

Research Article

Attitude Active Disturbance Rejection Control of the Quadrotor and Its Parameter Tuning

Suiyuan Shen  and Jinfa Xu 

National Key Laboratory of Rotorcraft Aeromechanics, Nanjing University of Aeronautics & Astronautics, Nanjing, China

Correspondence should be addressed to Jinfa Xu; 2728475686@qq.com

Received 4 August 2020; Revised 16 October 2020; Accepted 27 October 2020; Published 17 November 2020

Academic Editor: Seid H. Pourtakdoust

Copyright © 2020 Suiyuan Shen and Jinfa Xu. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The internal uncertainty and external disturbance of the quadrotor will have a significant impact on flight control. Therefore, to improve the control system's dynamic performance and robustness, the attitude active disturbance rejection controller (ADRC) of the quadrotor is established. Simultaneously, an adaptive genetic algorithm-particle swarm optimization (AGA-PSO) is used to optimize the controller parameters to solve the problem that the controller parameters are difficult to tune. The performance of the proposed ADRC is compared with that of the sliding mode controller (SMC). The simulations revealed that the dynamic performance and robustness of the ADRC is better than that of the SMC.

1. Introduction

A quadrotor aircraft is a kind of multirotor helicopter which can realize vertical take-off and landing. It can complete the mission in various harsh environments and is widely used in military and civil fields. With the increase of application fields, the requirements for quadrotor aircraft's actual control effect are higher and higher. There are always internal uncertainty (unmodeled dynamic or parameter uncertainty) and external uncertainty (unknown disturbance of external environment) in quadrotor aircraft, which will lead to a large error between the mathematical model and the actual flight, so the accuracy of attitude control cannot be guaranteed in the real plane, which makes the design of flight controllers more difficult. Early researchers used the Proportional Integral Differential (PID) algorithm to design the flight control system of quadrotor aircraft [1–3]. The structure is simple and easy to implement, but the antidisturbance ability is weak, and the control effect is general. Researchers have recently turned their attention to intelligent control, such as the fuzzy control [4–6] and neural network [7–9]. Although good results have been achieved, it still has the disadvantages of weak robustness and difficulty in improving system performance.

Active disturbance rejection control (ADRC) [10–14] is a new control technology that does not depend on the system model. It can estimate and compensate for the sum disturbance caused by various external disturbances and internal disturbances determined by the system's mechanism in real time. Combined with the exceptional nonlinear state error feedback, excellent control quality can be obtained. ADRC has small overshoot characteristics, fast response speed, high control precision, strong antidisturbance ability, and simple structure [15, 16]. Reference [17] designed an ADRC based on an improved extended state observer for multiple-input multiple-output systems and achieved good control results. Reference [18] designed four improved ADRC for the nonlinear model of a 6-DOF quadrotor system for height and attitude stabilization and compared the PID controller's performance. The simulation results showed that the improved ADRC is stable and can offset external disturbance and system uncertainty, which is better than the PID controller.

The defect of ADRC is that there are many controller parameters, so it is difficult to tune the controller parameters. The particle swarm optimization (PSO) algorithm is widely used in the optimization of controller parameters. This method integrates the idea of "individual learning" by simulating the foraging activities of natural creatures. The algorithm is simple

and has a memory function, and it also has the characteristics of rapid convergence [19–21]. Reference [22] used the PSO algorithm to optimize the parameters of the ADRC and achieved good results. However, the PSO algorithm's global search ability is low, the search accuracy is not high, and it is easy to fall into the optimal local solution [23]. The genetic algorithm (GA) is a natural optimization algorithm with a diverse population and strong global search ability [24]. Reference [25] used GA to optimize the ADRC's parameters and achieved good results. Because of this, this paper proposes an adaptive genetic algorithm-particle swarm optimization (AGA-PSO), which combines the excellent global search ability of AGA to avoid particles falling into the local extremum prematurely, thus increasing the rate of obtaining optimal global solution and accelerating the convergence speed of the algorithm.

Aiming at the quadrotor's attitude control problem, this paper designs an ADRC strategy by analyzing the mathematical model of the quadrotor and considering the uncertain internal disturbance and external disturbance of the system. Aiming at the problem that the ADRC's parameters are difficult to tune, a parameter tuning method of the ADRC based on the AGA-PSO hybrid optimization algorithm is designed. Its effect is verified through simulation experiments. Finally, the quadrotor's attitude control with disturbance is simulated, which demonstrates the robustness and antisturbance of the ADRC.

2. ADRC for Quadrotor Attitude Control

2.1. Flight Dynamic Model. The quadrotor model shown in Figure 1 is a typical nonlinear controlled plant. Four-rotor systems generate control forces and control moments and form attitude motion under control forces and control moments. Therefore, the controlled plant of the quadrotor is a multiple-input multiple-output nonlinear system. The specific parameters of the quadrotor are shown in Table 1. The figure shows two sets of coordinate systems, namely, the North-East-Earth navigation coordinate system (indicated by G, the origin is at the take-off point, the coordinate axis x_G points north, y_G points east, and z_G is perpendicular to the ground down) and the aircraft body coordinate system (represented by P, the origin is at the center of mass of the body, the coordinate axis x_p is parallel to the plane of the paddle and points forward, y_p is parallel to the plane of the paddle and points to the right, and z_p is perpendicular to the plane of the paddle and points downward).

Assuming that the quadrotor is a uniformly symmetrical rigid body, and ignoring factors such as the rotor gyro effect, References [26, 27] obtain the quadrotor attitude mathematical model as follows.

$$\ddot{\theta} = \frac{(J_z - J_y)}{J_x} \dot{\phi} \dot{\psi} + \frac{lK_L}{J_x} (u_1 - u_2), \quad (1)$$

$$\ddot{\phi} = \frac{(J_x - J_z)}{J_y} \dot{\theta} \dot{\psi} + \frac{lK_L}{J_y} (u_4 - u_3), \quad (2)$$

$$\ddot{\psi} = \frac{(J_y - J_x)}{J_z} \dot{\theta} \dot{\phi} + \frac{K_n}{J_z} (u_1 + u_2) + \frac{K_s}{J_z} (u_3 + u_4), \quad (3)$$

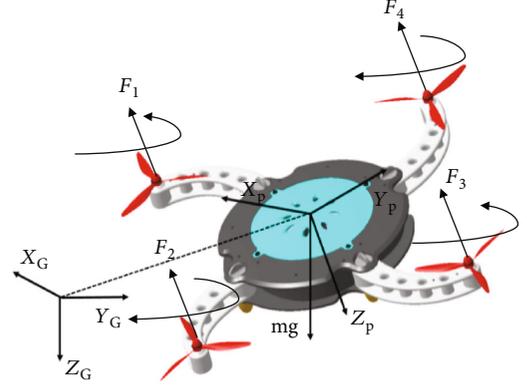


FIGURE 1: Schematic diagram of the quadrotor shape and coordinate system.

TABLE 1: Parameters of the quadrotor.

Parameter	Value	Unit
l	0.2	m
J_x	0.03	kg·m ²
J_y	0.03	kg·m ²
J_z	0.04	kg·m ²
K_L	0.12	N/V
$K_n = -K_s$	0.004	N·m/V

where ψ , θ , and ϕ are the yaw angle, pitch angle, and roll angle, respectively; u_1 , u_2 , u_3 , and u_4 are the voltage of the 4 rotor motors; l is the distance from the rotor axis to the center of mass of the quadrotor; J_x , J_y , and J_z are the moment of inertia of the pitch, roll, and yaw axes, respectively; K_L is the propeller lift coefficient; K_s is the clockwise torque coefficient of the rotor; and K_n is the counterclockwise torque coefficient of the rotor.

The attitude angle dynamic equations are second-order nonlinear coupled equations, so that the unknown nonlinear function of each attitude angle dynamic equation is

$$f_1(\bullet) = \frac{(J_z - J_y)}{J_x} \dot{\phi} \dot{\psi}, \quad (4)$$

$$f_2(\bullet) = \frac{(J_x - J_z)}{J_y} \dot{\theta} \dot{\psi}, \quad (5)$$

$$f_3(\bullet) = \frac{(J_y - J_x)}{J_z} \dot{\theta} \dot{\phi}, \quad (6)$$

the unknown disturbance quantity is ω_1 , ω_2 , and ω_3 , and the virtual control quantity is recorded as $M = [M_1 M_2 M_3]$; then, the three-axis attitude angle dynamic

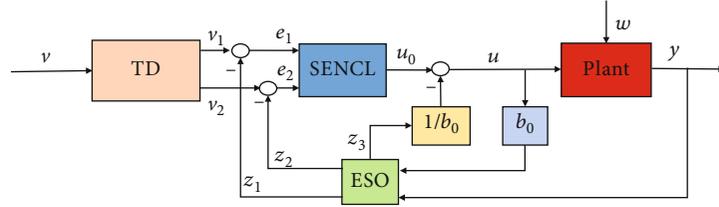


FIGURE 2: Block diagram of ADRC.

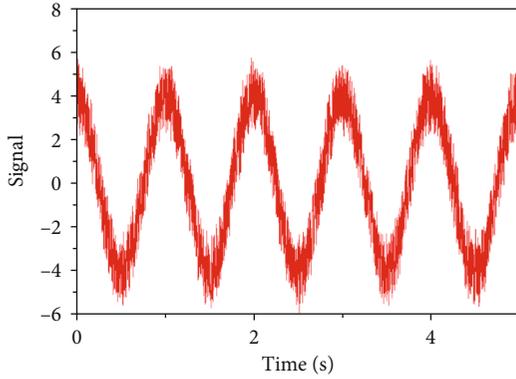


FIGURE 3: Input signal.

equation is expressed as

$$\ddot{\theta} = \omega_1 + f_1(\bullet) + M_1, \quad (7)$$

$$\ddot{\phi} = \omega_2 + f_2(\bullet) + M_2, \quad (8)$$

$$\ddot{\psi} = \omega_3 + f_3(\bullet) + M_3, \quad (9)$$

and the dynamic coupling $\omega_i + f_i(\bullet)$ of the internal and external disturbances is estimated by ESO through state expansion and is dynamically compensated as the feedback signal of the control law. The system becomes a linear integrator series control system. The system output after dynamic compensation linearization is $Y = [y_1 \ y_2 \ y_3]^T = [\theta \ \phi \ \psi]^T$.

The actual control quantity $U = [u_1 \ u_2 \ u_3 \ u_4]$ of the four motors of the quadrotor is obtained by transforming the virtual control quantity, namely,

$$\begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{bmatrix} = \begin{bmatrix} \frac{IK_L}{J_x} & 0 & \frac{K_n}{J_z} \\ -\frac{IK_L}{J_x} & 0 & \frac{K_n}{J_z} \\ 0 & -\frac{IK_L}{J_y} & \frac{K_s}{J_z} \\ 0 & \frac{IK_L}{J_y} & \frac{K_s}{J_z} \end{bmatrix}^T \begin{bmatrix} M_1 \\ M_2 \\ M_3 \end{bmatrix}. \quad (10)$$

2.2. ADRC Features. ADRC consists of the tracking differentiator (TD), extended state observer (ESO), and state error

nonlinear control law (SENCL), as shown in Figure 2. TD is used to quickly obtain the tracking signal and its derivative signal of the input signal. ESO estimates the extended state formed by the state quantity, internal uncertainty, and external disturbances. Then, the nonlinear combination of the state quantity's error is used as the control quantity of the system to realize the effective control of the quadrotor.

2.2.1. TD. Taking the second-order system as an example, a discrete form of the nonlinear tracking differentiator is designed as follows:

$$\begin{cases} x_1(k+1) = x_1(k) + hx_2(k), \\ x_2(k+1) = x_2(k) + hfst(x_1(k) - v(k), x_2(k), r, h), \end{cases} \quad (11)$$

$$fst(x_1, x_2, r, h) = \begin{cases} -r \operatorname{sign}(a), & |a| > d, \\ -r \frac{a}{d}, & |a| \leq d, \end{cases} \quad (12)$$

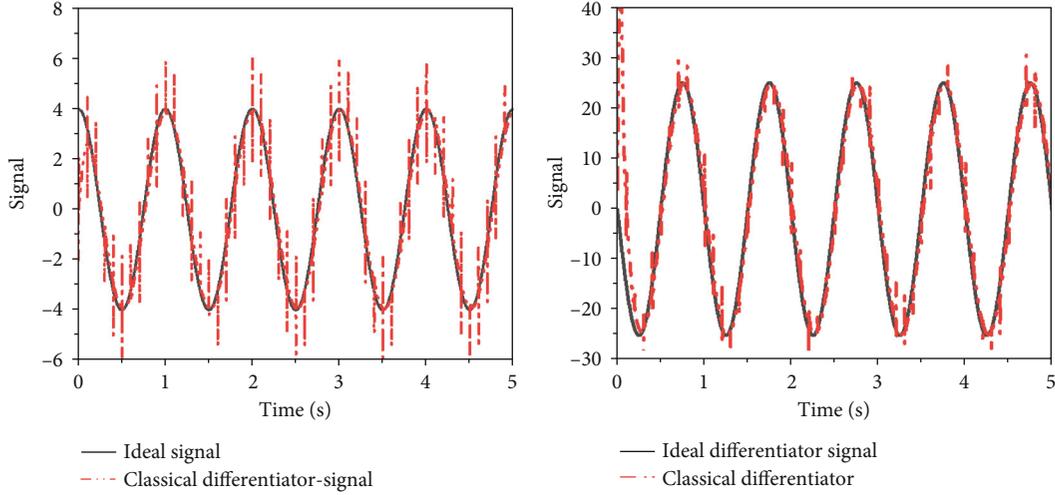
$$a = \begin{cases} x_2 + \frac{a_0 - d}{2} \operatorname{sign}(y), & |y| > d_0, \\ x_2 + \frac{y}{h}, & |y| \leq d_0, \end{cases} \quad (13)$$

$$\begin{cases} d = rh, \\ d_0 = dh, \\ a_0 = \sqrt{d^2 + 8r|y|}, \\ y = x_1 + x_2h. \end{cases} \quad (14)$$

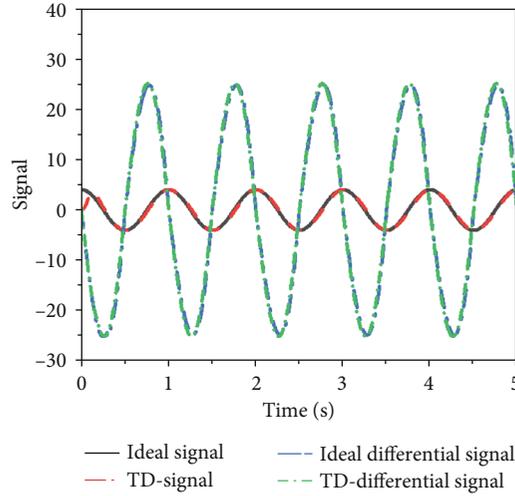
The output state variable $x_1(k)$ is the fast tracking signal of the input signal $v(k)$, and the state variable $x_2(k)$ is its differential signal. It not only overcomes the influence of noise signal amplification but also eliminates the phenomenon of origin flutter. In the formula, k is expressed as the k -th moment of signal sampling, h is the discrete sampling time step, r is the speed factor, and the h and r parameters are adjustable, which determine the speed of the output signal tracking.

2.2.2. Simulation Verification of TD. Given the input signal $v(t) = 4 \cos(2\pi t) + n(t)$, as shown in Figure 3, the discrete sampling period is $h = 0.001$, the velocity factor is $r = 800$, and the noise signal $n(t)$ is the random white noise of amplitude 2.

When a classic differentiator implemented with a short time constant inertia link performs signal differentiation processing, the differentiator's signal amplification effect is



(a) Classical differentiator output tracking signal and differential signal



(b) Output tracking signal and differential signal of TD

FIGURE 4: Effect curve of TD.

noticeable. Equation (15) is the classic differentiator processing formula.

$$y(t) = \frac{1}{T} \left(v - \frac{1}{Ts+1} v \right). \quad (15)$$

Record the output of the second inertia link as $v = (1/Ts + 1)v$; then, when the input signal changes slowly and the time constant T is small, there is

$$y(t) = \frac{1}{T} (v(t) - \bar{v}(t)) \approx \frac{1}{T} (v(t) - v(t-T)) \approx \dot{v}(t). \quad (16)$$

Obviously, the smaller the time constant T , the closer the output $y(t)$ is to the derivative $\dot{v}(t)$.

The tracking signal and differential signal simulation results of the classical differentiator and tracking differentiator are shown in Figure 4. The noise amplification effect of the products shown in Figure 4(a) is noticeable. As shown in Figure 4(b), the noise suppression effect of the results is noticeable.

2.2.3. ESO. Suppose the nonlinear uncertain plant with external disturbance is

$$\begin{cases} \dot{x}_1 = x_2, \\ \dot{x}_2 = f(x, w, t) + b(t)u, \\ y = x_1. \end{cases} \quad (17)$$

Let $x_3 = f(x, w, t) + (b(t) - b_0)u$ be the new expanded state variable, and x_3 contains its own uncertainties and

TABLE 2: Parameters of ADRC.

	TD	ESO	SENCL
Value	$h = 0.01, r = 18$	$\xi = 0.004, \beta_1 = 150, \beta_2 = 320, \beta_3 = 3000, \alpha_1 = 0.5, \alpha_2 = 0.25$	$\alpha_1 = 0.7, \alpha_2 = 2, \varsigma = 0.02, \beta_{01} = 5, \beta_{02} = 0.4$

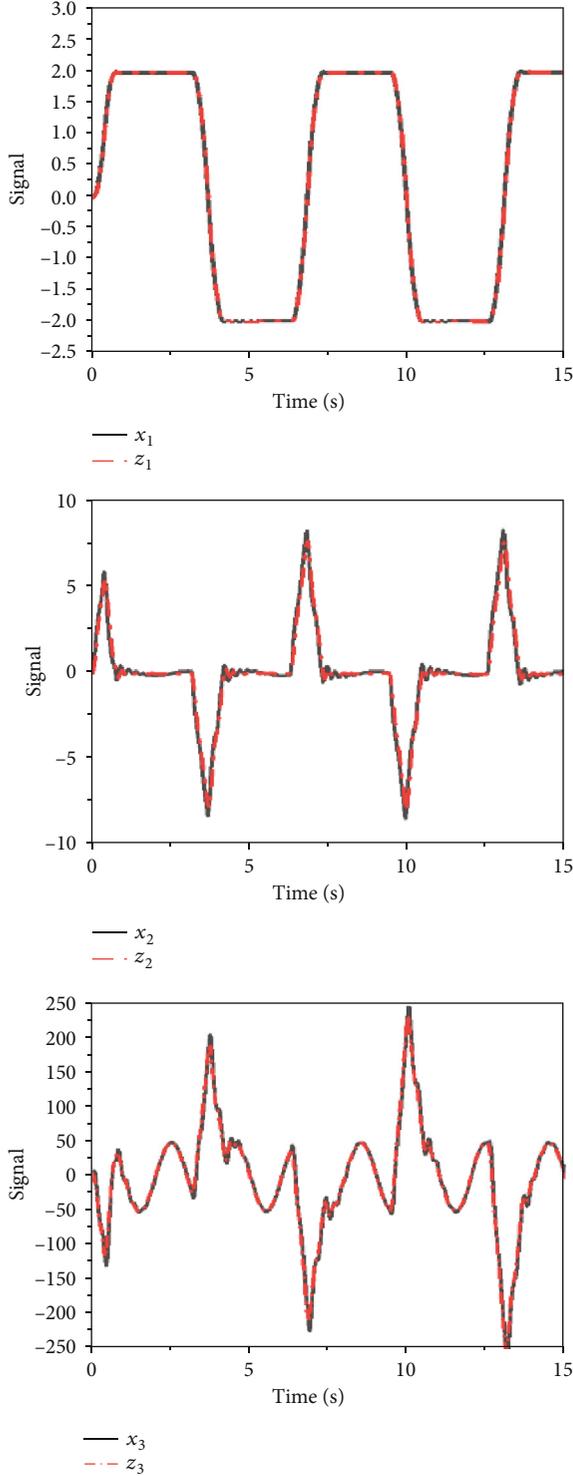


FIGURE 5: The observation effect of ESO.

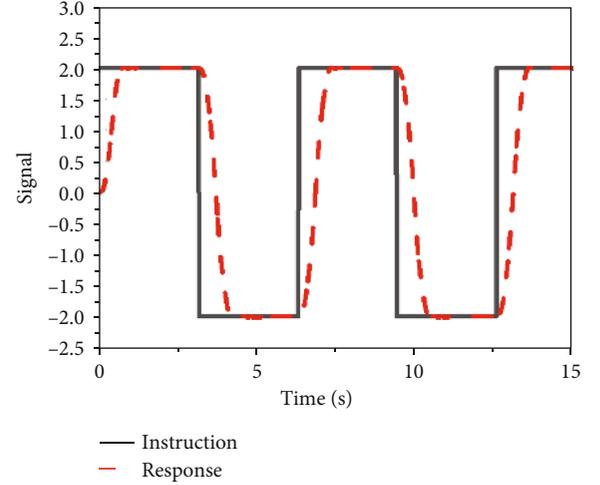


FIGURE 6: Square wave response tracking.

external disturbances; let $\dot{x}_3 = g(t)$, and the system (17) expands into a new control system, namely,

$$\begin{cases} \dot{x}_1 = x_2, \\ \dot{x}_2 = x_3 + b_0 u, \\ \dot{x}_3 = g(t), \\ y = x_1, \end{cases} \quad (18)$$

where u and y are the input and output signals of the system; $f(x, w, t)$ is the unknown nonlinear system function; $w(t)$ is the disturbance; x and \dot{x} represent the corresponding derivative of x ; and b_0 is the compensation coefficient.

Design the following ESO for the extended system (18) of the system (17).

$$\begin{cases} e = z_1 - y, \\ \dot{z}_1 = z_2 - \beta_1 e, \\ \dot{z}_2 = z_3 - \beta_2 \text{fal}(e, \lambda_1, \zeta) + b_0 u, \\ \dot{z}_3 = -\beta_3 \text{fal}(e, \lambda_2, \zeta), \end{cases} \quad (19)$$

$$\text{fal}(e, \lambda, \zeta) = \begin{cases} \frac{e}{\zeta^{1-\lambda}}, & |e| \leq \zeta, \\ |e|^\lambda \text{sign}(e), & |e| > \zeta, \end{cases} \quad (20)$$

where $\lambda_i > 0$ ($i = 1, 2, 3, \dots, n$), β_i ($i = 1, 2, 3, \dots, n+1$) is the gain coefficient, ζ is the step size, and $\text{fal}(\bullet)$ is the saturation nonlinear function; the output variable of ESO (19) can track the state variable of the system (18), namely,

$$z_1 \rightarrow x_1, z_2 \rightarrow x_2, z_3 \rightarrow x_3. \quad (21)$$

TABLE 3: ADRC tuning parameters.

Parameter	Description	
TD	r	The larger is the faster the tracking speed, but the more obvious the noise amplification
	h	The larger is the better the filtering effect, but the more serious the phase loss of the tracking signal
	β_1	Increase β_1 to suppress the overshoot of the system, but too large will lead to longer rise time
	β_2	Increase can reduce the steady-state error of the system, but too large will cause oscillation
ESO	β_3	Increase can reduce the lag of ESO's estimation of the total disturbance of the controlled system, but too large will cause oscillation
	ζ_1	Too large causes the current linear feedback to be too large, ESO cannot track the nonlinear signal well, and too small will cause jitter
SENCL	β_{01}	Too large causes the system overshoot to increase, the rise time becomes shorter, and the system oscillates
	β_{02}	Increase can reduce the oscillation, but too large will lead to longer rise time
	ζ_2	Same as ζ_1

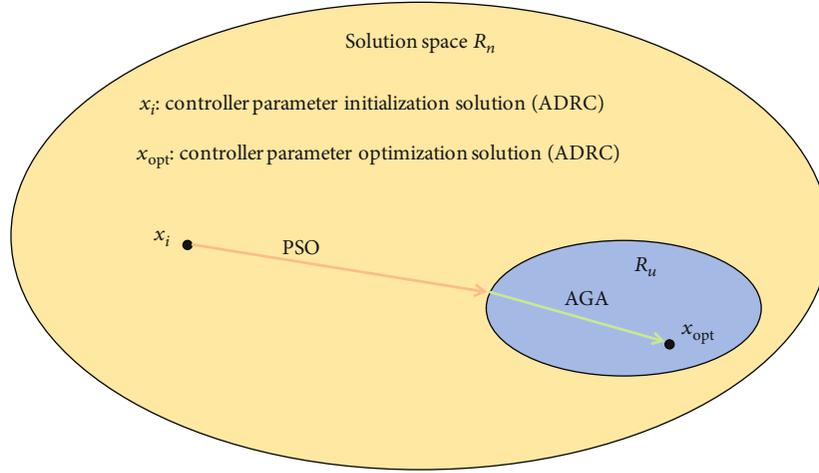


FIGURE 7: AGA-PSO algorithm diagram.

The estimated value z_{n+1} of the expanded state variable is used as a compensation value for the unknown plant and external disturbance, which linearizes the controlled system and can effectively suppress the interference signal in the state variable.

The control signal of the system (17) is designed as $u = u_0 - (z_3/b_0)$; then,

$$\begin{cases} \dot{x}_1 = x_2, \\ \dot{x}_2 = x_3 + b_0 \left(u_0 - \frac{z_3}{b_0} \right), \\ y = x_1. \end{cases} \quad (22)$$

Under the extended state estimation value compensation, the system becomes linear controlled plants, namely,

$$\begin{cases} \dot{x}_1 = x_2, \\ \dot{x}_2 = b_0 u_0, \\ y = x_1, \end{cases} \quad (23)$$

that is, the dynamic compensation linearization processing of the nonlinear controlled plant.

The essence of the system (18) is to regard the series part of the system (17) which is different from the standard integral series type as the total disturbance (including internal disturbance and external disturbance). ESO estimates the total disturbance by using the input and output information of the system and compensates for the total disturbance by designing a control law, thus transforming the system into a standard integral series structure.

2.2.4. *SENCL*. Taking the second-order system as an example, a SENCL algorithm using a combination of fal functions is

$$\begin{cases} e_1 = v_{11} - z_1, \\ e_2 = v_{21} - z_2, \\ u_0 = \beta_{01} \text{fal}(e_1, \alpha_1, \varsigma) + \beta_{02} \text{fal}(e_2, \alpha_2, \varsigma), \end{cases} \quad (24)$$

where v_{11} and v_{21} are the tracking signal and differential signal of TD, respectively; β_{01} and β_{02} are the adjustable control

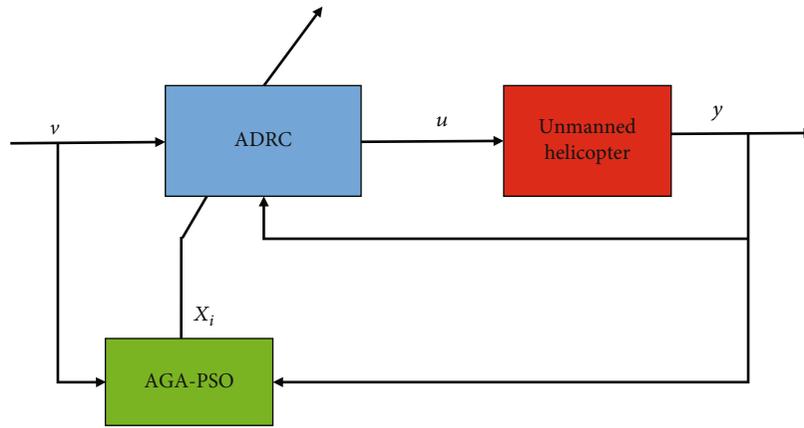


FIGURE 8: AGA-PSO algorithm combined with the ADRC system.

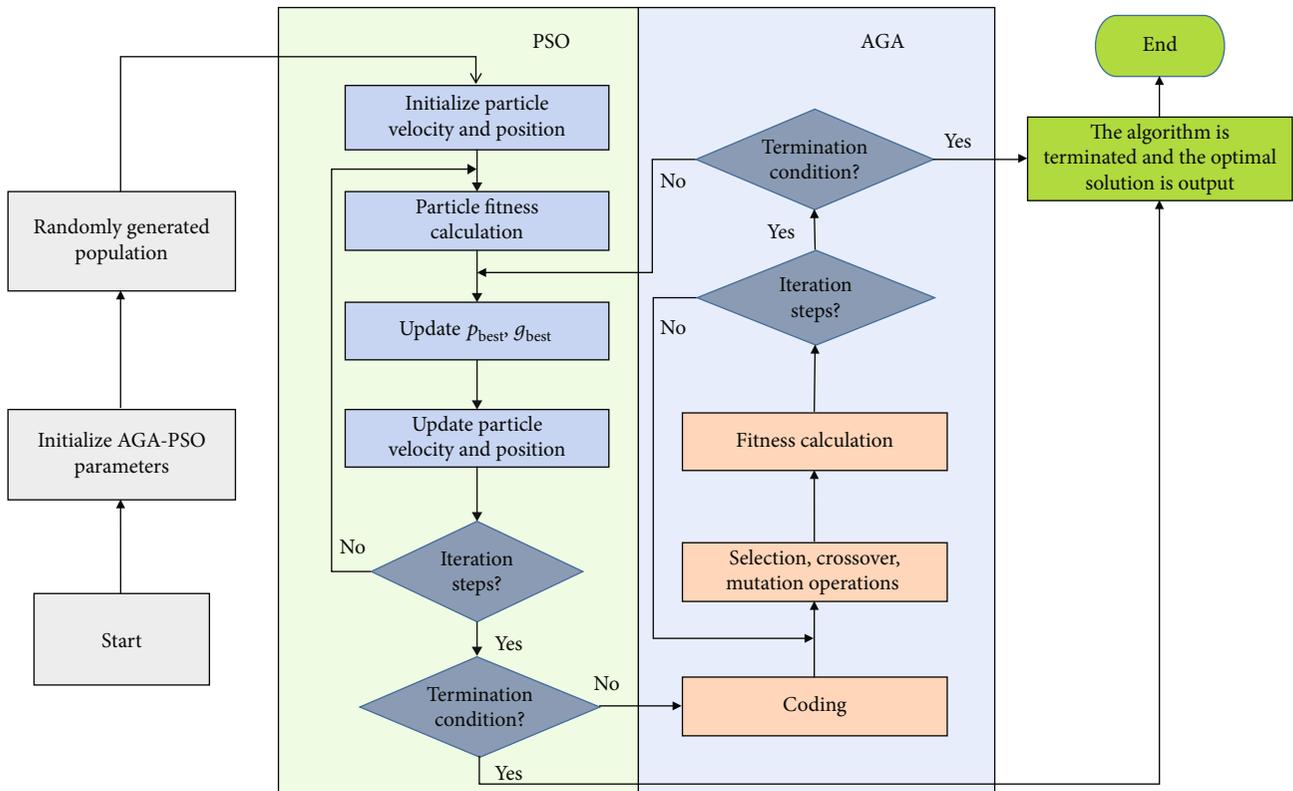


FIGURE 9: AGA-PSO flow chart.

gain; and fal function can effectively suppress the occurrence of signal chattering and has a good signal filtering effect.

2.2.5. *ADRC Simulation Verification Example.* Given the second-order controlled plant system model,

$$\begin{cases} \dot{x}_1 = x_2, \\ \dot{x}_2 = -30x_2 + 50 \sin(\pi t) + 200u, \end{cases} \quad (25)$$

where $f(x_1, x_2) = -30x_2$ represents an unknown function, $w(t) = 50 \sin(\pi t)$ represents a disturbance, and $b_0 = 200$.

After discretizing the abovementioned continuous domain controlled plant, simulation is performed with a square wave signal with an amplitude of 2. TD is designed according to Section 2.2.1, ESO is designed according to Section 2.2.3, and SENCL is designed according to Section 2.2.4; the relevant parameters are shown in Table 2.

The simulation results are shown in Figure 5. $z_1, z_2,$ and z_3 are the state estimates of ESO that respond to the process curve; $x_1, x_2,$ and x_3 are the response curves of the system (25) state variables and expansion state. It can be seen from the figure that the output estimation value of ESO can track the state variables of the system well, including the extended

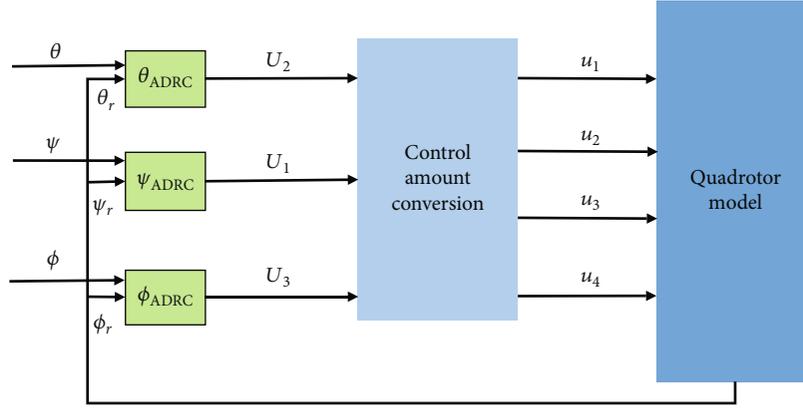


FIGURE 10: Block diagram of the control structure of the three-axis attitude angle by ADRC.

TABLE 4: The parameters of AGA-PSO and PSO.

Parameter	Description	AGA-PSO	PSO
NG	Evolutionary algebra	100	—
sizepop	Genetic algorithm population size	50	—
swarmsize	Number of particles	50	50
D	Number of parameters to be tuned	21	21
c_1	Acceleration factor	2	2
c_2	Acceleration factor	2	2
w_{\max}	Maximum inertia weight	0.9	0.9
w_{\min}	Minimum inertia weight	0.6	0.6
M	Number of iterations	100	100

state variables of the internal disturbance and external disturbance of the system.

Figure 6 shows the results of the square wave input control response under the action of SENCL, including the white noise to measure the interference signal. It can be seen that the control has no overshoot and can quickly enter a stable state, indicating that the controller has strong robustness.

3. Controller Parameter Tuning

3.1. Parameter Tuning Principle. Different parameters have different effects on the performance of the ADRC. Table 3 lists the parameters and functions of the ADRC that need to be tuned. It can be seen that the ADRC needs to tune 9 parameters, of which ζ_1 and ζ_2 are predetermined according to the characteristics of the controlled plant and do not need to be changed within a certain range. For this reason, the following AGA-PSO algorithm only sets 7 parameters.

3.2. AGA-PSO. The core idea of the AGA-PSO algorithm is to use the PSO algorithm to quickly find the neighborhood of the global optimal solution to form a new population, in which the AGA is used to obtain the optimal global solution. Its advantage is to avoid the blindness of

the initial stage of the AGA and shorten the global convergence time. The advantages of the two algorithms complement each other to achieve rapid acquisition of the optimal global solution. Figure 7 shows the relationship between the two algorithms.

Suppose that a population $\{X_i\}$ is composed of N particles, a single particle is composed of a D -dimensional space, and its position and velocity are described as

$$X_i = (x_{i1}, x_{i2}, x_{i3}, \dots, x_{iD}), \quad i = 1, 2, \dots, N, \quad (26)$$

$$V_i = (v_{i1}, v_{i2}, v_{i3}, \dots, v_{iD}), \quad i = 1, 2, \dots, N. \quad (27)$$

The particle velocity determines the search direction of the particle, and the position x_i of each particle is a potential solution to the target problem. The optimal position found by a single particle individual is the individual optimal, denoted as

$$p_{\text{best}} = (p_{i1}, p_{i2}, p_{i3}, \dots, p_{iD}), \quad i = 1, 2, \dots, N. \quad (28)$$

The optimal position found by all the particles in the population is the global optimal, denoted as

$$g_{\text{best}} = (p_{g1}, p_{g2}, p_{g3}, \dots, p_{gD}). \quad (29)$$

The particle individual optimization search velocity and position update formulas are

$$v_{ij}(t+1) = \omega v_{ij}(t) + c_1 r_1 [p_{ij}(t) - x_{ij}(t)] + c_2 r_2 [p_{gj}(t) - x_{ij}(t)], \quad (30)$$

$$x_{ij}(t+1) = x_{ij}(t) + v_{ij}(t+1), \quad (31)$$

where r_1 and r_2 are random numbers between $[0, 1]$, c_1 and c_2 are learning factors, ω is the inertia weight, v_{ij} is the particle velocity, and x_{ij} is the particle position.

Inertia weight ω determines the influence of the previous velocity on the current velocity. A large ω is good for the particles to jump out of the optimal local solution,

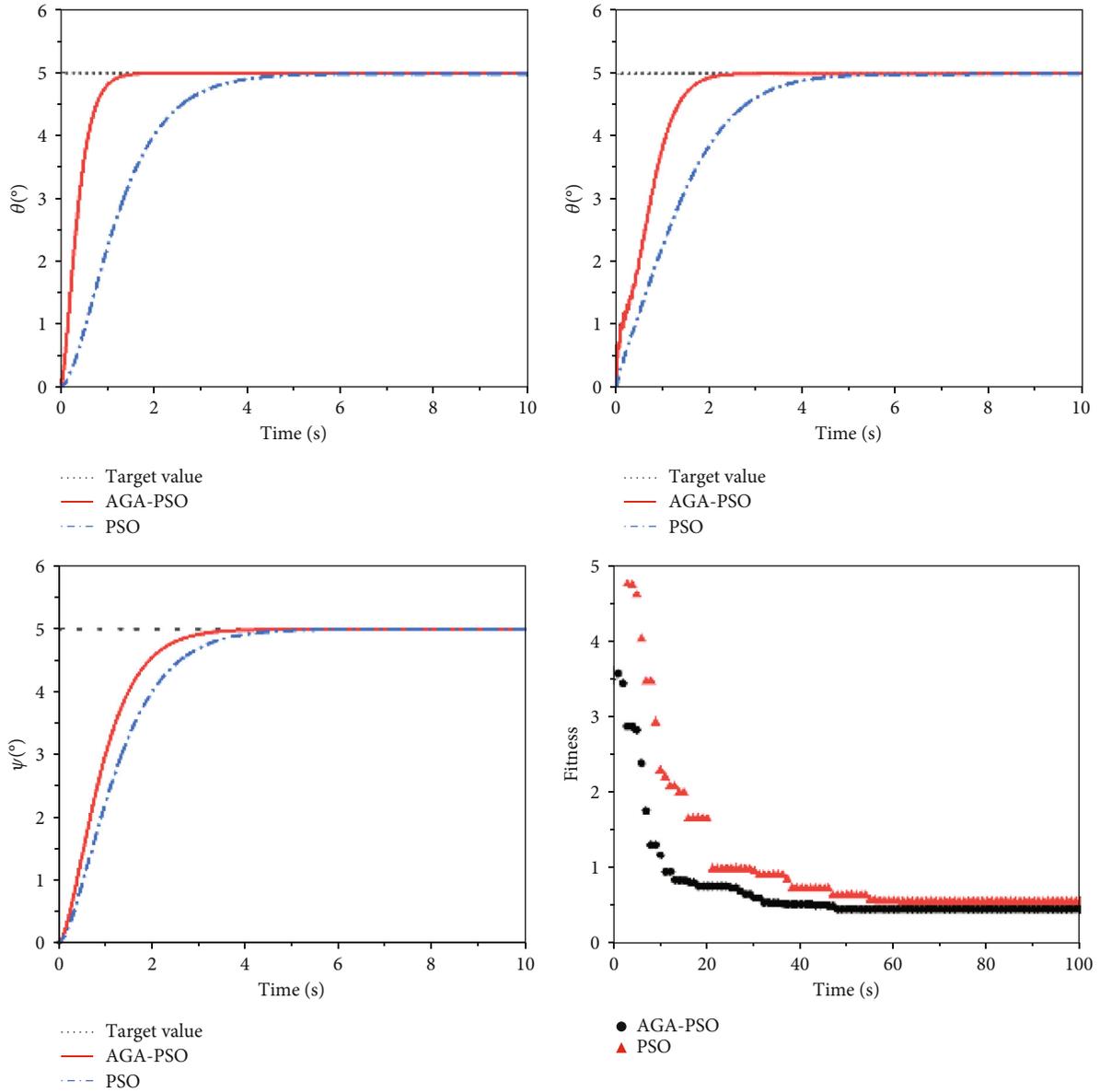


FIGURE 11: The response of attitude (step signal).

which is suitable for global search, and a small w is good for accurate local search. Taking into account the convergence of the algorithm and the balance between the algorithm's global search and local improvement, the following nonlinear adaptive law is used to modify the inertia weight coefficient, namely,

$$w = \begin{cases} w_{\min} - \frac{(w_{\max} - w_{\min})(f - f_{\min})}{f_{\text{avg}} - f_{\min}}, & f \leq f_{\text{avg}} \\ w_{\max}, & f > f_{\text{avg}}, \end{cases} \quad (32)$$

where w_{\min} and w_{\max} are the minimum and maximum values of w , respectively; f is the current fitness of particles; f_{avg} and f_{\min} are the average fitness and minimum

fitness of all particles, respectively. The integral time absolute error (ITAE) criterion is used to calculate the particle fitness of a single control loop.

$$J = \int_0^{\infty} t|e(t)|dt, \quad (33)$$

where $|e(t)|$ is the absolute value of the difference between the input and the actual output of the control system. The smaller the fitness value is, the better the solution is. The fitness calculation formula of the whole control system is

$$J_{\text{all}} = \sum_{i=1}^H \int_0^{\infty} t \left| \frac{e_i(t)}{Q_i(t)} \right| dt, \quad (34)$$

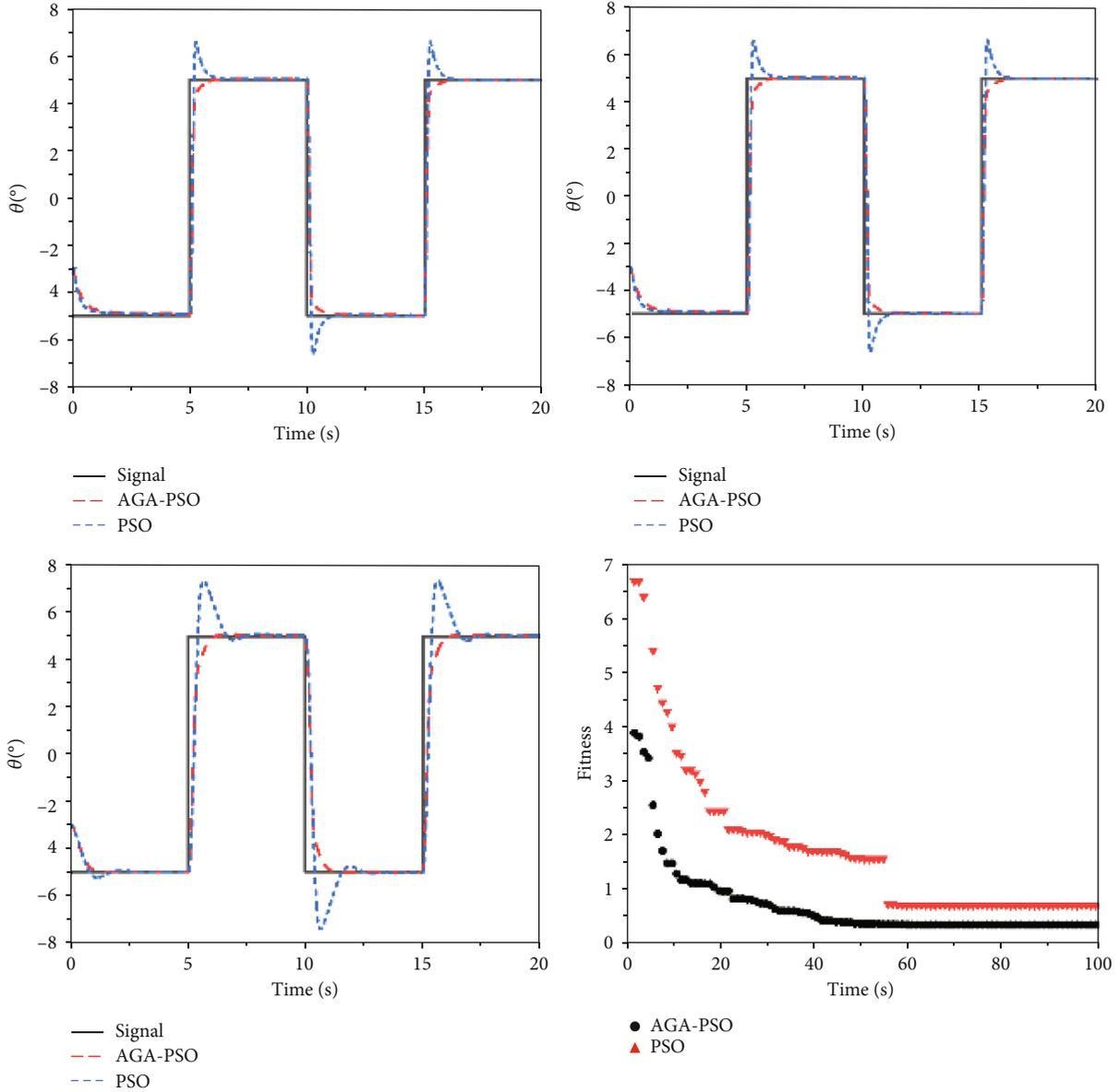


FIGURE 12: The response of attitude (square wave signal).

TABLE 5: Parameters of ADRC and SMC (square wave signal).

	TD	ESO	SENCL	SMC
ψ	$h = 0.021, r = 5.2$	$\xi = 0.004, \beta_1 = 4015, \beta_2 = 2011, \beta_3 = 50, \alpha_1 = 0.5, \alpha_2 = 0.25, b_0 = 0.06$	$\alpha_1 = 0.5, \alpha_2 = 0.25, \zeta = 1, \beta_{01} = 582, \beta_{02} = 127$	$\lambda_1 = 67, \lambda_2 = 232, a = 5, b = 3, p = 7, q = 5$
θ	$h = 0.022, r = 9.5$	$\xi = 0.1, \beta_1 = 6121, \beta_2 = 2982, \beta_3 = 53, \alpha_1 = 0.5, \alpha_2 = 0.25, b_0 = 0.4$	$\alpha_1 = 0.5, \alpha_2 = 0.05, \zeta = 3, \beta_{01} = 292, \beta_{02} = 29$	$\lambda_1 = 41, \lambda_2 = 156, a = 7, b = 2, p = 8, q = 5$
ϕ	$h = 0.021, r = 9.7$	$\xi = 0.1, \beta_1 = 6096, \beta_2 = 2973, \beta_3 = 49, \alpha_1 = 0.5, \alpha_2 = 0.25, b_0 = 0.4$	$\alpha_1 = 0.5, \alpha_2 = 0.05, \zeta = 3, \beta_{01} = 286, \beta_{02} = 31$	$\lambda_1 = 41, \lambda_2 = 156, a = 7, b = 2, p = 8, q = 5$

where H is the number of channels in the control system and $Q_i(t)$ represents the channel input signal, so as to ensure that the influence weight of each channel error on the fitness value is the same, that is, normalization processing.

The process of the PSO algorithm is as follows:

- (1) The particle swarm is randomly initialized, including the number of particles and the position and velocity of each particle. The dimension of each particle

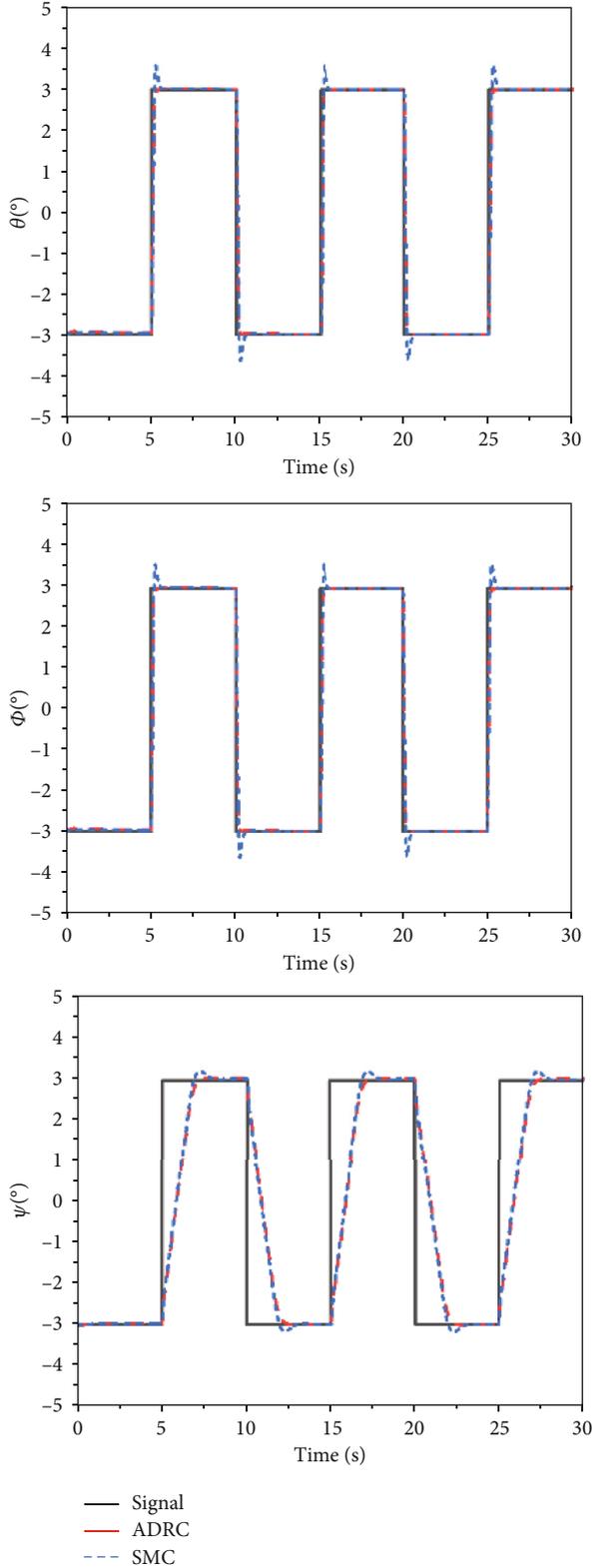


FIGURE 13: The simulation results of ADRC (square wave signal).

depends on the number of parameters of the ADRC. The position of each particle in the particle swarm is the potential solution of the controller parameters

TABLE 6: Parameters of the quadrotor (changed).

Parameter	Value	Unit
l	0.16	m
J_x	0.014	kg·m ²
J_y	0.014	kg·m ²
J_z	0.032	kg·m ²
K_L	0.09	N/V
$K_n = -K_s$	0.003	N·m/V

- (2) The particle position as a potential solution is substituted into the ADRC system for control simulation, and the fitness of each solution is calculated
- (3) The velocity and position of particles are updated iteratively according to formulas (30) and (31)
- (4) Analyze and determine the optimal solution of the current individual particle and the optimal global solution of the particle swarm, and update p_{best} and $\mathcal{G}_{\text{best}}$
- (5) Judge whether it has reached the number of iteration steps; if it is satisfied, output the final p_{best} and the corresponding fitness value, jump out of this algorithm, and enter the subsequent process; otherwise, return to (2)

The parameters of the ADRC obtain the global optimal neighborhood solution through the PSO algorithm. Then, through the iterative operation of selection, crossover, and mutation of the AGA, the target problem gradually approaches the optimal global. The process is as follows:

- (1) *Coding.* Use real number codes to initialize population genes.
- (2) *Selection.* Take out a certain number of individuals from the population, select them according to their fitness, and enter the progeny population to reproduce until the new population size is the same as the original population size.
- (3) *Crossover.* As a core operation of the genetic algorithm, new individuals can be obtained, that is, new possible solutions. If the crossover probability is too large, the randomness of the algorithm is too strong, and the convergence speed of the algorithm is too slow to effectively suppress premature maturity. The adaptive crossover probability operation as shown below will benefit parameter optimization:

$$P_c = \begin{cases} k_1 - (k_1 - k_2) \frac{(f_c - f_{\text{avg}})}{f_{\text{avg}} - f_{\text{min}}}, & f_c \geq f_{\text{avg}}, \\ k_1, & f_c < f_{\text{avg}}, \end{cases} \quad (35)$$

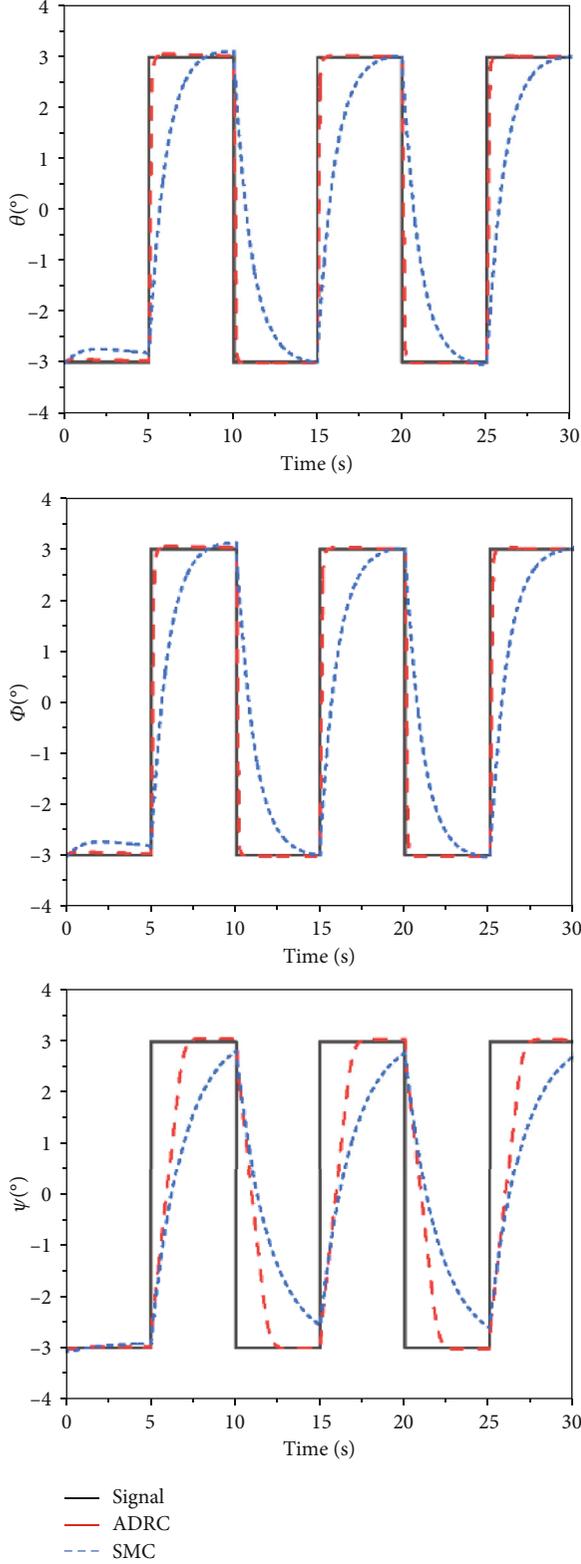


FIGURE 14: The simulation results of ADRC (model parameter changes).

where f_c is the larger fitness of the cross individuals; f_{avg} is the average fitness of the population; and generally, $k_1 = 0.9$ and $k_2 = 0.6$.

Randomly select two chromosomes from the population for crossover. First, determine the crossover probability P_c by formula (35) and then generate a random number in the range $(0, 1)$. When the random number is more significant than P_c , use the arithmetic crossover to develop a new chromosome. Otherwise, no crossover will be formed. The specific operation of the cross is as follows:

$$C_k = (c_1^k, c_2^k, \dots, c_N^k), \quad k = 1, 2, \quad (36)$$

$$C_i^1 = \lambda p_i^1 + (1 - \lambda) p_i^2, \quad (37)$$

$$C_i^2 = \lambda p_i^2 + (1 - \lambda) p_i^1, \quad (38)$$

where p_i^k is the chromosome before crossing, C_i^k is the chromosome after crossing, and λ is a constant. When two chromosomes are crossed, two chromosomes are randomly selected for crossover operation again until the crossing times reach the set number.

- (4) *Mutation*. As another core operation of the genetic algorithm, it can also obtain new individuals, that is, new possible solutions. If the mutation probability is too large, it will weaken the inheritance of excellent genes. If it is too small, it will easily fall into the local optimum, which cannot optimize the parameters. The following adaptive mutation probability operation is beneficial to parameter optimization:

$$P_m = \begin{cases} k_3 - (k_3 - k_4) \frac{(f - f_{avg})}{f_{max} - f_{min}}, & f' \geq f_{avg}, \\ k_3, & f' < f_{avg}, \end{cases} \quad (39)$$

where f is the fitness of the variant individual; f_{max} is the maximum fitness of the population; and generally, $k_3 = 0.1$ and $k_4 = 0.01$.

A chromosome is randomly selected from the population to mutate. Firstly, the mutation probability P_m is determined by formula (39), and then a random number in the range of $(0, 1)$ is generated. When the random number is greater than P_m , the polynomial mutation method is used to mutate; otherwise, no mutation will be carried out. The specific operation of mutation is as follows:

$$\delta = \begin{cases} (2u)^{1/(\eta+1)} - 1, & u < 0.5, \\ 1 - [2(1-u)]^{1/(\eta+1)}, & u > 0.5, \end{cases} \quad (40)$$

$$c_i' = c_i + \delta \Delta_{max}, \quad (41)$$

where c_i is the quasimutated gene, c_i' is the mutated gene, u is a random number between $(0, 1)$, η is the distribution index, and Δ_{max} is the maximum allowable disturbance range. After

TABLE 7: Parameters of ADRC and SMC (step signal).

	TD	ESO	SENCL	SMC
ψ	$h = 0.034, r = 4.2$	$\xi = 0.004, \beta_1 = 4172, \beta_2 = 1903, \beta_3 = 61, \alpha_1 = 0.5, \alpha_2 = 0.25, b_0 = 0.06$	$\alpha_1 = 0.5, \alpha_2 = 0.25, \zeta = 1, \beta_{01} = 571, \beta_{02} = 137$	$\lambda_1 = 45, \lambda_2 = 217, a = 4, b = 1, p = 11, q = 7$
θ	$h = 0.031, r = 8.7$	$\xi = 0.1, \beta_1 = 6319, \beta_2 = 2893, \beta_3 = 53, \alpha_1 = 0.5, \alpha_2 = 0.25, b_0 = 0.4$	$\alpha_1 = 0.5, \alpha_2 = 0.05, \zeta = 3, \beta_{01} = 252, \beta_{02} = 31$	$\lambda_1 = 37, \lambda_2 = 98, a = 9, b = 2, p = 4, q = 3$
ϕ	$h = 0.031, r = 8.6$	$\xi = 0.1, \beta_1 = 6306, \beta_2 = 2861, \beta_3 = 49, \alpha_1 = 0.5, \alpha_2 = 0.25, b_0 = 0.4$	$\alpha_1 = 0.5, \alpha_2 = 0.05, \zeta = 3, \beta_{01} = 255, \beta_{02} = 33$	$\lambda_1 = 37, \lambda_2 = 98, a = 9, b = 2, p = 4, q = 3$

the chromosome mutation is completed, a chromosome is randomly selected for mutation operation again until the mutation number reaches the set number.

In summary, the schematic diagram of the combination of the AGA-PSO algorithm and the ADRC system is shown in Figure 8. The flow of the AGA-PSO algorithm is shown in Figure 9. The process is as follows:

- (1) Initialize the AGA-PSO algorithm parameters, including c_1, c_2, w_{\min} , and w_{\max} of PSO, maximum evolution algebra NG, and other parameters
- (2) Generate a population randomly and then enter the PSO algorithm flow
- (3) Judge whether the solution result of the PSO algorithm has reached the global optimal solution condition (the minimum fitness value is less than the set value). If the result is satisfied, the result will be directly output as the controller optimization parameter. Otherwise, the global optimal neighborhood solution obtained by the PSO algorithm will be substituted into the AGA for encoding, selecting, crossover, and mutation operations to seek the optimal global solution
- (4) Individuals of the new population enter the ADRC system simulation and calculate the fitness value to determine whether the number of iteration steps is reached (the minimum population fitness is less than the set value, or the number of iteration steps is reached). Then, it is determined whether the termination condition is met (the minimum population fitness is less than the set value). If it is satisfied, the optimization result will be output, and the controller optimization parameters will be obtained. Otherwise, it will return to the PSO algorithm flow

3.3. AGA-PSO Algorithm Verification. Figure 10 shows the control structure block diagram of the three-axis attitude ADRC, which is composed of three control loops: yaw angle, pitch angle, and roll angle.

Each attitude loop needs to tune 7 parameters, and three loops need to tune a total of 21 parameters. The specific parameters of AGA-PSO and PSO are shown in Table 4. Given a step and square wave signal with an amplitude of 5° , Figures 11 and 12 show the attitude response and convergence curves of the unoptimized PSO and AGA-PSO algorithms, respectively. It can be seen that the AGA-PSO

algorithm has a shorter adjustment time for the three attitude loops than the PSO algorithm, and the overshoot is smaller, indicating that the AGA-PSO algorithm is feasible.

4. Simulation Verification of Attitude Control

The initial condition of simulation is as follows: set the yaw angle, pitch angle, and roll angle as 3° , and the simulation time is 30 s. The expected input of each loop is a square wave signal with an amplitude of 3° . The sliding mode controller (SMC) and the ADRC with the same control structure are selected. The AGA-PSO algorithm tuning ADRC parameters is shown in Table 5. The SMC is designed according to the literature [28], and the specific parameters are shown in Table 5.

Figure 13 shows the simulation results of the ADRC and SMC attitude angle control loops. It can be seen that the ADRC can track the set values of the three attitude loops more quickly without evident overshoot and steady-state error, which indicates that the ADRC attitude loop has better system performance.

To verify the ability of ADRC to overcome the model uncertainty (internal uncertainty), the ADRC and SMC parameters shown in Table 5 are kept unchanged, while the model parameter values are reduced by about 20%, as shown in Table 6. The simulation result is shown in Figure 14. It can be seen that the control performance of ADRC has little change, while the control performance of SMC has dropped sharply, and the three attitude angles can no longer be effectively controlled. This experiment shows that even if the internal parameters of the quadrotor change greatly, the ADRC can still maintain stable control performance.

In order to verify the resistance of ADRC to external disturbances, a step signal with an amplitude of 4° is given and a white noise with a peak value of 3° is added as an external disturbance in the attitude loops. The ADRC and SMC parameters are shown in Table 7. The simulation results shown in Figure 15 are obtained. It can be seen that the SMC has obvious oscillation in three attitude loops. However, the ADRC still keeps a good control effect, which shows that the ADRC designed in this paper has good response ability of tracking step input, and the control system also has better antidisturbance performance.

5. Conclusion

In this paper, the ADRC for attitude loops is designed by analyzing the quadrotor's dynamic attitude model. At the same

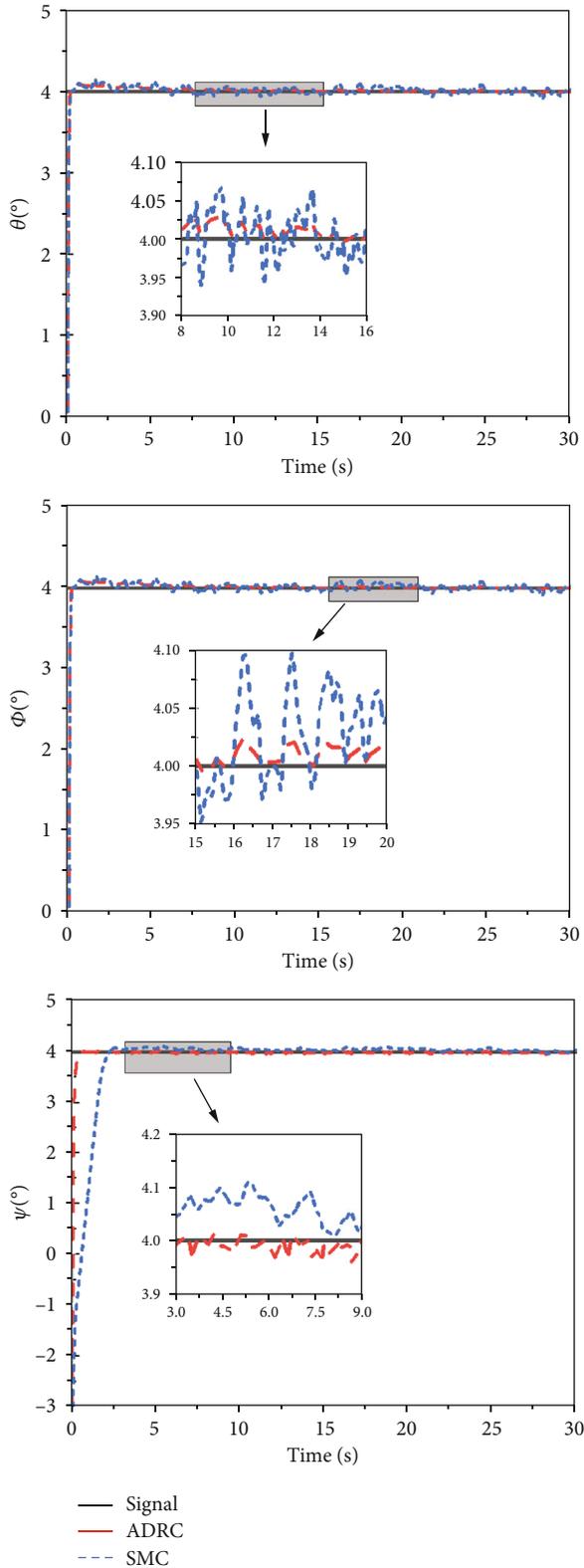


FIGURE 15: The simulation results of ADRC (step signal).

time, the AGA-PSO algorithm is used to optimize the parameters of ADRC. Finally, the simulation results show the following:

- (1) The AGA-PSO algorithm combines the global convergence of the AGA and the fast convergence of the PSO algorithm. Compared with the unmodified PSO algorithm, the AGA-PSO algorithm's performance is much improved, and a better controller parameter optimization effect is achieved
- (2) Each loop of ADRC has better dynamic tracking performance, small steady-state error, short adjustment time, strong disturbance rejection, and better robustness than that of SMC
- (3) The ADRC can effectively suppress disturbance by observing and compensating the internal uncertainty and external disturbance of the system

Data Availability

The data is simulated by MATLAB. All of the used data can be verified and replicated by reading the manuscript and simulation.

Conflicts of Interest

The authors declare that there is no conflict of interest regarding the publication of this paper.

Acknowledgments

This work was supported by the National Key Laboratory of Rotorcraft Aeromechanics Fund Program of China (9140C400504130C4148) and supported by the Priority Academic Program Development of Jiangsu Higher Education Institutions of China.

References

- [1] D. H. Shim, H. J. Kim, and S. Sastry, "Control system design for rotorcraft based unmanned aerial vehicle using time-domain system identification," in *Proceedings of the 2000 IEEE International Conference on Control Applications*, pp. 3632–3637, Anchorage, Alaska, USA, 2000.
- [2] H. J. Kim and D. H. Shim, "A flight control system for aerial robots: algorithms and experiments," *Control Engineering Practice*, vol. 11, no. 12, pp. 1389–1400, 2003.
- [3] B. Mettler, *Identification Modeling and Characteristics of Miniature Rotorcraft*, Kluwer Academic Publishers, Norwell, 2003.
- [4] B. Erginer and E. Altug, "Design and implementation of a hybrid fuzzy logic controller for a quadrotor VTOL vehicle," *International Journal of Control Automation and Systems*, vol. 10, no. 1, pp. 61–70, 2012.
- [5] C. Coza, C. Nicol, C. J. B. Macnab, and A. Ramirez-Serrano, "Adaptive fuzzy control for a quadrotor helicopter robust to wind buffeting," *Journal of Intelligent and Fuzzy Systems*, vol. 22, no. 5,6, pp. 267–283, 2011.
- [6] A. Tayebi and S. McGilvray, "Attitude stabilization of a VTOL quadrotor aircraft," *IEEE Transaction on Control Systems Technology*, vol. 14, no. 3, pp. 562–571, 2006.
- [7] H. Razmi and S. Afshinfar, "Neural network-based adaptive sliding mode control design for position and attitude control

- of a quadrotor UAV,” *Aerospace Science and Technology*, vol. 91, pp. 12–27, 2019.
- [8] T. Dierks and A. Jagannathan, “Neural network output feedback control of a quadrotor UAV,” in *47th IEEE Conference on Decision and Control. Cancun*, pp. 3633–3639, Mexico, 2008.
- [9] T. Dierks and S. Jagannathan, “Output feedback control of a quadrotor UAV using neural networks,” *IEEE Transactions on Neural Network*, vol. 21, no. 1, pp. 50–66, 2010.
- [10] J. Q. Han, “Control theory, is it a model analysis approach of a direct control approach,” *Journal of Systems Science and Mathematical Sciences*, vol. 4, pp. 328–335, 1989.
- [11] J. Q. Han, “Auto disturbances rejection control technique,” *Frontier Science*, vol. 1, pp. 24–31, 2007.
- [12] J. Q. Han, “A new type of controller–NLPID,” *Control and Decision*, vol. 6, pp. 401–407, 1994.
- [13] J. Q. Han, “State feedback realization of nonlinear control systems,” *Control and Decision*, vol. 6, no. 3, pp. 161–167+216, 1991.
- [14] J. Q. Han and W. A. N. G. Wei, “Nonlinear tracking-differentiator,” *Journal of Systems Science and Mathematical Sciences*, vol. 2, pp. 177–183, 1994.
- [15] W. R. Abdul-Adheem, I. K. Ibraheem, A. J. Humaidi, and A. T. Azar, “Model-free active input-output feedback linearization of a single-link flexible joint manipulator: an improved active disturbance rejection control approach,” *Measurement and Control*, vol. 26, p. 002029402091717, 2020.
- [16] J. Q. Han, “From PID technique to active disturbances rejection control technique,” *Control Engineering of China*, vol. 3, pp. 13–18, 2002.
- [17] W. R. Abdul-Adheem, I. K. Ibraheem, A. T. Azar, and A. J. Humaidi, “Improved active disturbance rejection-based decentralized control for MIMO nonlinear systems: comparison with the decoupled control scheme,” *Applied Sciences*, vol. 10, no. 7, p. 2515, 2020.
- [18] A. A. Najim and I. K. Ibraheem, “Altitude and attitude stabilization of UAV quadrotor system using improved active disturbance rejection control,” *Arabian Journal for Science and Engineering*, vol. 45, no. 3, pp. 1985–1999, 2020.
- [19] J. Kennedy and R. Eberhart, “Particle swarm optimization,” in *Proc of IEEE International Conference on Neural Networks*, pp. 1942–1948, Piscataway, 1995.
- [20] R. Eberhart and J. Kennedy, “A new optimizer using particle swarm theory,” in *Proc of the 6th International Symposium on Micro Machine and Human Science. Piscataway, NJ*, pp. 39–43, IEEE Press, 1995.
- [21] L. Y. Chuang, H. W. Chang, C. J. Tu, and C. H. Yang, “Improved binary PSO for feature selection using gene expression data,” *Computational Biology and Chemistry*, vol. 32, no. 1, pp. 29–38, 2008.
- [22] W. B. Yu, *A PSO algorithm for the parameter tuning of ADRC and its application research*, Shanghai University of Engineering Science, 2016.
- [23] Y. Da and G. Xiurun, *An improved PSO-based ANN with simulated annealing technique*, vol. 63, no. none, 2005Neuro-computing, 2005.
- [24] A. Agarwal and N. Nanavati, “Association rule mining using hybrid GA-PSO for multi-plantive optimisation,” *IEEE International Conference on Computational Intelligence and Computing Research*, vol. 31, pp. 1–7, 2017.
- [25] Z. Huang, Z. Chen, Y. Zheng, M. Sun, and Q. Sun, “Optimal design of load frequency active disturbance rejection control via double-chains quantum genetic algorithm,” *Neural Computing & Applications*, vol. 28, 2020.
- [26] Y. Li, Z. Q. Chen, and Z. X. Liu, “Attitude control of a quadrotor robot based on ADRC,” *Journal of Harbin Institute of Technology*, vol. 46, no. 3, pp. 115–118+123, 2014.
- [27] M. X. Bao, W. X. Shi, and J. C. Guo, “Attitude control of four-rotor aircraft based on ADRC,” *Automation & Instrumentation*, vol. 32, no. 2, pp. 10–13+22, 2017.
- [28] L. J. Luo, S. Y. Dian, and M. Pu, “Non-singular fast terminal sliding mode control of attitude for quad-rotor aircrafts,” *Electronics Optics & Control*, vol. 25, no. 4, pp. 12–15+60, 2018.