Contents lists available at YXpublications

International Journal of Applied Mathematics in Control Engineering

Journal homepage: http://www.ijamce.com

Robust Sliding Mode Flight Controller for Quad-Rotor UAV with Image Recognition Xiaole Zhang^a, Jian Fu^{a,*}, Dong Wang^a, Ruoyin Xie^a

^a School of Energy and Power Engineering, Nanjing University of Sciences and Technology, Nanjing, 210094, CHN

TICLE INFO	ABSTRACT
cle history: eived 5 September 2021 epted 20 December 2021 ilable online 22 December 2021	Based on sliding mode control and image recognition method, a kind of trajectory tracking control systems is proposed in this paper for quad-rotor UAV with parameter uncertainty and external interference. This control system is divided into four different control loops which are position loop, velocity loop, attitude loop and angular rate loop according to time scale separation principle. And Hough Transform method is also utilized to analyze the tracking error between the centerline of track and flight position. Then, autonomous trajectory tracking control system of quad-rotor UAV is designed here. The simulation results show the effectiveness of the designed sliding mode controller and image recognition algorithm.
words: ing mode controller ge recognition ectory tracking d-rotor UAV	

Published by Y.X.Union. All rights reserved.

1. Introduction

Flight simulation

A R Arti Rec Acc Ava Key Slid Ima Traj Qua

In recent years, four-axis aircraft has attracted great attention from researchers in many countries (e.g., Louro P. et al., 2019; Tapan et al., 2018; Bao P., et al., 2019). The advantages of quad-rotor unmanned aerial vehicles (UAVs) include small size, simple mechanical structure and low cost compared with traditional aircraft. They have unique maneuverability features like hovering, take off and landing vertically (e.g., Mofid O et al., 2018; Ullah S et al., 2020). However, considerable uncertainty and external disturbances also exsits in the quad-rotor UAV system(e.g., Chen, Q et al., 2018), due to the stong state, input aerodynamic coulped, under-actuated and high nonlinear, etc. Then, it is necessary to find a robust control system to complete specific flight missions.

Sliding mode control, which is a robust nonlinear control method, has been proved available for quad-rotor UAV flight control system (e.g., Adilet T et al., 2021; Sun Y X et al., 1980; Feng K P et al., 2018). It ensures the dynamic performance of the system by designing the sliding mode dynamic surface with strong robustness(Shi S et al., 2018). This strategy is also widely combined with other control methods such as data-driven control (e.g. Yuyang T et al., 2020), fault-tolerant control (e.g. Jing W et al., 2018.), backstepping sliding mode algorithm (e.g. Yang, Z., et al., 2018) to design a suitable flight control system for quad-rotor UAVs. Based on image recognition and sliding mode control method, this paper focus on the design process

of robust tracking flight controller in quad-rotor UAVs.

The general control structure in this paper is given as follows: Firstly, the initial image information taken by the sensors is sent to the image recognition modules. Then, quad-rotor UAV control system makes the decision of next flight action by comparing the information with previous one. This decision will be transformed to the control signals and send to the sliding mode control systems. And the rest of this paper is organized as follows: In Section II, the mathematical model of the quad-rotor UAV is provided. Section III shows the details of the designed control system. Image recognition and path tracking is carried out in Section IV. The simulation results are obtained in Section V. And Section VI concerns the concluding remarks of this paper.

2. Mathematical model

The quad-rotor UAV consists of a flight control system, four equallength booms, propellers, and two counter-rotating rotors. It is maneuvered by changing the control inputs of four propeller engine systems (e.g. Jing D et al., 2021). The development of variablecomplex motion equations for quadrotor has been extensively studied (e.g., Gaitan T A et al., 2013; Das A et al., 2009; Bouabdallah S et al., 2004). Table 1 summarizes the main characteristics of quad-rotor UAV used in this study, and simulation model of the quad-rotor UAV can be shown as Fig.1.



Fig.1. Cross configuration and motor orientation

Parameter	Value
Length	180 mm
Width	180 mm
Moment of Inertia	0.00005828, 0.00007169,
Woment of metha	0.00010000
Length between the center of lift	0.044123
force and reference torque	

Table1 Main characteristics of quad-rotor UAV

The dynamical model (e.g. Wang C Y et al., 2021) of this system can be expressed as follow:

$$\begin{cases} \dot{x} = V_{ex} \\ \dot{y} = V_{ey} \\ \dot{z} = V_{ez} \end{cases}$$
(1)

$$\begin{cases} \dot{V}_{ex} = (\sin\psi\sin\varphi + \cos\psi\sin\theta\cos\varphi)\frac{U_1}{m} \\ \dot{V}_{ey} = (-\cos\psi\sin\varphi + \sin\psi\sin\theta\cos\varphi)\frac{U_1}{m} \\ \dot{V}_{ez} = (\cos\theta\cos\varphi)\frac{U_1}{m} - g \end{cases}$$
(2)

$$\begin{aligned} \dot{\phi} &= p + (q\sin\phi + r\cos\phi) \cdot \tan\theta \\ \dot{\theta} &= q\cos\phi - r\sin\phi \\ \dot{\psi} &= \frac{q\sin\phi + r\cos\phi}{\cos\theta} \end{aligned} \tag{3}$$

$$\begin{cases} \dot{p} = \frac{I_{YY} - I_{ZZ}}{I_{XX}} qr - \frac{J_{TP}}{I_{XX}} q\Omega + \frac{U_2 l}{I_{XX}} \\ \dot{q} = \frac{I_{ZZ} - I_{XX}}{I_{YY}} pr + \frac{J_{TP}}{I_{YY}} p\Omega + \frac{U_3 l}{I_{YY}} \\ \dot{r} = \frac{I_{XX} - I_{YY}}{I_{ZZ}} pq + \frac{U_4}{I_{ZZ}} \end{cases}$$
(4)

where x, y, z represent the positions along ground axis; ϕ , ϑ , ψ represent the Euler angles; V_{ex} , V_{ey} , V_{ez} are velocities along ground axis; p, q, r are angular rates; I_{XX} , I_{YY} , I_{ZZ} are the moments of inertia; m is the mass of the quadcopter; l represents the distance between UAV barycenter and rotary wing center; $U = [U_1, U_2, U_3, U_4]$ represents input control amount for each channel:

$$\begin{cases} U_{1} = b \left(\Omega_{1}^{2} + \Omega_{2}^{2} + \Omega_{3}^{2} + \Omega_{4}^{2} \right) \\ U_{2} = b \left(\Omega_{4}^{2} - \Omega_{2}^{2} \right) \\ U_{3} = b \left(\Omega_{3}^{2} - \Omega_{1}^{2} \right) \\ U_{4} = d \left(\Omega_{2}^{2} + \Omega_{4}^{2} - \Omega_{1}^{2} - \Omega_{3}^{2} \right) \end{cases}$$
(5)

where b represents the lift coefficient; Ω_i represent the rotational

speed of the *i* the rotor; *d* represents the reverse torque coefficient.

3. Contorl system design

Consider a class of non-linear system as follow:

$$\dot{\boldsymbol{x}} = f(\boldsymbol{x}) + g(\boldsymbol{x}) \cdot \boldsymbol{u} \tag{6}$$

where $\mathbf{x} = [x_1, \dots, x_n]^T \in \mathbb{R}^n$ is system state, $\mathbf{u} \in \mathbb{R}^n$ is control input, invertible matrix $g(\mathbf{x}) \in \mathbb{R}^n$, and $f: D \to \mathbb{R}^n$ is continuously differentiable mapping from $D \subset \mathbb{R}^n$ to \mathbb{R}^n .

The PI switching surface is designed as follow:

$$\boldsymbol{s}(\boldsymbol{x}_e) = \boldsymbol{x}_e + \boldsymbol{k}_1 \, \boldsymbol{x}_e dt \tag{7}$$

where $\mathbf{s}(\mathbf{x}_e) = [\mathbf{s}_1 \ \mathbf{s}_2 \ \mathbf{s}_3]^T$, $\mathbf{x}_e = \mathbf{x} - \mathbf{x}_c$ is desired system state, coefficient $k_1 = diag\{k_{11}, \dots, k_{mn}\}$.

In order to reduce the chattering phenomenon in slidng mode control, saturation function is introduced to reaching law.

According to taking the derivative of Eq.(7), one can get

$$\dot{\boldsymbol{s}}(\boldsymbol{x}_e) = \dot{\boldsymbol{x}}_e + k_1 \cdot \boldsymbol{x}_e = -k_2 \cdot \tanh(\boldsymbol{s}) \tag{8}$$

Lyapunov function is constructed as $V = \frac{1}{2}s^{T}s$, then it is noted that

$$V = \mathbf{s}^{*} \mathbf{s} = \mathbf{s}^{*} (-k_{2} \cdot \tanh(\mathbf{s}))$$

$$= -k_{2} \cdot [\mathbf{s}_{1} \ \mathbf{s}_{2} \ \mathbf{s}_{3}] \cdot \begin{bmatrix} \tanh(|\mathbf{s}_{1}|) & 0 & 0 \\ 0 & \tanh(|\mathbf{s}_{2}|) & 0 \\ 0 & 0 & \tanh(|\mathbf{s}_{3}|) \end{bmatrix} \cdot \begin{bmatrix} \operatorname{sgn}(\mathbf{s}_{1}) \\ \operatorname{sgn}(\mathbf{s}_{2}) \\ \operatorname{sgn}(\mathbf{s}_{3}) \end{bmatrix}$$

$$= -k_{2}(\mathbf{s}_{1} \cdot \tanh(|\mathbf{s}_{1}|) \cdot \operatorname{sgn}(\mathbf{s}_{1}) + \mathbf{s}_{2} \cdot \tanh(|\mathbf{s}_{2}|) \cdot \operatorname{sgn}(\mathbf{s}_{2})$$

$$+ \mathbf{s}_{3} \cdot \tanh(|\mathbf{s}_{3}|) \cdot \operatorname{sgn}(\mathbf{s}_{3}))$$

$$= -k_{2}(|\mathbf{s}_{1}| \cdot \tanh(|\mathbf{s}_{1}|) + |\mathbf{s}_{2}| \cdot \tanh(|\mathbf{s}_{2}|) + |\mathbf{s}_{3}| \cdot \tanh(|\mathbf{s}_{3}|)) \leq 0 \quad (9)$$

where hyperbolic tangent function

$$\tanh(s) = \frac{\exp(s) - \exp(-s)}{\exp(s) + \exp(-s)}$$
(10)

Basing on Lyapunov stability theory, the designed control systems are stable. The control systems have four independent control loops (e.g. Jian F et al., 2011) which are angular rate loop, velocity loop, attitude loop and position loop. And architecture of this flight system in this paper is shown as Fig.2.

3.1 Position Loop

Equations for this position loop are given in Eq.(1), and expressed as follow:

$$\dot{\boldsymbol{X}}_{p} = \boldsymbol{f}(\boldsymbol{X}_{p}) + \boldsymbol{g}(\boldsymbol{X}_{p}) \cdot \boldsymbol{V}$$
(11)

where position $X_p = [x; y; z]'$ are given as states; velocity $V = [V_{ex}; V_{ey}; V_{ez}]'$ are given as control inputs; $f(X_p) = [0;0;0]$ and $g(X_p) = diag\{1,1,1\}$; It means that the designed controller should send the reference velocity to next loop.

The tracking errors X_e between the desired outputs X_c and actual outputs X_c are defined as

$$\boldsymbol{X}_{e} = \boldsymbol{X}_{p} - \boldsymbol{X}_{c} \tag{12}$$



Fig.2. Control structure of quad-rotor UVA

It is noted that

$$\dot{X}_e = f(X_p) + g(X_p) \cdot V - \dot{X}_c$$
(13)

Then, the PI switching surfaces for the state X_p are given by:

$$s(\boldsymbol{X}_{p}) = \boldsymbol{X}_{e} + k_{p1} \int \boldsymbol{X}_{e} dt \tag{14}$$

Hence, the control inputs V can be expressed as:

$$\mathbf{V} = (\mathbf{g}(\mathbf{X}_{p}))^{-1}(-\mathbf{f}(\mathbf{X}_{p}) - k_{p1} \cdot \mathbf{X}_{e} - k_{p2} \cdot \tanh(s) + \dot{\mathbf{X}}_{c}) \quad (15)$$

where k_{p1} , k_{p2} are designed coefficients and $k_{p1} > 0$, $k_{p2} > 0$.

3.2 Velocity Loop

Equations for this loop are given in Eq.(2). In order to design a class of affine nonlinear controller, virtual control inputs are introduced as $\hat{u}_{V1} = \sin \varphi U_1$, $\hat{u}_{V2} = \sin \theta \cos \varphi U_1$ and $\hat{u}_{V3} = \cos \theta \cos \varphi U_1$. The simplified velocity system is expressed as follow.

$$\begin{cases} \dot{V}_{ex} = \frac{(\hat{u}_{v_1}\sin\psi + \hat{u}_{v_2}\cos\psi)}{m} \\ \dot{V}_{ey} = \frac{(-\hat{u}_{v_1}\cos\psi + \hat{u}_{v_2}\sin\psi)}{m} \\ \dot{V}_{ez} = \frac{\hat{u}_{v_3}}{m} - g \end{cases}$$
(16)

where

$$\theta = \arctan(\frac{\hat{u}_{V2}}{\hat{u}_{V3}}) ; \ \varphi = \arctan\left(\frac{\frac{u_{V1}}{\hat{u}_{V3}}}{1 + (\frac{\hat{u}_{V2}}{\hat{u}_{V3}})^2}\right) ; \ U_1 = \sqrt{\frac{\hat{u}_{V2}^2 + \hat{u}_{V3}^2}{2\cos\varphi}} ;$$

velocity $V = [V_{ec}; V_{ec}]'$ are given as states; $u_V = [\hat{u}_{V1}; \hat{u}_{V2}; \hat{u}_{V3}]'$ are given as virtual control inputs; θ_c , φ_c are control inputs, and the combined form of control equations can be described as:

$$\dot{\boldsymbol{V}} = \boldsymbol{f}_{\boldsymbol{V}}(\boldsymbol{V}) + \boldsymbol{g}_{\boldsymbol{V}}(\boldsymbol{V}) \cdot \boldsymbol{u}_{\boldsymbol{V}}$$
(17)

where

$$f_{V}(V) = \begin{bmatrix} 0\\ 0\\ -g \end{bmatrix} \qquad \qquad g_{V}(V) = \begin{bmatrix} \frac{\sin\psi}{m} & \frac{\cos\psi}{m} & 0\\ -\frac{\cos\psi}{m} & \frac{\sin\psi}{m} & 0\\ 0 & 0 & \frac{1}{m} \end{bmatrix}$$

The tracking errors V_e between the desired outputs V and reference outputs V_c are defined as

It is noted that

$$\dot{V}_e = f_V(V) + g_V(V) \cdot u_V - \dot{V}_c$$
(19)

(18)

Then, the sliding mode surfaces used in this loop are expressed as follow:

 $V_e = V - V_c$

$$s(\mathbf{V}) = \mathbf{V}_e + k_{V1} \left[\mathbf{V}_e dt \right]$$
(20)

and the virtual control inputs u_v are given as:

$$\boldsymbol{u}_{V} = (\boldsymbol{g}_{V}(\boldsymbol{V}))^{-1} [-\boldsymbol{f}_{V}(\boldsymbol{V}) - \boldsymbol{k}_{V1} \cdot \boldsymbol{V}_{e} - \boldsymbol{k}_{V2} \cdot \tanh(s) + \boldsymbol{V}_{c}] \qquad (21)$$

where designed coefficients $k_{V1} > 0$, $k_{V2} > 0$.

3.3 Attitude Loop

In this subsystem, $\Omega = [\phi; \theta; \psi]'$ are given as states; angular rate $\omega = [p;q;r]'$ are given as control inputs. Equations for this loop are given in Eq.(3), and can be also expressed as follow:

$$\boldsymbol{\Omega} = \boldsymbol{f}_{\Omega}(\boldsymbol{\Omega}) + \boldsymbol{g}_{\Omega}(\boldsymbol{\Omega}) \cdot \boldsymbol{\omega}$$
(22)

where

$$\boldsymbol{f}_{\Omega}(\boldsymbol{\Omega}) = \begin{bmatrix} 0\\ 0\\ 0 \end{bmatrix} \qquad \boldsymbol{g}_{\Omega}(\boldsymbol{\Omega}) = \begin{bmatrix} 1 & \tan\theta\sin\phi & \tan\theta\cos\phi\\ 0 & \cos\phi & -\sin\phi\\ 0 & \frac{\sin\phi}{\cos\theta} & \frac{\cos\phi}{\cos\theta} \end{bmatrix}$$

The tracking errors Ω_e between the desired reference Ω and actual outputs Ω_e are defined as

$$\boldsymbol{\Omega}_{e} = \boldsymbol{\Omega} - \boldsymbol{\Omega}_{c} \tag{23}$$

It is noted that

$$\hat{\boldsymbol{\Omega}}_{e} = \boldsymbol{f}_{\Omega}(\boldsymbol{\Omega}) + \boldsymbol{g}_{\Omega}(\boldsymbol{\Omega}) \cdot \boldsymbol{\omega}_{c} - \hat{\boldsymbol{\Omega}}_{c}$$
(24)

Then, the PI sliding surfaces is expressed as follow:

$$s(\boldsymbol{\Omega}_{e}) = \boldsymbol{\Omega}_{e} + k_{\Omega I} \int \boldsymbol{\Omega}_{e} dt \qquad (25)$$

The designed coefficients $k_{\Omega l} > 0$, $k_{\Omega 2} > 0$ and the the control inputs ω are given as

$$\boldsymbol{\omega} = \boldsymbol{g}_{\Omega}(\boldsymbol{\Omega})^{-1} [-\boldsymbol{f}_{\Omega}(\boldsymbol{\Omega}) - \boldsymbol{k}_{\Omega 1} \cdot \boldsymbol{\Omega}_{be} - \boldsymbol{k}_{\Omega 2} \cdot \tanh(s) + \dot{\boldsymbol{\Omega}}_{c}] \quad (26)$$

3.4 Angular Rate Loop

In this subsystem, $\omega = [p;q;r]'$ are given as states and $U_c = [U_2;U_3;U_4]'$ are given as control inputs. Equations for this position

loop are given in Eq.(4), and expressed as follow:

$$\dot{\boldsymbol{\omega}} = \boldsymbol{f}_{\omega}(\boldsymbol{\omega}) + \boldsymbol{g}_{\omega}(\boldsymbol{\omega}) \cdot \boldsymbol{U}_{c}$$
(27)

where

$$\boldsymbol{f}_{\boldsymbol{\omega}}(\boldsymbol{\omega}) = \begin{bmatrix} \frac{I_{YY} - I_{ZZ}}{I_{XX}} qr - \frac{J_{TP}}{I_{XX}} q\Omega \\ \frac{I_{ZZ} - I_{XX}}{I_{YY}} pr + \frac{J_{TP}}{I_{YY}} p\Omega \\ \frac{I_{XX} - I_{YY}}{I_{ZZ}} pq \end{bmatrix} \qquad \boldsymbol{g}_{\boldsymbol{\omega}}(\boldsymbol{\omega}) = diag \left\{ \frac{l}{I_{XX}}; \frac{l}{I_{YY}}; \frac{1}{I_{ZZ}} \right\}$$

The tracking errors ω_e between the desired outputs ω and actual outputs ω_e are defined as

$$\boldsymbol{\omega}_e = \boldsymbol{\omega} - \boldsymbol{\omega}_c \tag{28}$$

It is noted that

$$\dot{\boldsymbol{\omega}}_{e} = \boldsymbol{f}_{\omega}(\boldsymbol{\omega}) + \boldsymbol{g}_{\omega}(\boldsymbol{\omega}) \cdot \boldsymbol{U}_{c} - \dot{\boldsymbol{\omega}}_{c}$$
(29)

Then, the PI switching surfaces are given by:

$$s(\boldsymbol{\omega}_{e}) = \boldsymbol{\omega}_{e} + k_{\omega 1} \int \boldsymbol{\omega}_{e} dt$$
(30)

Hence, the control inputs U_c can be expressed as:

$$\boldsymbol{U}_{c} = \boldsymbol{g}_{\omega}(\boldsymbol{\omega})^{-1} [-\boldsymbol{f}_{\omega}(\boldsymbol{\omega}) - \boldsymbol{k}_{\omega 1} \cdot \boldsymbol{\omega}_{e} - \boldsymbol{k}_{\omega 2} \cdot \tanh(s) + \dot{\boldsymbol{\omega}}_{c}] \quad (31)$$

where designed coefficients $k_{\omega 1} > 0, k_{\omega 2} > 0$.

4.Image recognition and path tracking

Simulated test track is generated and the quad-rotor UAV will tracing along the track centerline. The images taken by the bottom camera as shown in Fig.3 are converted to RGB color format, and three 120 * 160 matrices of red, green and blue are generated.



Fig. 3. Simulated track image captured by the bottom camera

Step1 Image Pre-processing

Due to the image obtained in the simulation environment is ideal, the pretreatment directly process the three matrices. The elements in the three matrices represent corresponding color element values constituting the camera image, and these values vary from 0 to 255. **Step2** Gray Processing and Binarization

Binarization is an image processing technology that separates the pixel values of a digital image into two groups: white and black. Image binarization is used in this paper to separate simulation track and ground as shown in Fig.4.

All the pixel matrices meet the requirements (R>200,G<50,B<50) in the element matrix are extracted. These pixel matrices are combined to form a new image data matrix without changing the original size (e.g. Han H., 2014). If the value of the matrix element satisfies the proposed condition, then the new image data matrix element is 1, otherwise it is 0.



Fig.4. Image after gray processing and binarization

Step3 Hough Transform

According to the definition of Hough transformation, all points presenting a straight line in the original coordinate system, their slopes and intercepts are the same. The specific transformation process is as follows.



Fig.5. Coordinate transformation process

where θ is the angle between the correct expectation point and the intersection point at the previous moment.

In the image x - y coordinate space, the straight line passing through the point (x_i, y_i) is expressed as:

$$y_i = ax_i + b \tag{32}$$

where the parameter a is the slope, b is the intercept moment.

If x_i and y_i are considered as constants and the original parameters a and b are considered as variables, Eq.(31) can be expressed as

$$b = -x_i a + y_i \tag{33}$$

Conversely, all lines that intersect at the same point in the parameter space have points that correspond to them in the image coordinate space. According to this property, given some edge points in the image coordinate space, the equation of the line connecting these points can be determined by the Hough transform.

In practical applications, there is no way to represent a straight-line equation of the form y = kx + b for a line of the form x = c. In this paper, the parametric equation $p = xcos\theta + ysin\theta$ is used, so that a point on the image plane corresponds to a curve on the parametric $p - \theta$ plane. Input the binary image matrix into the internal functions, and then the image matrix containing only the center line of the track can be obtained. Fig.6 shows the straight path centerline identified by Hough transform.



Fig.6. Centerline extraction using Hough transform

Step4 Path Tracking Algorithm

Fig.7 shows the simulation track used in this paper. For path tracking, the emphasis should be on path adaptability (e.g. Huang Z Q., 2021). The simple tracking algorithm adopted in this project assumes that the position of the aircraft particle is at the origin of the image coordinate system. Then, the buffer range BL is defined to set the distance of the aircraft from the tracking point. Iterating through the coordinates of all points on the centerline, and the coordinate of the point closest to the origin and buffer is recorded. This coordinate is the expected point at the next moment.



Fig.7. Simulation track

5.Simulation results

It is noted that the simulated flight trajectory of quad-rotor UAV is formed with several straight tracks. In the tracking process, the center line of the track after image recognition is used to represent the target points. The simulation process is designed as follows:

The total simulation flight time is set as 80 seconds. The four rotor UAV takes off at the 20th second, and then image recognition starts at the 25th second. Fig.8 and Fig.9 show the change of flight trajectory in x direction and y direction respectively.





Fig.8. x axis trajectory tracking

Fig.9. y axis trajectory tracking



Fig.10. Tracking error

Tracking error is found at the junction of the tracks, on account of the integral of the track signal error introduced in the flight control systems. When the aircraft is located at the junction of the tracks, the tracking errors integrators should be cleared to avoid overshoot on the new track. In addition, the maximum overshoot in x direction and y direction is reduced to 2%. Generally, the simulation result shows that the trajectory tracking is effective, and the algorithm is robust and accurate enough even in the presence of about 10% model uncertainty.

6.Conclusion

According to the dynamic equation of the quad-rotor UAV, this paper has designed a class of control algorithm for the flight trajectory tracking control problem. Combined with image recognition method, the proposed sliding mode controller is able to overcome the uncertainty of model parameters. Simulation result has verified the robustness of control system, and the accuracy of path tracking algorithm.

References

- Mofid, O., 2018. Mobayen S. Adaptive sliding mode control for finite-time stability of quad-rotor UAVs with parametric uncertainties. Isa Trans. 72, 1-14.
- Ullah,S., et al, 2020. Robust Integral Sliding Mode Control Design for Stability Enhancement of Under-actuated Quadcopter. International Journal of Control, Automation and Systems. 18(7), 1671-1678.
- Bao, P., et al, 2019. Fusion of Gas Concentration Map Based on UAV Olfaction Using Image Processing. International Journal of Applied Mathematics in Control Engineering. 2(1), 25-31.
- Pereira, F., et al, 2015. Embedded Image Processing Systems for Automatic Recognition of Cracks using UAVs. IFAC Papers OnLine. 48(10), 16-21.
- Chen, Q., et al,2018. Finite Time Synergetic Control for Quadrotor UAV with Disturbance Compensation. International Journal of Applied Mathematics in Control Engineering. 1(1), 31-38.
- Adilet T., et al, 2021. Development of control algorithm for a quadcopter. Procedia Computer Science. 179, 242-251.
- Sun Y X., 2018. Decoupling Control System. Journal of CheKiang University(Engineering and Technology Edition). 000(003), 20-52.
- Feng K P., et al, 2018. Fuzzy adaptive sliding mode control for quad-rotor UAV. Flight Dynamics. 36(6), 49-53.
- Shi, S., et al, 2018. Fractional-Order Sliding Mode Control for a Class of Nonlinear Systems via Nonlinear Disturbance Observer. International Journal of Applied Mathematics in Control Engineering. 1(1), 103-110.
- Yuyang, T.,2020. Fault Diagnosis and Fault Tolerant Control of Quadrotor UAV. Lanzhou University of Technology.
- Jing, W., et al, 2018. NFTSM-based Fault Tolerant Control for Quadrotor Unmanned Aerial Vehicle with Finite-Time Convergence. International Federation of Automatic Control. 51(24), 229-252.
- Yang, Z., et al, 2018.Back-stepping Sliding Mode with Unidirectional Auxiliary Surfaces for HSV with Attitude Constrains. International Journal of Applied

X. Zhang et al. / IJAMCE 4 (2021) 151-156

Mathematics in Control Engineering. 1(2),187-193.

- Jing D., et al, 2021. Prescribed Performance Back-stepping Control of Fast Trajectory Tracking for Quad-rotor Aircraft PAN Shi-hua. Control Engineering of China. 28(11), 2119-2208.
- Gaitan, T., et al, 2013. Modeling and Robust Attitude Control of a Quadrotor System. International Conference on Electrical Engineering, Computing Science and Automatic Control, IEEE. 10(9), 7-13.
- Das,A., et al, 2009. Dynamic Inversion with Zero-Dynamics Stabilisation for Quadrotor Control. IET Control Theory & Applications. 3(3), 303-314.
- Bouabdallah S., et al, 2004. Design and Control of an Indoor Micro Quadrotor. International Conference on Robotics & Automation, IEEE. 5(4), 4393-4398.
- Wang, C, Y., et al, 2019. Design of Trajectory Tracking Control System for Quad-Rotor UAV. Electronics Optics & Control. 26(3), 103-107.
- Jian, F., et al, 2011. Sliding mode control for a miniature helicopter. International Conference on Automation & Computing, IEEE. 17(9), 98-103.
- Han,H., et al, 2014. Path Recognition Algorithm of Tracking Robots Based on Image Sensor. Journal of Donghua University (English Edition). 31(2), 137-140.
- Huang,Z,Q., 2011. Design of Tracking Robots Section PID Control Algorithm Based on CCD Camera. Electronic Design Engineering. 19(2), 55-57.



Xiaole Zhang is currently pursuing her BS study at School of Energy and Power Engineering, Nanjing University of Sciences and Technology, Nanjing, China. Her research interest covers sliding mode control and flight control.





Jian Fu received his Ph.D. degree in control theory and control engineering from Nanjing University of Aeronautics and Astronautics. In the same year, he taught at Nanjing University of Science and Technology. Now, he is an associate professor at NJUST. He mainly engaged in nonlinear control, robust control, sliding mode control, adaptive observer design and so on.

Dong Wang is currently studying for his BS degree at the School of Energy and Power Engineering, Nanjing University of Science and Technology. His research interest covers aircraft control technology and Image recognition.



Ruoyin Xie is currently studying for his BS degree at the School of Energy and Power Engineering, Nanjing University of Science and Technology. She mainly engaged in dynamic system modeling and algorithm research.