{R}^{3\times 3})$ is a diagonal positive definite gain matrix. The saturation vector function is defined as $\mathrm{sat_L}(e_1) = [\mathrm{sat_L}(e_{1x}), \mathrm{sat_L}(e_{1y}), \mathrm{sat_L}(e_{1z})]^{\mathrm{T}}$. The saturation function is defined as

$$\mathrm{sat_L}(e_{1i}) = \begin{cases} E_{\max}, & e_{1i} > E_{\max}, \\ e_{1i}, & |e_{1i}| < E_{\max}, \\ -E_{\max}, & e_{1i} < -E_{\max}, \end{cases}$$

where $E_{\max}$ is the designed maximum position error of the aerial robot. Differentiating the sliding surface with respect to time yields $\dot{s}_1 = \ddot{e}_1 + \sigma H \dot{e}_1$, where $H$ is a diagonal matrix with the element $h_i$ defined as $H = \mathrm{diag}(h_1, h_2, h_3)$, and

$$h_i = \begin{cases} 1, & |e_{1i}| < E_{\max}, \\ 0, & \text{otherwise}. \end{cases} \tag{6}$$

Substituting the tracking error into $\dot{s}_1 = \ddot{e}_1 + \sigma H \dot{e}_1$ yields

$$\ddot{x}_{1\mathrm{d}} - \dot{x}_2 = \dot{s}_1 - \sigma H \dot{e}_1. \tag{7}$$

By substituting the dynamic model (see Eq. (1)) into Eq. (7), the time derivative of the sliding manifold is obtained as

$$\dot{s}_1 = \ddot{x}_{1\mathrm{d}} + \sigma H \dot{e}_1 - f_1(x_2, x_4) - g_1(\dot{x}_1, T) - d_1. \tag{8}$$

With Eq. (8), the position controller is proposed, i.e.,

$$u_1 = M_{\mathrm{s}}(\ddot{x}_{1\mathrm{d}} - f_1(x_2, x_4) + \sigma H \dot{e}_1 + \widehat{F}(t) + a_1 \tanh(a_2 s_1) + a_3 \mathrm{sign}(s_1)), \tag{9}$$

where $\mathrm{sign}(s_1) = [\mathrm{sign}(s_x), \mathrm{sign}(s_y), \mathrm{sign}(s_z)]^{\mathrm{T}} \in \mathbb{R}^3$, $M_{\mathrm{s}} = \mathrm{diag}(m, m, m)$, and $\tanh(a_2 s_1) = [\tanh(a_2 s_x), \tanh(a_2 s_y), \tanh(a_2 s_z)]^{\mathrm{T}} \in \mathbb{R}^3$. $a_1$, $a_2$, and $a_3$ are diagonal positive definite gain matrices, and $\widehat{F}(t)$ is an estimation of the disturbance $d_1$ and external forces. According to Eq. (9), the derivative of the sliding surface with respect to the time $\dot{s}_1$ can be represented as

$$\dot{s}_1 = -a_1 \tanh(a_2 s_1) - a_3 \mathrm{sign}(s_1) + F(t) - \widehat{F}(t), \tag{10}$$

where $F(t)$ can be represented by an adaptive radial basis function neural network, i.e.,

$$F(t) = W^{\mathrm{T}} \Phi + \epsilon, \tag{11}$$

where $\epsilon$ denotes the approximation error which is bounded as $\|\epsilon_i\| \leqslant a_{3,\min}$. $a_{3,\min}$ is the minimum element in the matrix $a_3$. $W(\in \mathbb{R}^{m\times n})$ is the unknown parameter matrix. $m$ is the number of neurons, and $n$ is the number of outputs. In our case, $n = 3$. $\Phi\,(\in \mathbb{R}^m)$ is a vector of radial basis activation function. We take the inputs of the neural networks $\overline{x} = [x_{1\mathrm{d}}, \dot{x}_{1\mathrm{d}}, x_1, x_2]^{\mathrm{T}}$. Then, the estimation $\widehat{F}(t)$ can be denoted as

$$\widehat{F}(t) = \widehat{W}^{\mathrm{T}} \widehat{\Phi}(\widehat{c}, \widehat{\rho}), \tag{12}$$

where the estimation errors are defined as

$$\overline{\Phi} = \Phi - \widehat{\Phi}, \quad \overline{c} = c - \widehat{c}, \quad \overline{\rho} = \rho - \widehat{\rho}, \quad \overline{F} = F - \widehat{F}. \tag{13}$$

Equation (11) can be described as

$$F(t) = \overline{W}^{\mathrm{T}} \overline{\Phi}(\overline{c}, \overline{\rho}) + \overline{W}^{\mathrm{T}} \widehat{\Phi}(\widehat{c}, \widehat{\rho}) + \widehat{W}^{\mathrm{T}} \overline{\Phi}(\overline{c}, \overline{\rho}) + \widehat{W}^{\mathrm{T}} \widehat{\Phi}(\widehat{c}, \widehat{\rho}) + \epsilon. \tag{14}$$

Substituting Eq. (12) into the above equation yields

$$F(t) = \widehat{F}(t) + \overline{W}^{\mathrm{T}} \overline{\Phi}(\overline{c}, \overline{\rho}) + \overline{W}^{\mathrm{T}} \widehat{\Phi}(\widehat{c}, \widehat{\rho}) + \widehat{W}^{\mathrm{T}} \overline{\Phi}(\overline{c}, \overline{\rho}) + \epsilon. \tag{15}$$

$W^{\mathrm{T}}\Phi$ can be linearized at the points $c=\widehat{c}$ and $\rho=\widehat{\rho}$, i.e.,

$$W^{\mathrm{T}}\Phi = W^{\mathrm{T}}\widehat{\Phi}(\widehat{c},\widehat{\rho}) + W^{\mathrm{T}}\frac{\partial\widehat{\Phi}}{\partial\widehat{c}}(c-\widehat{c}) + W^{\mathrm{T}}\frac{\partial\widehat{\Phi}}{\partial\widehat{\rho}}(\rho-\widehat{\rho}). \tag{16}$$

To simplify Eq. (16), the redefined notations can be described as

$$\Delta\widehat{\Phi}_c = \frac{\partial\widehat{\Phi}}{\partial\widehat{c}}, \quad \Delta\widehat{\Phi}_\rho = \frac{\partial\widehat{\Phi}}{\partial\widehat{\rho}}. \tag{17}$$

Based on Eq. (17), Eq. (16) can be rewritten as

$$W^{\mathrm{T}}\Phi = W^{\mathrm{T}}\widehat{\Phi}(\widehat{c},\widehat{\rho}) + W^{\mathrm{T}}\Delta\widehat{\Phi}_c(c-\widehat{c}) + W^{\mathrm{T}}\Delta\widehat{\Phi}_\rho(\rho-\widehat{\rho}). \tag{18}$$

Thus, the estimation error of $\Phi$ can also be described by

$$\overline{\Phi}(\overline{c},\overline{\rho}) = \Delta\widehat{\Phi}_c\overline{c} + \Delta\widehat{\Phi}_\rho\overline{\rho}. \tag{19}$$

Substituting Eq. (19) into Eq. (15), we have the error of the estimation term $\overline{F}(t)$,

$$\overline{F}(t) = \overline{W}^{\mathrm{T}}\widehat{\Phi}(\widehat{c},\widehat{\rho}) + \widehat{W}^{\mathrm{T}}(\Delta\widehat{\Phi}_c\overline{c} + \Delta\widehat{\Phi}_\rho\overline{\rho}) + \overline{\epsilon}, \tag{20}$$

where $\overline{\epsilon} = \overline{W}^{\mathrm{T}}\overline{\Phi}(\overline{c},\overline{\rho}) + \epsilon$. $\overline{\epsilon}_{\max}$ is the maximum element in the matrix $\overline{\epsilon}$. Substituting Eq. (20) into Eq. (10) yields

$$\dot{s}_1 = -a_1\tanh(a_2 s_1) - a_3\mathrm{sign}(s_1) + \overline{W}^{\mathrm{T}}\widehat{\Phi}(\widehat{c},\widehat{\rho}) + \widehat{W}^{\mathrm{T}}(\Delta\widehat{\Phi}_c\overline{c} + \Delta\widehat{\Phi}_\rho\overline{\rho}) + \overline{\epsilon}. \tag{21}$$

Adaption laws are designed to match the convergence analysis in the following proof. The adaption laws are defined as

$$\frac{\mathrm{d}}{\mathrm{d}x}\overline{W}_i = -\frac{\mathrm{d}}{\mathrm{d}x}\widehat{W}_i = -a_4\widehat{\Phi}(\widehat{c},\widehat{\rho})\tanh(a_2 s_1), \tag{22}$$

$$\frac{\mathrm{d}}{\mathrm{d}x}\overline{c}_i = -\frac{\mathrm{d}}{\mathrm{d}x}\widehat{c}_i = -a_5\sum_{i=1}^{n}\Delta\widehat{\Phi}_c^{\mathrm{T}}\widehat{W}_i\tanh(a_2 s_1), \tag{23}$$

$$\frac{\mathrm{d}}{\mathrm{d}x}\overline{\rho}_i = -\frac{\mathrm{d}}{\mathrm{d}x}\widehat{\rho}_i = -a_6\sum_{i=1}^{n}\Delta\widehat{\Phi}_\rho^{\mathrm{T}}\widehat{W}_i\tanh(a_2 s_1), \tag{24}$$

where $\widehat{W}_i\,(\in\mathbb{R}^m)$ denotes the $i$th column of the parameter matrix $W\,(\in\mathbb{R}^{m\times n})$. Then, the parameters of the neural network are obtained by the adaptation laws (22)–(24).

**Theorem 1** *For the aerial robot system* (1), *if the controller is designed in the form of Eq.* (9) *with appropriate parameters, the adaptation laws are designed by Eqs.* (22)–(24). *If the condition* $a_{3,\min}\geqslant\overline{\epsilon}_{\max}$ *holds, then the trajectory tracking error will converge to the origin such that* $\lim\limits_{t\to\infty}e_1(t)=0$.

**Proof** Consider the Lyapunov function candidates,

$$V = \sum_{i=1}^{n}\Big(\frac{1}{a_{2i}}\ln(\cosh(a_{2i}s_{1i})) + \frac{1}{2a_4}\overline{W}_i^{\mathrm{T}}\overline{W}_i\Big) + \frac{1}{2a_5}\overline{c}^{\mathrm{T}}\overline{c} + \frac{1}{2a_6}\overline{\rho}^{\mathrm{T}}\overline{\rho}. \tag{25}$$

The time derivative of $V$ is

$$\dot{V} = \sum_{i=1}^{n}\Big(\tanh(a_{2i}s_{1i})\dot{s}_{1i} + \frac{1}{a_4}\overline{W}_i^{\mathrm{T}}\dot{\overline{W}}_i\Big) + \frac{1}{a_5}\overline{c}^{\mathrm{T}}\dot{\overline{c}} + \frac{1}{a_6}\overline{\rho}^{\mathrm{T}}\dot{\overline{\rho}}. \tag{26}$$

Substituting Eq. (21) into Eq. (26) yields

$$\dot{V}=\sum_{i=1}^{n}\Big(-a_{1i}\tanh(a_{2i}s_{1i})^2-a_{3i}\tanh(a_{2i}s_{1i})\mathrm{sign}(s_{1i})+\tanh(a_{2i}s_{1i})\overline{W}_i^{\mathrm{T}}\widehat{\Phi}(\widehat{c},\widehat{\rho})$$
$$+\tanh(a_{2i}s_{1i})\widehat{W}^{\mathrm{T}}(\Delta\widehat{\Phi}_c\overline{c}+\Delta\widehat{\Phi}_\rho\overline{\rho})+\tanh(a_{2i}s_{1i})\overline{\epsilon}_i+\frac{1}{a_4}\overline{W}_i^{\mathrm{T}}\dot{\overline{W}}_i\Big)+\frac{1}{a_5}\overline{c}^{\mathrm{T}}\dot{\overline{c}}+\frac{1}{a_6}\overline{\rho}^{\mathrm{T}}\dot{\overline{\rho}}. \quad (27)$$

The time derivative of $V$ can be rewritten as

$$\dot{V}=\sum_{i=1}^{n}(-a_{1i}\tanh(a_{2i}s_{1i})^2-a_{3i}\tanh(a_{2i}s_{1i})\mathrm{sign}(s_{1i})+\tanh(a_{2i}s_{1i})\overline{\epsilon}_i)$$
$$+\overline{W}_i^{\mathrm{T}}\Big(\sum_{i=1}^{n}\Big(\widehat{\Phi}(\widehat{c},\widehat{\rho})\tanh(a_2s_1)+\frac{1}{a_4}\dot{\overline{W}}_i\Big)\Big)+\overline{c}^{\mathrm{T}}\Big(\sum_{i=1}^{n}(\tanh(a_{2i}s_{1i})\Delta\widehat{\Phi}_c^{\mathrm{T}}\widehat{W}_i)+\frac{1}{a_5}\dot{\overline{c}}\Big)$$
$$+\overline{\rho}^{\mathrm{T}}\Big(\sum_{i=1}^{n}(\tanh(a_{2i}s_{1i})\Delta\widehat{\Phi}_\rho^{\mathrm{T}}\widehat{W}_i)+\frac{1}{a_6}\dot{\overline{\rho}}\Big). \quad (28)$$

Substituting the adaptation laws (22)–(24) into Eq. (28), we obtain

$$\dot{V}=\sum_{i=1}^{n}(-a_{1i}\tanh(a_{2i}s_{1i})^2-a_{3i}\tanh(a_{2i}s_{1i})\mathrm{sign}(s_{1i})+\tanh(a_{2i}s_{1i})\overline{\epsilon}_i). \quad (29)$$

Define $\alpha(s_{1i})=-a_{3i}\tanh(a_{2i}s_{1i})\mathrm{sign}(s_{1i})+\tanh(a_{2i}s_{1i})\overline{\epsilon}_i$. It can also be described as

$$\alpha(s_{1i})=-a_{3i}|\tanh(a_{2i}s_{1i})|+\tanh(a_{2i}s_{1i})\overline{\epsilon}_i. \quad (30)$$

Thus, if we take $a_{3,\min}\geqslant\overline{\epsilon}_{\max}$, we have $\alpha(s_{1i})\leqslant 0$. Then, $\dot{V}$ can be rewritten as

$$\dot{V}=\sum_{i=1}^{n}(-a_{1i}\tanh(a_{2i}s_{1i})^2+\alpha(s_{1i}))\leqslant 0.$$

This result guarantees the boundness of the trajectory tracking error of the aerial robot. The result also guarantees that the signal $e_1(t)$ of the position controller converges to zero.

**Remark 1** The theoretical proof of Theorem 1 is derived from the literature[29]. However, in contrast to the literature[29], a new sliding surface is proposed that takes into account the limited maneuverability and flight safety of the aerial robot. A new Lyapunov function is designed for convergence analysis. We also prove that the control law can be obtained under the novel defined sliding surface.

### 3.2 Controller design for attitude stabilization

In this section, an MFISMC is proposed to stabilize attitude angles. Designing a high-performance controller is a challenge due to parameter uncertainties and external noises. In the conventional SMC, it is difficult to obtain the variable function $f_2(x_4)$ due to the variation of the model parameters in order to meet some specific task requirements. Thus, an MFISMC is proposed to avoid the above problems. The proposed controller utilizes a linear mathematical model as part of the controller to stabilize both the Euler angles and angular rates of the aerial robot.

The desired attitude angles are denoted as $x_{3\mathrm{d}}=[\phi_{\mathrm{d}},\theta_{\mathrm{d}},\psi_{\mathrm{d}}]^{\mathrm{T}}$. Combine attitude angles with angular velocities to generate a new state vector as $x_l=[\phi,\theta,\psi,p,q,r]^{\mathrm{T}}$. The approximate linear dynamics in each iteration can be described as $\dot{x}_4=F_1x_l+Gu$, where $u$ denotes the additional moment acting on the aerial robot. $F_1\,(\in\mathbb{R}^{3\times 6})$ and $G\,(\in\mathbb{R}^{3\times 3})$ are constant matrices whose elements are required to be identified. It should be noted that the matrix $G$ satisfies $\mathrm{rank}\,G=3$.

The model-free controller is defined as $u = G^{-1}(\ddot{x}_{3\mathrm{d}} - F_1 x_l - u_c)$, where $\ddot{x}_{3\mathrm{d}}$ is the second-order derivative of the desired states. $u_c$ is a feedback controller for tracking the desired states. In our case, we design the auxiliary feedback controller $u_c$ as

$$u_c = -\tau H_1 \dot{e}_3 - k_1 \tanh(k_2 s_2) - k_3 \mathrm{sign}(s_2), \tag{31}$$

where $e_4 (= \dot{x}_{3\mathrm{d}} - x_4)$ is the derivative of the trajectory tracking error. $\tau (\in \mathbb{R}^{3\times 3})$ is a diagonal positive definite gain matrix. $k_1$, $k_2$, and $k_3$ are diagonal positive definite gain matrices. $s_2 = [s_\phi, s_\theta, s_\psi]^{\mathrm{T}}$ is the designed sliding surface described as $s_2 = \tau \mathrm{sat}_{\mathrm{L}}(e_3) + \dot{e}_3$.

The derivative of the sliding surface $s_2$ is described as

$$\dot{s}_2 = \tau H_1 \dot{e}_3 + \ddot{e}_3, \tag{32}$$

where $H_1$ is defined similarly to the function $H$. By using Eq. (31), we have

$$u = G^{-1}(\ddot{x}_{3\mathrm{d}} - F_1 x_l + \tau H_1(\dot{x}_{3\mathrm{d}} - x_4) + k_1 \tanh(k_2 s_2) + k_3 \mathrm{sign}(s_2)). \tag{33}$$

Replacing the definition of $e_3$ into Eq. (32) yields

$$\ddot{x}_{3\mathrm{d}} - \dot{x}_4 = \dot{s}_2 - \tau H_1 \dot{e}_3. \tag{34}$$

By using Eq. (34), we obtain

$$\dot{s}_2 = -k_1 \tanh(k_2 s_2) - k_3 \mathrm{sign}(s_2) + e_{\mathrm{est}}, \tag{35}$$

where $e_{\mathrm{est}} (= F_1 x_l - F_1(x_4) - d_2)$ denotes the estimation error.

**Theorem 2** *For the aerial robot system* (3)*, if the attitude controller is designed in the form of Eq.* (33) *with appropriate parameters, the attitude tracking error will converge to the origin even in the presence of disturbance such that* $\lim\limits_{t\to\infty} e_3(t) = 0$.

**Proof** Consider the Lyapunov function candidates,

$$V_i = \frac{1}{k_2} \ln(\cosh(k_2 s_{2i})), \quad i = 1, 2, 3, \tag{36}$$

where $s_{2i}$ denotes the designed sliding surface. By using Eq. (35), the time derivative of $V_i$ is

$$\dot{V}_i = \tanh(k_2 s_{2i})(-k_1 \tanh(k_2 s_{2i}) - k_3 \mathrm{sign}(s_{2i}) + e_{\mathrm{est}}). \tag{37}$$

The time derivative of $V_i$ can be rewritten as

$$\dot{V}_i = -k_1 \tanh(k_2 s_{2i})^2 - k_3 \mathrm{sign}(s_{2i}) \tanh(k_2 s_{2i}) + e_{\mathrm{est}} \tanh(k_2 s_{2i}). \tag{38}$$

Thus, if we take $e_{\mathrm{est}} = 0$, the Lyapunov equation $\dot{V}_i \leqslant 0$ can guarantee that the attitude angles and angular velocities of the aerial robot will converge to zero along the designed sliding surface. In other words, the attitude angles and angular velocities of the aerial robot will converge to zero under the condition that the estimated nonlinear system dynamics are sufficiently accurate.

To identify effective strategies for parameter optimization, we use the reinforcement learning algorithm state-action-reward-state-action (SARSA)[33–34]. The space of states is composed of the mean absolute error (MAE) of the system's states. The specific form of the states of the roll channel is described as $\Omega_\phi = (E_\phi, E_p)$. The states of the pitch channel $\Omega_\theta$ and the states of the yaw channel $\Omega_\psi$ in the reinforcement learning algorithm are defined similarly, i.e.,

$$E_i = \frac{1}{T_1} \int_0^{T_1} |e_i| \mathrm{d}t, \quad i = \phi, \theta, \psi, p, q, r,$$

where $T_1$ denotes the iteration duration of the experiment. $e_i \, (i = \phi, \theta, \psi, p, q, r)$ denotes the state tracking error relative to original parameters in the attitude controller (33).

The action space is a transformation of the controller parameters $k_1$ and $\tau$ described as $(\Delta k_1, \Delta\tau)$. Eventually, the rewards for the reinforcement learning algorithm in three control channels are denoted as

$$R_l = \begin{cases} 2, & \overline{E}_l + \overline{E}_j < 0.5(E_l + E_j), \\ 1, & 0.5(E_l + E_j) < \overline{E}_l + \overline{E}_j < E_l + E_j, \\ -1, & E_l + E_j < \overline{E}_l + \overline{E}_j < 1.5(E_l + E_j), \\ -2, & 1.5(E_l + E_j) < \overline{E}_l + \overline{E}_j, \end{cases}$$

where $l = \phi, \theta, \psi$, and $j = p, q, r$. The definition of $\overline{E}_i$ $(i = \phi, \theta, \psi, p, q, r)$ is given by

$$\overline{E}_i = \frac{1}{T_1} \int_0^{T_1} |\overline{e}_i| \mathrm{d}t, \quad i = \phi, \theta, \psi, p, q, r,$$

where $\overline{e}_i$ $(i = \phi, \theta, \psi, p, q, r)$ is the tracking error utilizing the parameters that are optimized by the reinforcement learning algorithm. With the design of states, action spaces, and rewards, we can obtain the optimal parameters for the controller in Eq. (33).

From the results of Theorem 2, the algorithm for implementing the MFISMC for the attitude angles of the aerial robot is shown in Algorithm 1. We can therefore conclude that such an algorithm can stabilize attitude angles.

Algorithm 1 Model-free iterative sliding mode algorithm

Input $\tau$, $k_1$, $k_2$, and $k_3$.

(I) Select an initial control input $u^{(0)}$, system dynamics $F^{(0)}(t)$ and $G^{(0)}(t)$, and a computation accuracy $\epsilon > 0$. Let $i = 0$.

(II) Obtain $u^{(i+1)}(x)$.

$$u^{(i+1)}(x) = {G^{(i)}}^{-1}(\ddot{x}_{3\mathrm{d}} - F^{(i)} x_l^{(i+1)} + \tau H_1(\dot{x}_{3\mathrm{d}} - x_4^{(i+1)}) + k_1 \tanh(k_2 s_2^{(i+1)}) + k_3 \mathrm{sign}(s_2^{(i+1)})).$$

(III) Apply $u^{(i+1)}(x)$ as the control input to generate new data $x^{(i+1)}$ and $\dot{x}^{(i+1)}$, and estimate approximate system dynamics $F^{(i+1)}(t)$ and $G^{(i+1)}(t)$ by the least square method.

$$[F^{(i+1)} G^{(i+1)}] = [(Z^{(i+1)})^{\mathrm{T}} Z^{(i+1)}]^{-1} (Z^{(i+1)})^{\mathrm{T}} \dot{X}^{(i+1)},$$
$$Z^{(i)} \triangleq [z_{[1]}^{(i+1)}, \cdots, z_{[N]}^{(i+1)}]^{\mathrm{T}}, \quad \dot{X}^{(i)} \triangleq [\dot{x}_{[1]}^{(i+1)}, \cdots, \dot{x}_{[N]}^{(i+1)}]^{\mathrm{T}}$$

with $z_{[l]}^{(i+1)} = [x_{[l]}^{(i+1)} u_{[l]}^{(i+1)}]$ $(l = 1, 2, \cdots, N)$.

(IV) If the error satisfies $\|F^{(i+1)}(x) - F^{(i)}(x)\| \leqslant \epsilon$, then use the SARSA algorithm to obtain a group of better controller parameters. Otherwise, set $i := i + 1$ to obtain appropriate control inputs until the error satisfies $\|F^{(i+1)}(x) - F^{(i)}(x)\| \leqslant \epsilon$.

## 4 Simulation results

In this section, the numerical simulations are provided to illustrate the efficiency and superiority of the proposed control strategy, including the performance of position tracing and attitude stabilization. Besides, the comparisons among the strategy developed in this paper, the traditional PID, backstepping (BS), and SMC are carried out to show the improvement. The main parameters of the aerial robot used in the simulations are given in Table 1.

The simulations are performed by MATLAB2020b/Simulink, which is equipped with a computer composed of an Intel Core i7-11700 2.5 GHz CPU with 16 GB RAM and a 100 GB solid-state disk drive.

The desired trajectory used is a circle, which is governed by

$$x_{\mathrm{d}} = \frac{1}{2}\cos\left(\frac{\pi t}{20}\right), \quad y_{\mathrm{d}} = \frac{1}{2}\sin\left(\frac{\pi t}{20}\right), \quad z_{\mathrm{d}} = 3 - 2\cos\left(\frac{\pi t}{20}\right), \quad \psi_{\mathrm{d}} = \sin\left(\frac{\pi t}{20}\right).$$

**Table 1** Main parameters of the aerial robot

| Parameter | Value | Unit |
|---|---|---|
| $b$ | $1.105 \times 10^{-5}$ | – |
| $k$ | $1.489 \times 10^{-7}$ | – |
| $m$ | 1.500 | kg |
| $l$ | 0.225 | m |
| $J_x$ | $1.745 \times 10^{-2}$ | $\mathrm{kg \cdot m^2}$ |
| $J_y$ | $1.745 \times 10^{-2}$ | $\mathrm{kg \cdot m^2}$ |
| $J_z$ | $3.175 \times 10^{-2}$ | $\mathrm{kg \cdot m^2}$ |

The initial conditions of the aerial robot are $x_1(0) = [0.45, 0.05, 0.45]^{\mathrm{T}}\,\mathrm{m}$, $x_2(0) = [0, 0, 0]^{\mathrm{T}}\,\mathrm{m/s^2}$, $x_3(0) = [0.1, 0.1, 0.1]^{\mathrm{T}}\,\mathrm{rad}$, and $x_4(0) = [0, 0, 0]^{\mathrm{T}}\,\mathrm{rad/s}$. Besides, according to Theorem 1, the gains are set as $\sigma = [2.8, 1.8, 3]^{\mathrm{T}}$, $a_1 = \mathrm{diag}(0.25, 0.25, 20.5)$, $a_2 = \mathrm{diag}(10, 10, 1)$, $a_3 = \mathrm{diag}(0.01, 0.01, 0.01)$, $a_4 = 0.005$, $a_5 = 0.009$, and $a_6 = 0.2$. Based on Eqs. (22)–(24), the random values into the set $[-1, 1]$ are selected as the initial parameters of the matrices $\Phi$, $\rho$, and $c$. As for the attitude controller, the main parameters are selected as follows: $\tau = \mathrm{diag}(5, 5, 5)$, $k_1 = \mathrm{diag}(5, 5, 5)$, $k_2 = \mathrm{diag}(10, 9, 10)$, and $k_3 = \mathrm{diag}(1, 1, 1)$. According to Algorithm 1, the initial parameters in the matrices $F^{(i)}$ and $G^{(i)}$ are randomly selected into the set [0, 1]. With iterations increasing, the parameters in the matrices $F^{(i)}$ and $G^{(i)}$ are obtained from flight data. The unmodeled dynamics $d_1$ is defined as

$$d_{1x} = 0.01\sin(0.5t + 50), \quad d_{1y} = 0.01\sin(0.5t + 50), \quad d_{1z} = 0.03\sin(0.5t + 50).$$

To reduce the computational load, the state spaces of the attitude controller are discretized. The thresholds to discretize the space of states in the attitude controller are described as

$$\begin{cases} s_\phi = [0, 0.06, 0.08, 0.1, 0.12, 0.15, 0.18], \\ s_\theta = [0.1, 0.15, 0.2, 0.25, 0.3, 0.35, 0.4, 0.45, 0.5], \\ s_\psi = [0.05, 0.1, 0.15, 0.2, 0.25, 0.3, 0.35, 0.4, 0.45], \\ s_p = [0.2, 0.25, 0.3, 0.35, 0.4, 0.45, 0.5, 0.55], \\ s_q = [0.2, 0.25, 0.3, 0.35, 0.4, 0.45, 0.5, 0.55, 0.6, 0.65, 0.7], \\ s_r = [0.1, 0.15, 0.2, 0.22, 0.28, 0.3, 0.32, 0.35, 0.37, 0.4, 0.42, 0.45, 0.5]. \end{cases}$$

The action space applied in SARSA is a combination of two kinds of actions. These two kinds of actions ($\Delta a_k$ and $\Delta a_\tau$) can be described as

$$\Delta a_k = [-3, -1, 0, 1, 3], \quad \Delta a_\tau = [-0.2, 0, 0.2].$$

There are 15 elements in the action space, $\Delta a_k$ and $\Delta a_\tau$. The simulation results of trajectory tracking errors are shown in Fig. 3. It can be seen that the proposed strategy enables the aerial robot to follow the reference trajectory. Figure 3 shows the translational tracking errors. From Fig. 3, the translational tracking is achieved, and the steady-state tracking error using the proposed controller is the smallest compared with the other controllers. The proposed strategy in the translational tracking has better performance in terms of overshoot and settling time compared with the other traditional controllers. Due to the influence of adaptive neural networks, the proposed algorithm is still effective in tracking the desired trajectory without prior knowledge of physical parameters compared with the conventional SMC. Also, due to the influence of the constrained sliding surface, the proposed algorithm has a slightly smaller tracking error than the conventional SMC in the $x$-axis direction. Compared with the other conventional algorithms, the proposed algorithm generates smaller tracking errors for trajectory tracking tasks and can achieve faster convergence.

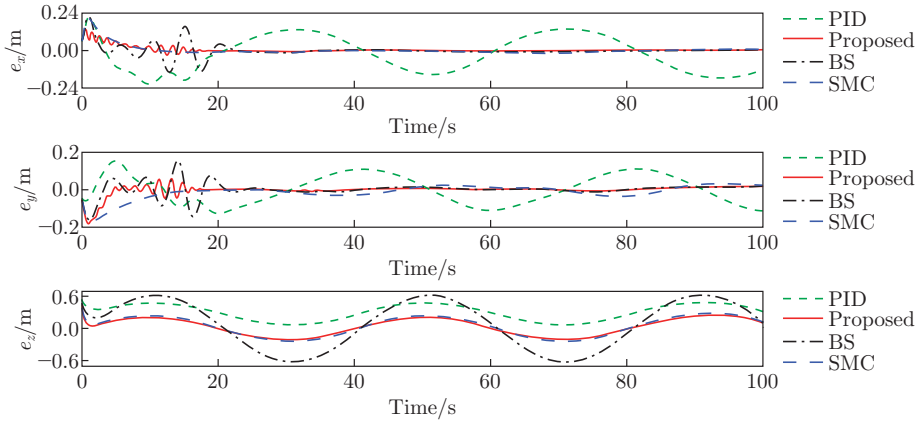


**Fig. 3** Comparisons of position trajectory tracking errors (color online)

The root mean square (RMS) values of the tracking errors are presented in Table 2. We can observe that, the proposed control scheme presents the best tracking accuracy, which confirms the advantages of the proposed control scheme. Specifically, compared with the other conventional algorithms, the proposed controller maximally reduces the RMS tracking error by 29.27% in the forward position tracking task, 65% in the lateral position tracking task, and 63.9% in the altitude tracking task.

**Table 2** RMS values of position errors $e_x$, $e_y$, and $e_z$ in the time interval $0\,\mathrm{s} \leqslant t \leqslant 100\,\mathrm{s}$

| Error | Proposed | PID | BS | SMC |
|---|---|---|---|---|
| $e_x$/m | 0.024 91 | 0.035 22 | 0.031 05 | 0.030 95 |
| $e_y$/m | 0.017 59 | 0.050 27 | 0.034 02 | 0.028 19 |
| $e_z$/m | 0.066 90 | 0.146 20 | 0.185 30 | 0.074 10 |

As for the attitude controller, Fig. 4 shows the variation of the parameters in the estimated matrices $F^{(i)}$ and $G^{(i)}$ during the iterative experiments. As observed, the parameters in the matrices $F^{(i)}$ and $G^{(i)}$ tend to be stable after the fourth iteration. It demonstrates the validity of the proposed MFISMC.

Furthermore, to demonstrate the superiority of the proposed MFISMC, the comparisons with a PID controller in the inner loop are conducted based on the previous settings. The angle controller and angular velocity controller using PID are $u_4 = k_{p_4} e_4 + k_{d_4} \frac{\mathrm{d}e_4}{\mathrm{d}t} + k_{i_4} \int_0^t e_4(\tau)\mathrm{d}\tau$. The fine-tuning parameters of the PID controller are listed as follows: $k_{p_3} = \mathrm{diag}(7.5, 10, 3.8)$, $k_{p_4} = \mathrm{diag}(0.21, 0.14, 0.82)$, $k_{d_4} = \mathrm{diag}(0.01, 0, 0)$, and $k_{i_4} = \mathrm{diag}(0.002, 0.002, 0.02)$.

Figure 5 shows the Euler angle tracking performance using different controllers, and it can be seen that the rotational tracking is achieved by the proposed MFISMC. Both the PID controller and the proposed controller achieve null steady-state errors in the $\phi$ and $\theta$ tracking tasks. However, there exists a steady error caused by using the PID controller in the $\psi$ tracking task.

Figure 6 shows the rotational tracking errors induced by utilizing different controllers. The proposed control strategy can achieve null steady-state errors in the $\psi$ tracking task. The PID controller needs almost 21 s to achieve null steady-state errors in the $\phi$ and $\theta$ tracking tasks. The proposed controller spends almost 1 s achieving null steady-state errors. It is demonstrated that the proposed controller achieves faster convergence and zero steady-state errors compared with the PID controller. In this simulation example, the reference trajectory of the yaw angle $\psi$ is a sinusoidal curve. Because the derivative of the reference trajectory is present in the

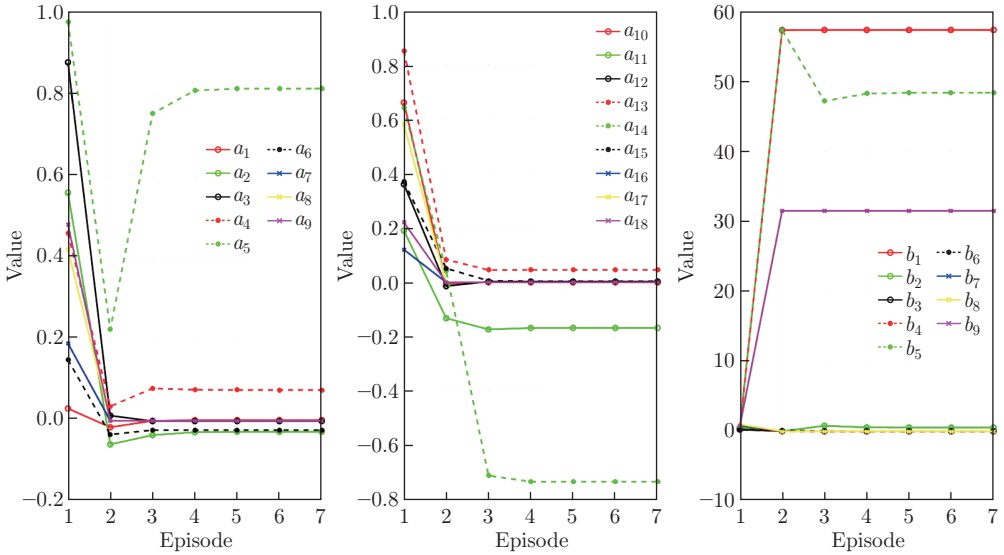


**Fig. 4** Variations of parameters in the matrices $F^{(i)}$ and $G^{(i)}$ (color online)

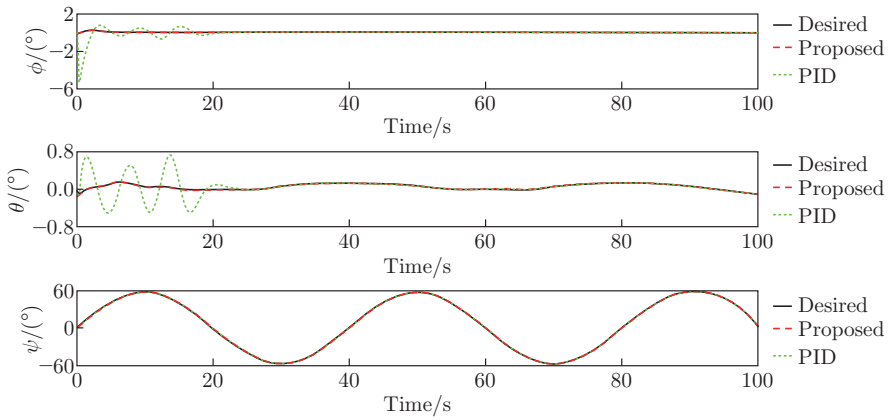


**Fig. 5** Euler angle trajectories of different controllers and their corresponding desired signals (color online)

designed controller, an approximate step response will be generated in the control input at the start of the tracking task. Finally, the control input causes an overshoot at the beginning of the tracking task. This phenomenon can be reduced by using the discrete differential tracker which will be implemented in future works.

To verify the noise immunity and robustness of the proposed controller, we consider a Gaussian noise with zero mean and variance of $2.5\times10^{-5}$ rad in the measurement process. The Euler angle tracking errors when taking into account the Gaussian noise are exhibited in Fig. 7. It can be seen that both the PID controller and the proposed attitude controller achieve tracking control of the desired trajectory despite the presence of the Gaussian noise. However, at the beginning of the $\phi$ tracking task using the PID controller, there exists an overshoot of up to 15°. The tracking error generated by the proposed algorithm does not show any overshoot. At the beginning of the $\theta$ tracking task, the maximum tracking error using the PID controller is up to 1.3°. The maximum tracking error produced by the proposed algorithm is 0.8°, which demonstrates that the proposed algorithm has good noise immunity.

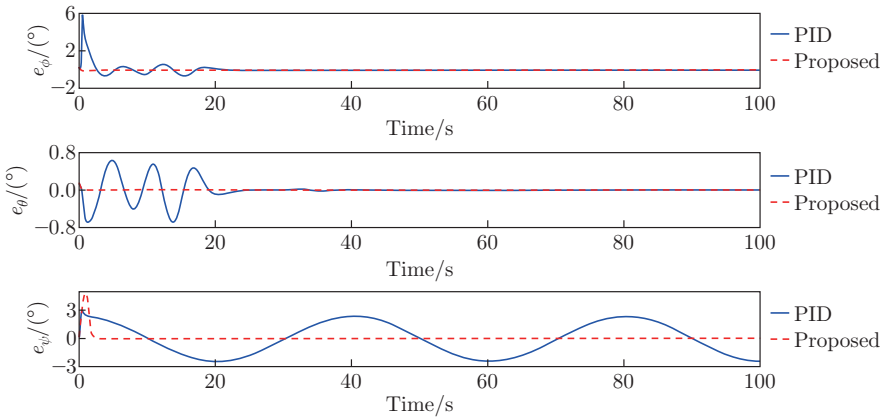


**Fig. 6** Euler angle tracking errors for different controllers (color online)

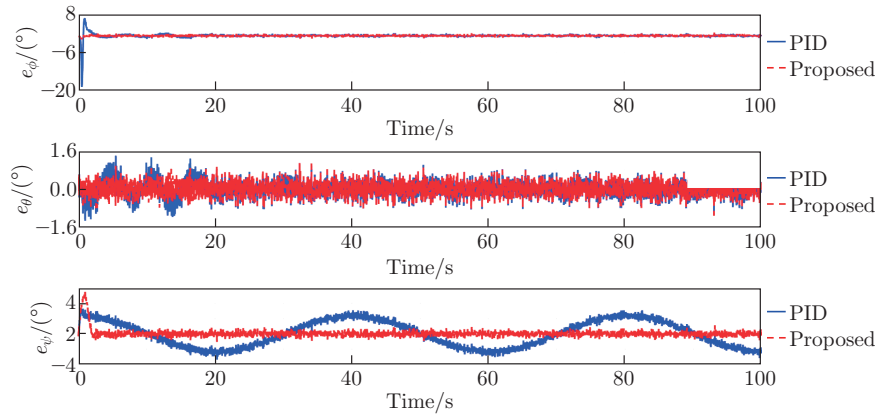


**Fig. 7** Euler angle tracking errors for different controllers with the Gaussian noise (color online)

Finally, to obtain a group of better parameters, the SARSA algorithm is used to improve the control performance of the attitude controller. The control performances of the improved attitude controller are shown in Fig. 8. Due to the variations of control parameters ($\Delta k$ and $\Delta\tau$), the sum of the MAEs of the Euler angle and angular velocity can be significantly reduced using the SMC-based SARSA algorithm. Compared with the original control performance, the sum of the MAEs in the angle $\phi$ and angular velocity $p$ is significantly reduced by 92.97% using the SARSA algorithm. Similarly, the sum of the MAEs for the angle $\theta$ and angular velocity $q$ is decreased by 78.18%. The sum of the MAEs for the angle $\psi$ and angular velocity $r$ is reduced by 0.37%.

## 5 Discussion

In this paper, we propose a hierarchical control strategy consisting of an adaptive SMC and an MFISMC. One of the contributions of this study is to devise a position controller that combines the sliding surface consisting of constrained states with RBFNNs. Another contribution is to introduce a linear approximate model as a part of the attitude controller to stabilize both the Euler angles and angular rates of the aerial robot. However, there exists a limitation of the proposed position controller that not all trajectories can be controlled to the desired position. Since the RBFNN is used as a part of the proposed position controller, continuous excitation must be maintained. Considering the limits of maneuverability and flight safety consideration,

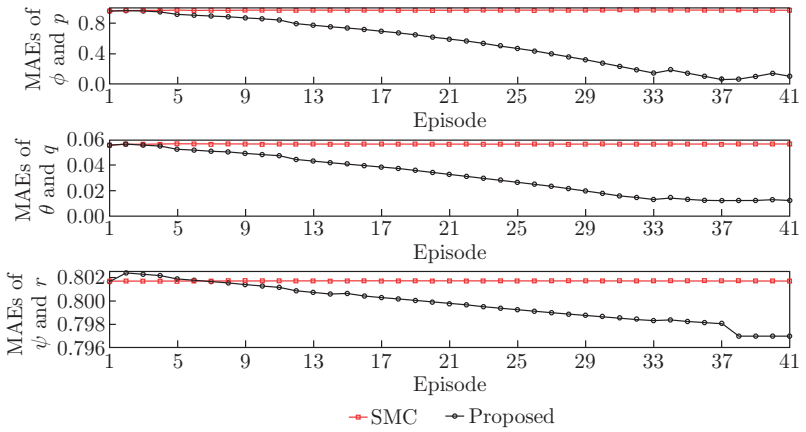


**Fig. 8** Comparisons of MAEs of the proposed and SMC algorithms (color online)

the desired trajectories and their first-order derivatives should be continuously changing within a certain amplitude to maintain continuous excitation. Many trajectories meet this condition, such as the figure-of-eight flight trajectory, the circular trajectory mentioned in this paper, and the spiral upward trajectory. Although the proposed position controller is a preliminary theoretical design, the simulations indicate that it is effective in tracking the desired trajectory and generates fewer tracking errors.

As for the applications of the proposed attitude controller, the quadrotor system is realized by the PX4 autopilot, and system states can be measured by onboard sensors. Based on these signals and the identified physical parameters, we can apply the control scheme proposed in this paper to real-world experiments. Due to the performance of the sensor and external noises on a low-cost quadcopter, the angular velocities change rapidly. In addition, unlike the design of a sliding surface with the angular velocities in numerical simulations, the derivatives of the roll angle and pitch angle errors should be used to formulate the attitude controller in real-world experiments. With the designed sliding surface, the control scheme proposed in this paper can be employed in complex curve-tracking tasks such as drone racing and thermal soaring.

## 6 Conclusions

In this work, a constructive hierarchical controller consisting of an adaptive sliding mode controller constrained by position tracking errors and an MFISMC is developed to solve the position and attitude tracking problem of a quadrotor aerial robot. For the position tracking problem of a quadrotor aerial robot, a constrained adaptive RBFNN controller for translational motion is proposed, and it achieves smooth performance in the position tracking task. Compared with the conventional SMC, by using the adaptive RBFNN, the proposed position controller is effective in tracking a desired trajectory, even though all the dynamic parameters are unknown. Compared with the other conventional control algorithms, the proposed position controller maximally reduces the RMS tracking error by 29.27%, 65%, and 63.9% in three control channels.

On the other hand, an MFISMC guarantees that the aerial robot follows the desired attitude trajectory. An approximate linear system is introduced to formulate the attitude controller. Then, the SARSA algorithm is used to improve the performance of the attitude controller. The robustness and effectiveness of the proposed control scheme are also demonstrated. The proposed controller achieves faster convergence and zero steady-state errors compared with the PID controller. Compared with the original control performance, the sum of the MAE in the

Euler angle and angular velocity is reduced by 92.97%, 78.18%, and 0.37% in three control channels.

In the near future, the proposed controller will be implemented in a real quadrotor system. The quadrotor system is being built, which includes a BMI088 sensor to obtain the attitude angles and a global positioning system (GPS) module to obtain the position of the quadrotor. Based on these signals and the identified physical parameters, the control scheme proposed in this paper can be employed in complex curve-tracking tasks such as drone racing and thermal soaring.

**Conflict of interest** The authors declare no conflict of interest.

## References


[1] LYU, J. B., ZENG, Y., ZHANG, R., and LIM, T. J. Placement optimization of UAV-mounted mobile base stations. *IEEE Communications Letters*, **21**(3), 604–607 (2016)

[2] STAMPA, M., SUTORMA, A., JAHN, U., THIEM, J., WOLFF, C., and ROEHRIG, C. Maturity levels of public safety applications using unmanned aerial systems: a review. *Journal of Intelligent & Robotic Systems*, **103**(1), 1–13 (2021)

[3] HERISSE, B., HAMEL, T., MAHONY, R., and RUSSOTTO, F. Landing a VTOL unmanned aerial vehicle on a moving platform using optical flow. *IEEE Transactions on Robotics*, **28**(1), 77–89 (2012)

[4] SUNG, I. and NIELSEN, P. Zoning a service area of unmanned aerial vehicles for package delivery services. *Journal of Intelligent & Robotic Systems*, **97**(3), 719–731 (2020)

[5] UMEMOTO, K., ENDO, T., and MATSUNO, F. Dynamic cooperative transportation control using friction forces of $n$ multi-rotor unmanned aerial vehicles. *Journal of Intelligent & Robotic Systems*, **100**(3), 1085–1095 (2020)

[6] AN, N. B., WANG, Q. S., ZHAO, X. C., and WANG, Q. Y. Differential flatness-based distributed control of underactuated robot swarms. *Applied Mathematics and Mechanics* (*English Edition*), **44**(10), 1777–1790 (2023) https://doi.org/10.1007/s10483-023-3040-8

[7] LIU, S. K., MICHAEL, W., KARTIK, M., SUN, K., SUBHRAJIT, B., CAMILLO, J. T., and VIJAY, K. Planning dynamically feasible trajectories for quadrotors using safe flight corridors in 3-D complex environments. *IEEE Robotics and Automation Letters*, **2**(3), 1688–1695 (2017)

[8] ZHOU, B. Y., PAN, J., GAO, F., and SHEN, S. J. RAPTOR: robust and perception-aware trajectory replanning for quadrotor fast flight. *IEEE Transactions on Robotics*, **37**(6), 1992–2009 (2021)

[9] SUN, G. J., ZHOU, M. Q., and JIANG, X. Q. Non-cooperative spacecraft proximity control considering target behavior uncertainty. *Astrodynamics*, **6**, 399–411 (2022)

[10] CHEN, T., SHAN, J. J., WEN, H., and XU, S. D. Review of attitude consensus of multiple spacecraft. *Astrodynamics*, **6**, 329–356 (2022)

[11] MARSHALL, J. A., SUN, W., and ANDREA, L. A. A survey of guidance, navigation, and control systems for autonomous multi-rotor small unmanned aerial systems. *Annual Reviews in Control*, **52**(1), 390–427 (2021)

[12] GUERRERO-SANCHEZ, M. E., LOZANO, R., CASTILLO, P., HERNANDEZ-GONZALE, Z. O., GARCIA-BELTRAN, C. D., and VALENCIA-PALOMO, G. Nonlinear control strategies for a UAV carrying a load with swing attenuation. *Applied Mathematical Modelling*, **91**(1), 709–722 (2021)

[13] WANG, B. F., WANG, Q., ZHOU, Q., and LIU, Y. L. Active control of flow past an elliptic cylinder using an artificial neural network trained by deep reinforcement learning. *Applied Mathematics and Mechanics* (*English Edition*), **43**(12), 1921–1934 (2022) https://doi.org/10.1007/s10483-022-2940-9

[14] LIU, S., ZHANG, R., WANG, Q. Y., and HE, X. Y. Sliding mode synchronization between uncertain Watts-Strogatz small-world spatiotemporal networks. *Applied Mathematics and Mechanics* (*English Edition*), **41**(12), 1833–1846 (2020) https://doi.org/10.1007/s10483-020-2686-6

[15] GE, X. S., YAO, Q. J., and CHEN, L. Q. Control strategy of optimal deployment for spacecraft solar array system with initial state uncertainty. *Applied Mathematics and Mechanics* (*English Edition*), **39**(10), 1437–1452 (2018) https://doi.org/10.1007/s10483-018-2378-8

[16] NOORMOHAMMADI, A. A., ESRAFILIAN, O., AHANGAR, M., and HAMID, T. System identification and $H_\infty$ based control of quadrotor attitude. *Mechanical Systems and Signal Processing*, **135**, 171–183 (2020)

[17] MIRANDA, C. R. and AGUILAR, L. T. Robust PID control of quadrotors with power reduction analysis. *ISA Transactions*, **98**, 47–62 (2020)

[18] SUN, J. T., WANG, Q. G., and GAO, H. Q. Delay-dependent robust stability and $H_\infty$ analysis of stochastic systems with time-varying delay. *Applied Mathematics and Mechanics* (*English Edition*), **31**(2), 255–262 (2010) https://doi.org/10.1007/s10483-010-0213-6

[19] REKABI, F., SHIRAZI, F. A., SADIGH, M. J., and SAADAT, M. Nonlinear $H_\infty$ measurement feedback control algorithm for quadrotor position tracking. *Journal of the Franklin Institute*, **357**(11), 6777–6804 (2020)

[20] BOUABDALLAH, S., MURRIERI, P., and SIEGWART, R. Towards autonomous indoor micro VTOL. *Autonomous Robots*, **18**(2), 171–183 (2005)

[21] ELKHATEM, A. and ENGIN, S. N. Robust LQR and LQR-PI control strategies based on adaptive weighting matrix selection for a UAV position and attitude tracking control. *Alexandria Engineering Journal*, **61**(8), 6275–6292 (2022)

[22] VAZQUEZ-NICOLAS, J. M., ZAMORA, E., GONZALEZ-HERNANDEZ, I., LOZANO, R., and SOSSA, H. PD+SMC quadrotor control for altitude and crack recognition using deep learning. *International Journal of Control, Automation and Systems*, **18**(4), 834–844 (2020)

[23] MARTINS, L., CARDEIRA, C., and OLIVEIRA, P. Feedback linearization with zero dynamics stabilization for quadrotor control. *Journal of Intelligent & Robotic Systems*, **101**, 1–17 (2021)

[24] BENALLEGUE, A., MOKHTARI, A., and FRIFMAN, L. M. High-order sliding-mode observer for a quadrotor UAV. *International Journal of Robust and Nonlinear Control*, **18**(4), 312–322 (2008)

[25] WANG, H. P., YE, X. F., TIAN, Y., ZHENG, G., and CHRISTOV, N. Model-free based terminal SMC of quadrotor attitude and position. *IEEE Transactions on Aerospace and Electronic Systems*, **52**(5), 2519–2528 (2012)

[26] AMINURRASHID, N., BASRI, M. A. M., and MOHAMED, Z. Position and attitude tracking of MAV quadrotor using SMC-based adaptive PID controller. *Drones*, **6**(9), 1–18 (2022)

[27] LABBADI, M. and MOUSSAOUI, H. E. An improved adaptive fractional-order fast integral terminal sliding mode control for distributed quadrotor. *Mathematics and Computers in Simulation*, **188**, 120–134 (2021)

[28] HUA, C. C., CHEN, J. N., and GUAN, X. P. Adaptive prescribed performance control of QUAVs with unknown time-varying payload and wind gust disturbance. *Journal of the Franklin Institute*, **355**(14), 6323–6338 (2018)

[29] LOPEZ, S. I., ROSSOMANDO, F., PEREZ, A. R., SORIA, C., and MORENO, J. Adaptive trajectory tracking control for quadrotors with disturbances by using generalized regression neural networks. *Neurocomputing*, **460**, 243–255 (2021)

[30] DIERKS, T. and JAGANNATHAN, S. Output feedback control of a quadrotor UAV using neural networks. *IEEE Transactions on Neural Networks*, **21**(1), 50–66 (2010)

[31] RANDAL, W. B. *Small Unmanned Aircraft*: *Theory and Practice*, Princeton University Press, New Jersey, 30–38 (2012)

[32] ISLAM, S., LIU, P. X., and EI-SADDIK, A. Robust control of four-rotor unmanned aerial vehicle with disturbance uncertainty. *IEEE Transactions on Industrial Electronics*, **62**(3), 1563–1571 (2015)

[33] MAHOOTCHI, M., TIZHOOSH, H., and PONNAMBALAM, R. K. Oppositional extension of reinforcement learning techniques. *Information Sciences*, **275**, 101–114 (2014)

[34] DAVIDE, Z., PIETER, R. R., and SANDER, B. Learning continuous-time working memory tasks with on-policy neural reinforcement learning. *Neurocomputing*, **461**, 635–656 (2021)