

Cascade PID Control of Quadrotor UAV Based on Particle Swarm Optimization Algorithm

Hanwen Chen

Ulster College, Shaanxi University of Science and Technology, Xi'an, Shaanxi, 710021, China

ABSTRACT

This paper addresses the parameter tuning problem of cascade PID control for quadrotor UAVs, which are characterized by nonlinearity, strong coupling, and underactuation. An improved particle swarm optimization (PSO) algorithm is proposed to optimize the parameters of the cascade PID controllers. First, the kinematic and dynamic models of the quadrotor UAV are established, and its motion characteristics are analyzed. The structure of cascade PID control and its application in UAV systems are introduced, highlighting the limitations of the traditional Ziegler-Nichols tuning method. To overcome these limitations, the PSO algorithm is enhanced by adopting the ITAE criterion as the fitness function and introducing a boundary rebound mechanism to improve particle diversity and avoid local optima. A complete cascade PID control system, including altitude, horizontal position, and attitude controllers, is built in MATLAB/Simulink based on the DJI Spark UAV model. The improved PSO algorithm is then applied to tune 36 PID parameters. Simulation results demonstrate that the optimized system achieves faster response, lower overshoot, and higher steady-state accuracy in both position tracking and attitude regulation, validating the effectiveness and superiority of the proposed method. Future work is discussed, including hardware-in-the-loop verification, robustness enhancement, and integration with advanced control strategies.

KEYWORDS

Quadrotor UAV; Cascade PID control; Particle swarm optimization; Parameter tuning; ITAE criterion; Boundary rebound; MATLAB/Simulink

1 Introduction

1.1 Research Background And Research Significance

In contemporary society, the quadrotor UAVs can move in the three-dimensional space. As an air machine which has four propellers, quadrotor UAV has the characteristic of vertical take-off and landing, hovering, flexible mobility and so on. Therefore, the quadrotor UAVs have a wide application prospect and a position of great research value in aspects of aerial photography, three-dimensional space reconstruction and multiple drones' cooperation^[1]. Nevertheless, due to the nonlinear characteristics (strong coupling, underactuation) of quadrotor UAVs, their dynamic models are relatively complex and extremely sensitive to external disturbances^[2]. So, it confronts various challenges to achieve stable and precise flight control. Although traditional PID control theory is applied widely in actual control engineering, even the proportions of application in some fields are more than 80%, parameters of PID are relatively fixed so that the UAV is hard to tackle slow response, large overshoot and so on. Thus, to combine optimization algorithm with PID control is the key to handle the problems^[3]. As an optimization algorithm based on swarm intelligence, particle swarm optimization (PSO) simulates group behaviors such as bird foraging. It searches out the best solution through mutual cooperation and information sharing among particles^[4].

1.2 The Main Research Work of This Thesis

In this study, the integration of particle swarm optimization (PSO) algorithm with cascade PID control introduces novel perspectives to aircraft control theory. This paper thoroughly investigates the mechanism of optimizing cascade PID parameters using the PSO algorithm and its impact on the complex system of a quadrotor UAV. Furthermore, it enhances the control theory for nonlinear and strongly coupled systems, providing valuable insights for related research.

2 Model of Quadrotor UAV And PID Control

2.1 Basic Principle of Quadrotor UAV

2.1.1 Fundamental States of Motion of Quadrotor UAV

Before studying the motions of UAV, this paper need to firstly introduce two coordinate frames to describe its motions: As for translation of drones, Earth-centered, earth-fixed (ECEF) coordinate system is utilized to describe its translation from one position to another position in three-dimension space. ECEF coordinate system is shown in Figure 1. As for rotation of drones, Body coordinate system is utilized to describe its rotation from one posture to another posture in three-dimension space. Body coordinate system is fixed in the body of the drone. It is shown in Figure 2. In the most case of rotation analyzing, Euler angles are used to depict rotation motions:

Roll angle(ϕ): rotates along x_b axis.

Pitch angle(θ): rotates along y_b axis.

Yaw angle(ψ): rotates along z_b axis.

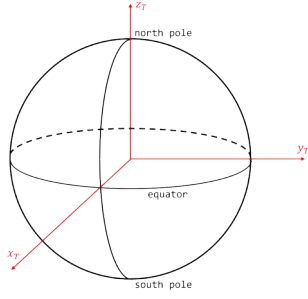


Figure 1 ECEF Coordinate System

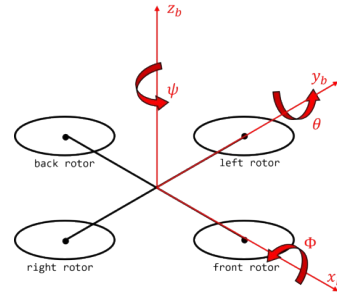


Figure 2 Body Coordinate System And The Euler Angles

Quadrotor UAV has four independent rotors which distributed in an "X" shape. It can adjust its posture via shifting velocity of each rotor. The kinematic and dynamic characteristics of the rotors are highly coupled with each others. The torque and lift force generated by rotation of each rotor of a quadrotor UAV are shown in Figure 3. The adjacent rotors are oriented in opposite directions to ensure that the total torque is zero.

General quadrotor UAVs possess following several states of motions:

Translation motions: forward and backward motion (move along the x axis); leftward and rightward motion (move along the y axis); upward and downward motion (move along the z axis).

Rotation motions: rolling motion (rotate along x axis); pitch motion (rotate along y axis); yaw motion (rotate along z axis).

Hovering (total lift force equals to the gravity).

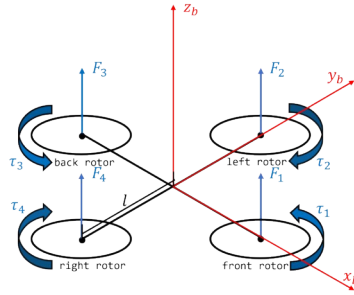


Figure 3 The Torque And Lift Force Generated by Rotation of Each Rotor of The UAV

2.1.2 Kinetic And Dynamic Model of Quadrotor UAV

In the absence of elastic vibration and deformation ^[5], it is clear to derive 12 equations of kinematics and dynamics of motions through classifying basic motions to translation and rotation:

$$\dot{x} = v_x \quad (1)$$

$$\dot{y} = v_y \quad (2)$$

$$\dot{z} = v_z \quad (3)$$

$$m \dot{v}_x = F_t (\cos\phi \cos\psi \sin\theta + \sin\phi \sin\psi) \quad (4)$$

$$m \dot{v}_y = F_t (\cos\phi \sin\theta \sin\psi - \sin\phi \cos\psi) \quad (5)$$

$$m \dot{v}_z = F_t \cos\phi \cos\theta - mg \quad (6)$$

$$\dot{\phi} = p + (r \cos\phi + q \sin\phi) \tan\theta \approx p \quad (7)$$

$$\dot{\theta} = q \cos\phi - r \sin\phi \approx q \quad (8)$$

$$\dot{\psi} = \frac{1}{\cos\theta} (r \cos\phi + q \sin\phi) \approx r \quad (9)$$

$$I_x \dot{p} = (I_y - I_z) q r + \tau_p \quad (10)$$

$$I_y \dot{q} = (I_z - I_x) p r + \tau_q \quad (11)$$

$$I_z \dot{r} = (I_x - I_y) p q + \tau_r \quad (12)$$

Note: F_t represents the total lift force, $F_t = F_1 + F_2 + F_3 + F_4$.

$F_i \propto f_{\text{air}} (i = 1, 2, 3, 4)$, where f_{air} is the air reaction.

p , q and r respectively represent angular velocities of Euler angles ϕ , θ and ψ .

I_x , I_y and I_z respectively represent rotational inertias of UAV along x, y and z axes.

τ_p is rolling moment, $\tau_p = F_2 l - F_4 l$.

τ_q is pitching moment, $\tau_q = F_3 l - F_1 l$.

τ_r is yawing moment, $\tau_r = (F_1 + F_3 - F_2 - F_4) d$. (d is the torque coefficient).

(1) - (6) are equations of linear motion, (7) - (12) are equations of angular motion. From these equations above, it is clear that quadrotor UAV is an extremely strong coupled system.

2.2 PID Control And Z-N Method

2.2.1 PID Control And Cascade PID Control

PID is a controller based on the proportion, integration and differential of errors, which is used widely in almost all UAV controlling [6]. By contrast, cascade PID control is more suitable to strongly coupled, nonlinear and multivariate system resembling quadrotor UAV. It is composed of two or more PID controllers in cascade and is divided into an outer ring and an inner ring [7].

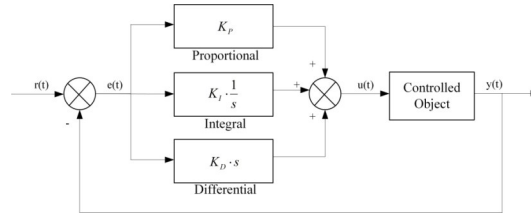


Figure 4 The Flow Frame Chart of PID control

The equation of PID control is shown:

$$u(t) = K_p \cdot e(t) + K_i \cdot \int_0^t e(\tau) d\tau + K_d \cdot \frac{de}{dt} \quad (13)$$

Note: $e(t)$ is the resulted error, $e(t) = r(t) - y(t)$. After the Laplace transform, (2.13) can be convert into (2.14): (the time domain is changed to the complex domain)

$$\begin{aligned} U(s) &= K_p \cdot E(s) + K_i \cdot \frac{1}{s} E(s) + K_d \cdot s \cdot E(s) \\ &= \left(K_p + \frac{K_i}{s} + K_d \cdot s \right) E(s) \end{aligned} \quad (14)$$

Note: K_p is proportional coefficient

K_i is integral coefficient, $K_i = \frac{K_p}{T_i}$, T_i is integration time.

K_d is differential coefficient, $K_d = K_p \cdot T_d$, T_d is differential time.

Generally, PID control can be divided into three links:

Proportional link(K_p): This part is to respond to the error signals of the control system timely. The greater the value of K_p , the faster the response of the system. Nevertheless, excessive proportional gain will result in excessive overshoot, which will decrease the stability of the system.

Integral link($K_i \cdot \frac{1}{s}$): This part is to eliminate the steady-state errors in order to improve indifference of the system. The shorter the T_i , the stronger the integral action effect. Nevertheless, the import of the section will result in reducing stability and deferring dynamic response of the system. Thus, integral link is usually combined with other links to form PI control or PID control.

Differential link($K_d \cdot s$): This part is to optimize the response speed and stability of the system. Different from integral link, the larger the T_d , the stronger the differential action effect. An appropriate value of K_d can effectively diminish overshoot and settling time. Differential link can be regarded as a high-pass filter so that it has a amplified action to noise interference as well [8]. Therefore, the role of differential links cannot be blindly strengthened.

Cascade PID control is a controlling structure to improve system control performance by using multiple PID controllers working in series, having outer loops and inner loops. Outer loop is to directly handle the final controlling target of the system. Its calculated error is inner loop's input. Inner loop is to response out loop's command quickly, focused on responsiveness. When adjusting parameters, debugging should be done loop by loop from inside to outside [9].

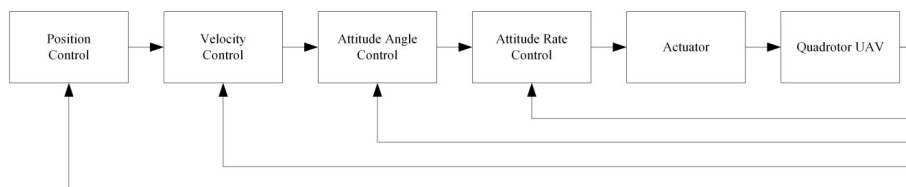


Figure 5 The Outer And Inner Loops of Quadrotor UAVs

As for the quadrotor UAV shown in Figure 5, position control is the first loop; velocity is the second loop; attitude angle is the third loop; attitude rate is the fourth loop. So, the first loop need to be adjusted is attitude rate control and the last is position control. Cascade PID can decompose complex problems into inner and outer loop tasks through hierarchical control strategy, taking into account response speed and stability, and is suitable for high-precision or anti-interference systems like quadrotor drones.

Table 1 Comparison of Ordinary PID Control And Cascade PID Control

	Ordinary PID Control	Cascade PID Control
Structure	Single Loop	Multi Loop
Parameters Adjusting	Simple	Complex
Anti-interference Capability	Weak	Strong
Applicable System	Univariate, Low Coupling	Multi-variable, Strong Coupling

2.2.2 Ziegler–Nichols Parameters Setting Method

Ziegler-Nichols method(Z-N method) is a frequently-used method of setting the control parameters of PID controller.^[10] shows the method proposed in the 1940s. Z-N tuning method is typically executed via a systematic procedure. Initially, K_I and derivative K_D gains are set to zero. The proportional gain K_p is then incrementally increased from zero until the ultimate gain K_u is reached, at which the control output exhibits sustained, constant-amplitude oscillations with a period referred to as the ultimate period P_u ^[11].

Table 2 Conclusions of Z-N Method

	K_p	K_I	K_D
P Control	$0.5K_u$	-	-
PI Control	$\frac{K_u}{2.2}$	$\frac{1.2K_p}{P_u}$	-
PID Control	$0.6K_u$	$\frac{2K_p}{P_u}$	$0.125K_p \cdot P_u$

Figure 6 shows a simulation in MATLAB/SIMULINK which need to manually adjust K_u until constant wavelength and amplitude oscillations occur, where $F(s)=(s^3+6s^2+5s)^{-1}$ is an arbitrary transfer function. When the constant wavelength and amplitude oscillations occur, $K_u=30$, $P_u=2.810$. The output scope is shown in Figure 7, where the yellow function image is a step signal and the blue one is the image of constant wavelength and amplitude oscillation.

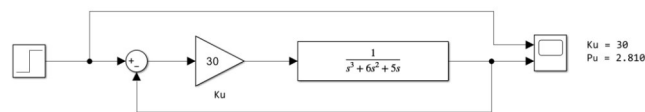


Figure 6 Searching for K_u and P_u of The Debugged $F(s)$ by MATLAB/SIMULINK

Figure 7 is a founded PID control used Z-N method by MATLAB/SIMULINK. In this PID controller system, the last row of data in Table 2 (when the control type is complete PID, the relationship between the three parameters K_p , K_I , K_D) is used. It is obvious that the system include Proportional, Integral and Differential three sections. The output scope of this PID control simulation is shown in Figure 9.

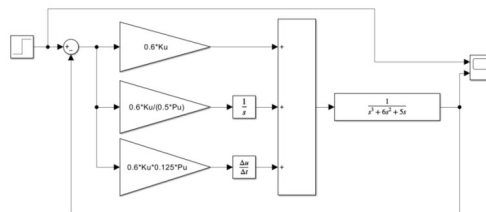


Figure 7 PID Control Simulation Utilized Z-N Method in MATLAB/SIMULINK

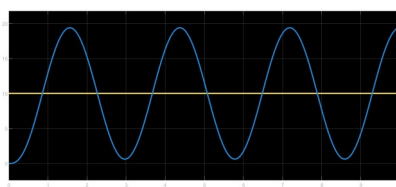


Figure 8 Output Scope of Figure 6

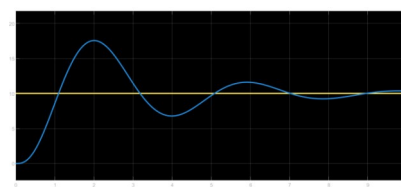


Figure 9 Output Scope of Figure 7

After the preliminary simulation of Z-N method on MATLAB/SIMULINK, it is not hard to find that the convergence of the simulated signal is not ideal. Therefore, utilizing another method to set parameters is extremely essential to the multi-variable and strong coupling system especially the quadrotor UAV. Optimization algorithm is a valid way to tackle the problem.

3 Particle Swarm Optimization Algorithm

3.1 Brief Introduction to PSO Algorithm

The particle swarm optimization (PSO) algorithm, first introduced by Kennedy and Eberhart in 1995, was initially developed to mimic the foraging behavior of birds^[12]. At its core, the algorithm involves randomly initializing a swarm of massless, volumeless particles—each representing a feasible solution to the target optimization problem. A pre-defined fitness function is used to assess the performance of each particle as it traverses the feasible solution space, with its movement direction and step size governed by a velocity parameter. The PSO algorithm converges on the global optimal solution through iterative searching, where each particle dynamically adjusts its trajectory by tracking two key benchmarks in every iteration: its own historical best performance (p_i^t) and the optimal solution achieved by the entire swarm (p_g^t)^[13].

Suppose a population of M particles traveling at a certain speed in D -dimensional space. The state properties of particle i at time t are set as follows:

$$x_i^t = [x_{i1}^t, x_{i2}^t, \dots, x_{id}^t]^T \quad (15)$$

$$v_i^t = [v_{i1}^t, v_{i2}^t, \dots, v_{id}^t]^T \quad (16)$$

$$p_i^t = [p_{i1}^t, p_{i2}^t, \dots, p_{id}^t]^T \quad (17)$$

$$p_g^t = [p_{g1}^t, p_{g2}^t, \dots, p_{gd}^t]^T \quad (18)$$

Note: $1 \leq d \leq D, 1 \leq i \leq M$.

$x_{id}^t \in [L_d, U_d]$, L_d and U_d are the lower and upper limits of the search space respectively.

$v_{id}^t \in [v_{min,d}, v_{max,d}]$, $v_{min,d}$ and $v_{max,d}$ are the minimum and maximum velocities respectively.

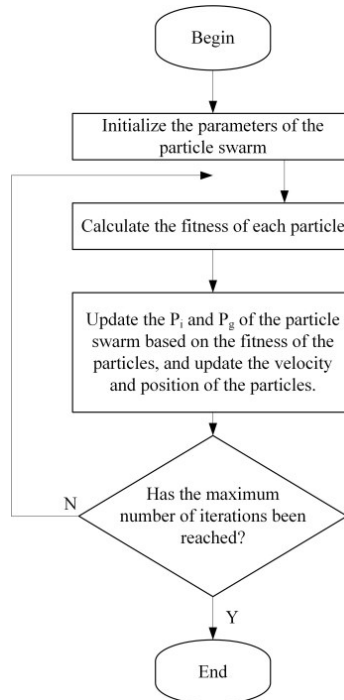


Figure 10 Flowchart of Particle Swarm Optimization Algorithm

The updating formula of positions and velocities of particles at time $t + 1$ is as follow:

$$v_{id}^{t+1} = v_{id}^t + c_1 r_1 (p_{id}^t - x_{id}^t) + c_2 r_2 (p_{gd}^t - x_{id}^t) \quad (19)$$

$$x_{id}^{t+1} = x_{id}^t + v_{id}^{t+1} \quad (20)$$

Note: r_1 and r_2 are two random numbers which are uniformly distributed in $[0, 1]$.

c_1 and c_2 are called learning factors, commonly existing: $c_1 = c_2 = 2$ ^[14].

3.2 Standard PSO Algorithm

In 1998, Eberhart and Shi brought inertia weight ω into equation (19) and (20) as shown in (21) and (22), named

standard particle swarm optimization algorithm ^[15]:

$$v_{id}^{t+1} = \omega v_{id}^t + c_1 r_1 (p_{id}^t - x_{id}^t) + c_2 r_2 (p_{gd}^t - x_{id}^t) \quad (21)$$

$$x_{id}^{t+1} = x_{id}^t + v_{id}^{t+1} \quad (22)$$

$$\omega = \omega_{max} - \frac{(\omega_{max} - \omega_{min}) \times T}{T_{max}} \quad (23)$$

Note: T is the times of iterations and T_{max} is the maximum number of iterations.

The standard PSO algorithm is utilized more commonly in scientific studies currently. Confirming the value range of ω is a high-profile scholar problem.

4 PID Control Based on Particle Swarm Optimization Algorithm

4.1 Preliminary Selection of the Fitness Function

Designing the fitness function is one of the most crucial steps in the PSO algorithm, as it directly determines whether the algorithm can find the optimal solution to the problem. The commonly used comprehensive performance evaluation criteria on PID adjusting are mainly based on the relationship between the system's error $e(t)$ and time t , owning following forms ^[16]:

(1) Integral Square Error (ISE):

$$\int_0^{\infty} e(t)^2 dt \quad (24)$$

(2) Integral Time Square Error (ITSE):

$$\int_0^{\infty} t e(t)^2 dt \quad (25)$$

(3) Integral Absolute Error (IAE):

$$\int_0^{\infty} |e(t)| dt \quad (26)$$

(4) Integral Time Absolute Error (ITAE):

$$\int_0^{\infty} t |e(t)| dt \quad (27)$$

In the process of controlling, the less these integral values, the better the function of adjusting the parameters of PID. After comparing, the PID controller with ITAE owns higher response speed ^[17]. So ITAE is chosen to comprise the fitness function.

To ensure the high velocity of response and limit the overshoot, introduce two auxiliary function:

$$RiseTime = \begin{cases} t_{90\%} - t_{10\%} & t_{90\%} \text{ and } t_{10\%} \text{ exist} \\ \infty & \text{otherwise} \end{cases} \quad (28)$$

$$Overshoot = \begin{cases} 100 \max \left(0, \frac{y_p - y_{ss}}{y_{ss}} \right) & t_p < t_{90\%} \\ 0 & \text{otherwise} \end{cases} \quad (29)$$

Note: $t_{10\%}$ and $t_{90\%}$ are the time when the system response first reaches the final value of 10% and 90% respectively. y_p and y_{ss} are the peak value and stable state value of the system response respectively.

Thus, in this essay, preliminarily designing the fitness function as (30):

$$f(t) = \frac{1}{0.6ITAE + 0.2RiseTime + 0.2|Overshoot| + 0.01} \quad (30)$$

Note: The reason to add 0.01 is to avoid the denominator equalling to zero.

4.2 Convergence Testing and Stability Testing

The convergence is the most necessary step in this loop. If the output y has no nonnumeric, infinite value or values exceeding 10^5 , the response is convergent:

$$Convergence = \begin{cases} True & \text{no nonnumeric, infinite value or values exceeding } 10^5 \\ False & \text{otherwise} \end{cases} \quad (31)$$

In order to guarantee the stability of system, avoid the particles focusing on locally optimal solution and eliminate the situation of aggregation on the boundary, giving priority to ensuring stability of system before optimizing is paramount. Therefore, the overshoot, settling time t_s and steady state error e_{ss} are the factors need to be limited:

$$t_s = \min \{ t \mid \forall \tau \geq t, |y(\tau) - y_{ss}| \leq 0.02 \} \quad (32)$$

$$e_{ss} = |r - y_{ss}| \quad (33)$$

$$Stability = \begin{cases} True & \text{Overshoot} \leq \text{threshold}, t_s \leq \text{threshold}, e_{ss} \leq \text{threshold} \\ False & \text{otherwise} \end{cases} \quad (34)$$

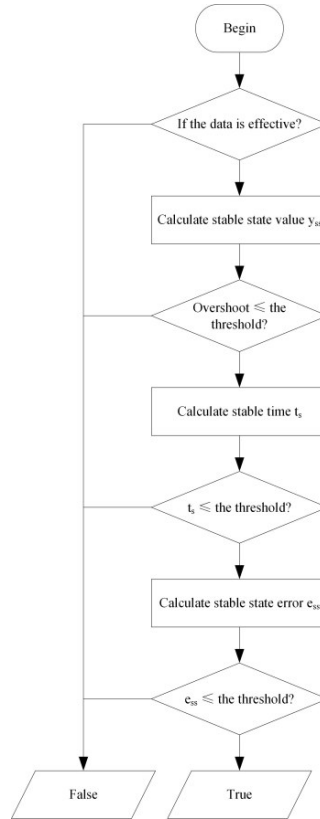


Figure 11 Flowchart of Stability Testing

The process of stability testing is a multi-objective optimization problem. The objective of multi-objective optimization is to find the Pareto optimal solution of the problem under consideration. In comparison to tradition method, the optimizing direction Pareto optimal method confronts the whole Pareto front rather than the single path.

After convergence testing and stability testing, the fitness function can be finally defined:

$$f(t) = \begin{cases} -10^6 & \text{Convergence} = \text{False} \\ -10^4 & \text{Stability} = \text{False} \\ (0.6ITAE + 0.2RiseTime + 0.2|Overshoot| + 0.01)^{-1} & \text{otherwise} \end{cases} \quad (35)$$

4.3 Particle Boundary Rebound Rule

The particle boundary rebound effect can augment robustness of the system. The main principle is to simulate the elastic collision behavior of an actual physical system. Compared with direct truncation on boundaries, the particles will not gather on the boundaries, which extremely increases the diversity of particle swarm. So, the standard PSO formulas can be upgraded to the following:

$$x_{id}^{t+1} = \begin{cases} L_d - (x_{id}^t - L_d) = 2L_d - x_{id}^t & x_{id}^t < L_d \\ U_d + (U_d - x_{id}^t) = 2U_d - x_{id}^t & x_{id}^t > U_d \\ x_{id}^t + v_{id}^{t+1} & \text{otherwise} \end{cases} \quad (36)$$

$$v_{id}^{t+1} = \begin{cases} -0.7v_{id}^t & x_{id}^t < L_d \text{ or } x_{id}^t > U_d \\ \omega v_{id}^t + c_1 r_1 (p_{id}^t - x_{id}^t) + c_2 r_2 (p_{gd}^t - x_{id}^t) & \text{otherwise} \end{cases} \quad (37)$$

$$\omega = \omega_{max} - \frac{(\omega_{max} - \omega_{min}) \times T}{T_{max}} \quad (38)$$

This paper employs formulas (36) - (38) as the improved PSO algorithm to optimize PID controller.

4.4 Simulation Experiment Result

Utilizing MATLAB to code the final PSO algorithm program and linking to the simulated PID model in SIMULINK. In the program, setting the thresholds as followed:

- (1) Overshoot threshold = 150%
- (2) Steady state error $e_{ss} = 0.02$
- (3) Settling time $t_s = 10$

In the SIMULINK model, randomly setting the transfer function as: $\frac{1}{s^2 + 6s + 5}$. After 50 iterations, eventually deriving three parameters: $K_p = 46.8103$, $K_i = 5.1974$, $K_d = 49.9677$. The PID control simulation is shown in Figure 4.2 and the output scope is shown in Figure 4.3:

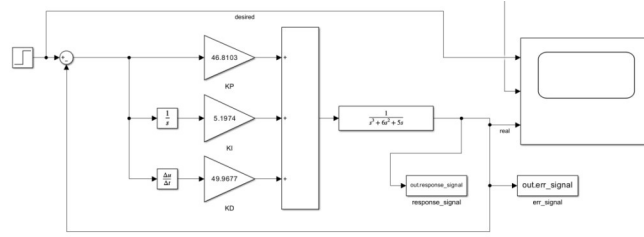


Figure 12 PID Control Simulation Based on PSO Algorithm in MATLAB/SIMULINK

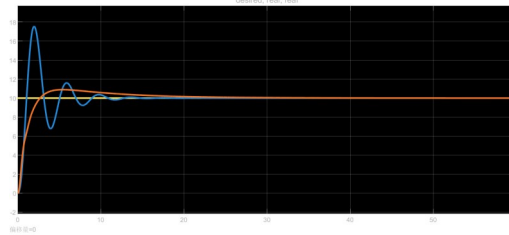


Figure 13 Output Scope of Figure 12

The blue curve is the Z-N method line and the orange curve represents PSO line. According to comparing with the line utilized traditional Z-N method, PSO algorithm is much better, owing to less overshoot, faster settling time and inappreciable stable state error.

5 Application of Optimized Algorithm in Quadrotor UAV System

5.1 Determination of the UAV Control System

Choose the classic type of UAV, DJI Spark as the testing object ^[18-19]. The technical parameters are shown in Table 3.

Table 3 Parameter summary table of DJI Spark

parameters	values
mass m	0.3kg
distance between shafts l	0.35m
g	9.8m/s ²
I_x	0.004593kg · m ²
I_y	0.004593kg · m ²
I_z	0.009188kg · m ²
tension coefficient c_T	$1.4865 \times 10^{-7} \text{N} \cdot \text{s}^2/\text{rad}^2$
moment coefficient c_M	$2.925 \times 10^{-9} \text{N} \cdot \text{m} \cdot \text{s}^2/\text{rad}^2$
electromechanical time constant T_m	0.11ms

Build the UAV control system in SIMULINK as shown as Figure 14

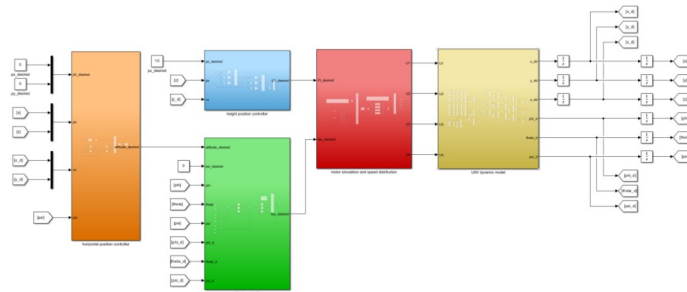


Figure 14 The Control Model of the DJI Spark Drone in MATLAB/SIMULINK

The model includes three parts of controller. The blue module is height position controller, the orange module is horizontal position controller and the green module is attitude controller. These controllers are all utilized cascaded PID controller in order to augment the anti-interference capability, respectively shown in Figure 15, Figure 16 and Figure 17.

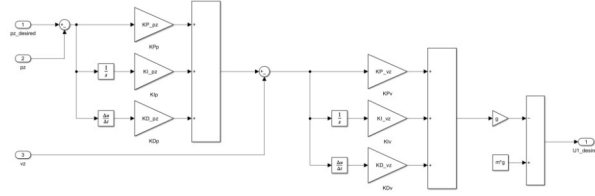


Figure 15 Height Position Controller

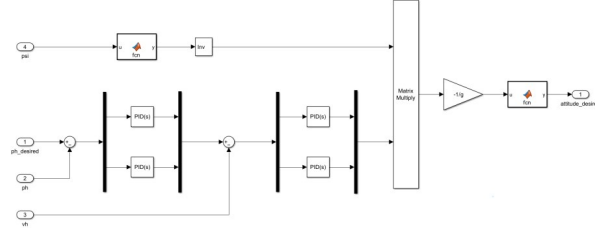


Figure 16 Horizontal Position Controller

In order to prevent excessive complexity, the horizontal position controller employs PID controller provided by MATLAB/SIMULINK. The output of the first MATLAB function in the controller represents $\begin{bmatrix} \sin\psi & \cos\psi \\ -\cos\psi & \sin\psi \end{bmatrix}$. The second MATLAB function is used to make the result value greater than 0.2 equals to 0.2 and the value smaller than -0.2 equals to -0.2. Thus, the desired attitude value can be limited in a more controllable range.

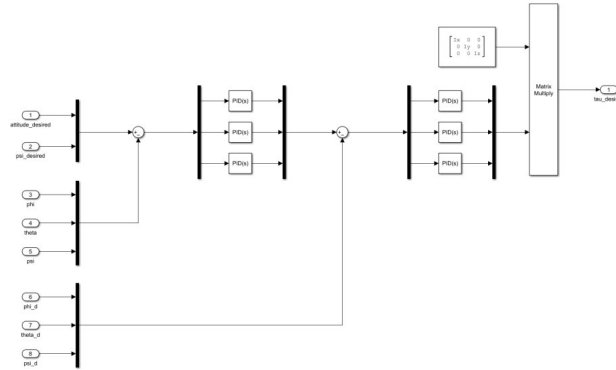


Figure 17 Attitude Controller

In order to prevent excessive complexity, the attitude controller employs PID controller provided by MATLAB/SIMULINK.

Therefore, according to the above three controllers, there are 6 parameters need to be adjusted in height position controller, 12 parameters in horizontal position controller and 18 parameters in attitude controller.

The red module called motor simulation and speed distribution is to simulate the dynamic responses of four motors of drone and distribute their rotate speed into four parameters: total lift force F_t , rolling moment τ_r , pitching moment τ_q and yawing moment τ_r . To control them directly in the UAV model, rewrite these four variables as U_1 , U_2 , U_3 and U_4 respectively.

The relationship between the total lift force and three moments is shown as follow:

$$\begin{bmatrix} U_1 \\ U_2 \\ U_3 \\ U_4 \end{bmatrix} = \begin{bmatrix} c_T & c_T & c_T & c_T \\ \frac{\sqrt{2}}{2} l_{c_T} & -\frac{\sqrt{2}}{2} l_{c_T} & -\frac{\sqrt{2}}{2} l_{c_T} & \frac{\sqrt{2}}{2} l_{c_T} \\ \frac{\sqrt{2}}{2} l_{c_T} & \frac{\sqrt{2}}{2} l_{c_T} & -\frac{\sqrt{2}}{2} l_{c_T} & -\frac{\sqrt{2}}{2} l_{c_T} \\ c_M & -c_M & c_M & -c_M \end{bmatrix} \begin{bmatrix} \omega_1^2 \\ \omega_2^2 \\ \omega_3^2 \\ \omega_4^2 \end{bmatrix} \quad (39)$$

Note: ω represents the rotate speed of one motor.

c_T and c_M are two constants which respectively refer to tension coefficient and moment coefficient.

The dynamic response equation of the motor is based on the following first-order low-pass filtering model ^[20]:

$$\omega_{ss} = \frac{1}{T_m s + 1} \omega \quad (40)$$

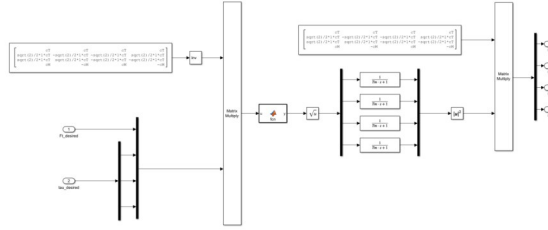


Figure 18 Speed Distribution and Motor Simulation

According to equation(1) - (12), construct the yellow module called UAV dynamic model, shown as following in Figure 19:

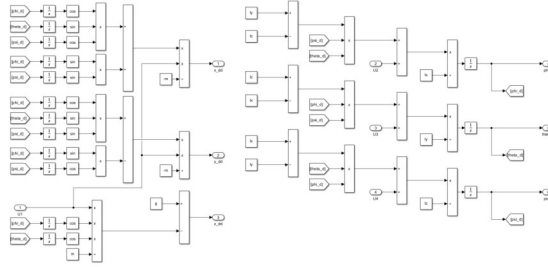


Figure 19 UAV Dynamic Model

Taking one with another of the UAV control model, it needs four desired parameters: the desired position on axis x , y , and z and the desired yawing angle ψ .

5.2 Optimizing PID Parameters in UAV Controllers Using PSO Algorithm

Using the PSO algorithm in the controllers, derive the optimized PID parameters shown in Table 4. And in the simulation model, the desired values of x , y , z and ψ are preset as 5m, 5m, 10m and 0 respectively. The six curves of three positions x , y and z , and three attitude angles ϕ , θ and ψ are shown in following:

The optimized PID parameters obtained by the PSO algorithm are presented in Table 4, and the corresponding

Table 4 PID Optimization Parameter Table

Control Channels	Parameter Type	Parameters	Value
Horizontal Position Controller	Outer Loop(Position Loop)	$K_{P_{ox}}$	12.1963
		$K_{I_{ox}}$	17.0957
		$K_{D_{ox}}$	0.0000
	Inner Loop(Velocity Loop)	$K_{P_{ix}}$	5.5509
		$K_{I_{ix}}$	0.0000
		$K_{D_{ix}}$	4.0948
Horizontal Position Controller	Outer Loop(Position Loop)	$K_{P_{oy}}$	12.1963
		$K_{I_{oy}}$	0.0000
		$K_{D_{oy}}$	17.0957
	Inner Loop(Velocity Loop)	$K_{P_{iy}}$	5.5509
		$K_{I_{iy}}$	0.0000
		$K_{D_{iy}}$	4.0948
Height Position Controller	Outer Loop(Position Loop)	$K_{P_{oz}}$	49.2589
		$K_{I_{oz}}$	0.0000
		$K_{D_{oz}}$	41.1652
	Inner Loop(Velocity Loop)	$K_{P_{iz}}$	9.5197
		$K_{I_{iz}}$	28.5752
		$K_{D_{iz}}$	6.1429
Attitude Controller	Outer Loop(Attitude Angle Loop)	$K_{P_{o\phi}}$	4.9440
		$K_{I_{o\phi}}$	0.0000
		$K_{D_{o\phi}}$	1.4085
	Inner Loop(Attitude Rate Loop)	$K_{P_{i\phi}}$	2.4792
		$K_{I_{i\phi}}$	0.0126
		$K_{D_{i\phi}}$	0.1579
Attitude Controller	Outer Loop(Attitude Angle Loop)	$K_{P_{o\theta}}$	4.9440
		$K_{I_{o\theta}}$	0.0000
		$K_{D_{o\theta}}$	1.4085
	Inner Loop(Attitude Rate Loop)	$K_{P_{i\theta}}$	2.4792
		$K_{I_{i\theta}}$	0.0126
		$K_{D_{i\theta}}$	0.1579
Attitude Controller	Outer Loop(Attitude Angle Loop)	$K_{P_{o\psi}}$	4.8382
		$K_{I_{o\psi}}$	0.0002
		$K_{D_{o\psi}}$	0.3952
	Inner Loop(Attitude Rate Loop)	$K_{P_{i\psi}}$	28.9762
		$K_{I_{i\psi}}$	0.0003
		$K_{D_{i\psi}}$	0.4746

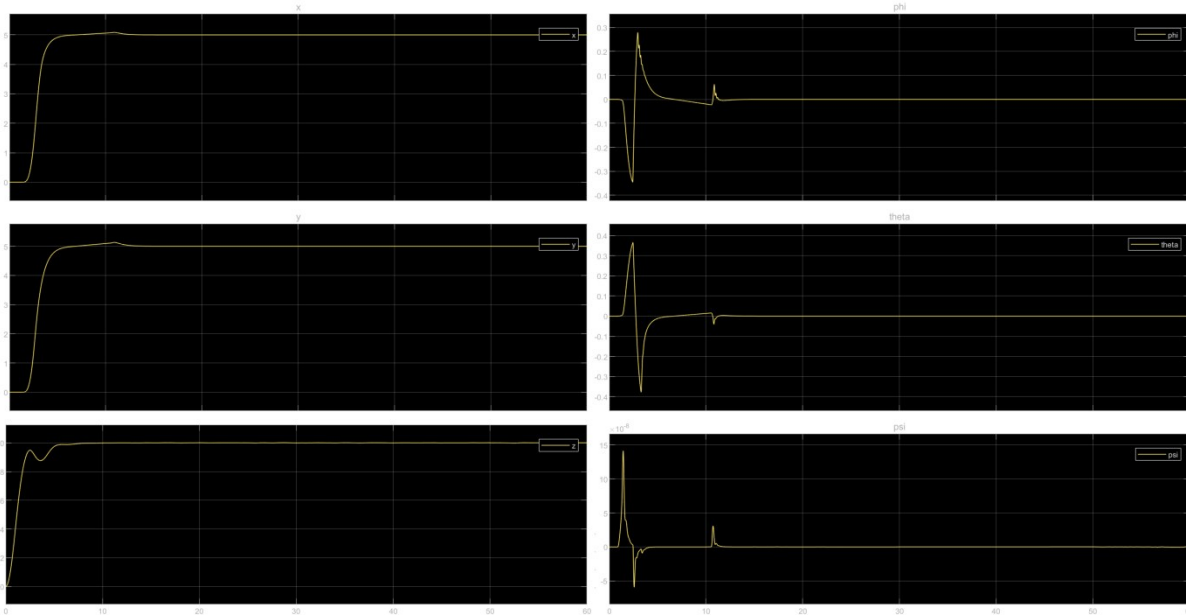


Figure 20 Parametric Curves of Position and Attitude Angle

dynamic response is shown in Figure 5.7. The simulation results show that the actual position of the DJI Spark quadrotor can quickly track the preset values of $x=5\text{m}$, $y=5\text{m}$, $z=10\text{m}$, and the yaw angle ψ stably converges to 0 without obvious overshoot, oscillation, or steady-state error. The roll and pitch angles exhibit only a small transient response at the initial stage and quickly return to stability, which is consistent with the strongly coupled dynamic characteristics of the quadrotor UAV.

These results fully verify that the improved PSO algorithm can effectively search for the optimal parameter combination for the cascaded PID controller, significantly improving the dynamic and static performance of the control system and overcoming the limitations of traditional manual tuning methods for nonlinear, strongly coupled systems.

6 Summary And Prospect

6.1 Summary

This paper focuses on the parameter tuning problem of cascaded PID control for quadrotor UAVs, which are characterized by nonlinearity, strong coupling, and underactuation. Taking the DJI Spark as the research object, the main contributions are as follows:

- (1) An improved PSO algorithm is proposed, which uses the ITAE criterion as the fitness function and introduces boundary rebound rules to avoid local optima, significantly improving the parameter tuning efficiency and accuracy.
- (2) A cascaded PID control system is built in MATLAB/Simulink, and the improved PSO algorithm is applied to tune the parameters of the height, horizontal position, and attitude controllers. Simulation results show that the optimized system exhibits excellent trajectory tracking and stability, validating the feasibility and superiority of the proposed method.

6.2 Prospect

Although the proposed method achieves good simulation results, there are still limitations that need to be addressed in future work:

- (1) Hardware-in-the-loop and real-flight verification: The current research is based on ideal simulation conditions. Future work will involve building a hardware-in-the-loop platform and conducting real-flight tests to validate and refine the control parameters under actual environmental conditions.
- (2) Robustness enhancement: The current model does not fully consider external disturbances such as wind fields and internal uncertainties such as mass changes. Future research will focus on improving the system's robustness and adaptability to these factors.
- (3) Algorithm fusion and function expansion: The proposed method can be combined with other advanced control strategies (e.g., model predictive control, sliding mode control) and extended to integrate autonomous flight functions such as path planning and obstacle avoidance, thereby expanding its application scope in complex environments.

Overall, the cascaded PID control method based on the improved PSO algorithm shows great potential in quadrotor UAV control. With further research and optimization, it is expected to play an increasingly important role in practical engineering applications.

References

- [1] Mahony, R. E., Kumar, V., and Corke, P. (2012) "Multirotor Aerial Vehicles: Modeling, Estimation, and Control of Quadrotor." Journal Article. *IEEE Robot. Automat. Mag.* 19 (3): 20 – 32. https://kns.cnki.net/kcms2/article/abstract?v=AGqaFAGiW58KYF_hEaF5EhadRWKBtiSDBHrLKKG_gYTQRmVpeFDI29hJ7PKElVRzQpOy-bFYebUhBzscX0Fj3SHwiAtrCuV1cimlwZHjMklAyg0FRdtJPYRCh0yQ7ZOUrKmZGenGQI5EkwlGfNxPf2xN1qZJV08OoUi3LEl1GdIllyt6Q4HaL-8pHIQha9LPsvEeypywJ9Cmo6tRgLFUHKmhAC4_Rd&uniplatform=NZKPT&language=CHS.
- [2] Ye, Z., L. Xu, L. Huang, Y. Zhao, M. Xi, and Z. He. (2025) "Quadratic Optimization of Fractional-Order PID for Attitude Control of Quadrotor Unmanned Aerial Vehicles." Journal Article. *Journal of Hubei Institute of Automobile Industry*, 1 – 5. <https://link.cnki.net/urlid/42.1448.TH.20250103.1433.004>.
- [3] Tan, G. (2013) "Design and Implementation of Attitude Control System for a Quadrotor UAV." Thesis. https://kns.cnki.net/kcms2/article/abstract?v=JtACmXrF272HIW_sXmQAeo5ULXumBBZr3AV06KGzT1NmQinguB7AtVv0O-YvQfH9hzaT8Hp4lGBF5qSIKW6VNTtWlehbWEM3oQKhUYLmPbtE0A50tQ6X_vTqFtepercqvlLAeifdyvIzv8a7xL1HJzTgdQpUUjIJBTKmlrfr1PwDipcdC8VFh_zi2iFZlqX7tO6o5ygaA=&uniplatform=NZKPT&language=CHS.
- [4] Cheng, S., Shi, Y, Qin, Q., and Ting, T. O. (2012) "Population Diversity Based Inertia Weight Adaptation in Particle Swarm Optimization." Journal Article. *2012 IEEE Fifth International Conference on Advanced Computational Intelligence (ICACI)*, 395–403.
- [5] Huang, X. (2010) "Design of a Four-Rotor Unmanned Helicopter Flight Controller." Thesis. https://kns.cnki.net/kcms2/article/abstract?v=JtACmXrF272eVLFwXH76m64gFVSEA62Sa5Ekk819AsSVrZ9NxmApodND1tbSI7ml4diz7fgAM9BAfPbypqpMiq5zHZU-hBj58fpYb5ickUtNTcVu5dvMIZKmOxbvAhGVelAo_WmLbatppBhGDhaCGgPDkCjXT1NKhkUiZYHCXp3BEY9jaWXVMP5LLgS6qwgETpI9vt0J-Q=&uniplatform=NZKPT&language=CHS.
- [6] Zhao, J., Z. Chen, and R. Zhang. (2021) "Multi-Rotor Aircraft Control Based on Cascade PID for Precisely Hovering over Fixed Point." Journal Article. *Electronics Optics And Control* 28 (08): 17–20+30. <https://link.cnki.net/urlid/41.1227.TN.20210414.1734.020>.
- [7] Han, J. Q. (2009) "From PID to Active Disturbance Rejection Control." Journal Article. *IEEE Transactions on Industrial Electronics* 56 (3): 900–906.
- [8] Jia, Y. (2015) *Research on Control Algorithm for Quadrotor Aircraft*. Journal Article. https://kns.cnki.net/kcms2/article/abstract?v=JtACmXrF271JwL0phP1Mg197YpXsu6CzNQ0UqWGrxR-sWfKC6Z6rIBgNqej8B4tgh05B3N2v46rNH10x5pJ6YffZaKX013ROInF_qFOF9m_GmFwBpgCPQRCcLrj2CZbSn-bNlSkSuomwNAoPMSmYt0xdtdXH5ivQVPJz-ZCrAUoz8EnnbWR-bZQgd9w63Rw0Fg8qzQDtg=&uniplatform=NZKPT&language=CHS.
- [9] Behera, A., T. K. Panigrahi, P. K. Ray, and A. K. Sahoo. (2019) "A Novel Cascaded PID Controller for Automatic Generation Control Analysis with Renewable Sources." Journal Article. *IEEE-Caa Journal of Automatica Sinica* 6 (6): 1438–51.
- [10] Ziegler, J. G., and N. B. Nichols. (1993) "Optimum Setting for Automatic Controllers." Journal Article. *Journal of Dynamic Systems Measurement and Control*, 759–68.
- [11] Valerio, Duarte, and Jose Sa da Costa. (2006) "Tuning of Fractional PID Controllers with Ziegler-Nichols-Type Rules." Journal Article. *Signal Processing: The Official Publication of the European Association for Signal Processing (EURASIP)* 86 (10): 2771–84. <https://d.wanfangdata.com.cn/periodical/Ch9QZXJpb2RpbY2FuSU5HTmV3UzlwMjQwOTewMTY1MjU1EiBmMGZmNDJfYVWmMzI0YzVmMjUwYjhlODYxN2ExOTZkYholMjRkNjIyeHU=>.
- [12] Lu, Y., Guo, J., and Li, Z. (2022) "Rotor UAV PID Parameter Tuning Algorithm Based on Improved Particle Swarm Optimization." Journal Article. *Journal of Detection and Control* 44 (05): 116 – 24. https://kns.cnki.net/kcms2/article/abstract?v=JtACmXrF273GfJg7SseaVgZL-awspmvK3cBxqEx_3EVIZs9JL64iVJ6qi1vbfGgK1TKAOt5OirOC6jR2XOnWjNjCD0Pty6jq0RXoP3il-lpG2F5WGHsk0ws5z4e-HOo_LWgkZ2da05VQvHtQXfAhcKO2SJoLGk_fkDfe_uFX-P8xaPmk_vCQXMNaG2TVuSJLjQlaW0C1Vw=&uniplatform=NZKPT&language=CHS.
- [13] Valle, Y. del, Venayagamoorthy, G. K., Mohagheghi, S., Hernandez, J. C., and Harley, R. G. (2008) "Particle Swarm Optimization: Basic Concepts, Variants and Applications in Power Systems." Journal Article. *IEEE Transactions on Evolutionary Computation* 12 (2): 171–95.
- [14] Bratton, D., and Kennedy, J. (2007) "Defining a Standard for Particle Swarm Optimization." Journal Article. *2007 IEEE Swarm Intelligence Symposium*, 120–27.
- [15] Shi, Y., and Eberhart, R. (1998) "A Modified Particle Swarm Optimizer." Journal Article. *1998 IEEE International Conference on Evolutionary Computation Proceedings. IEEE World Congress on Computational Intelligence (Cat. No.98TH8360)*, 69–73.
- [16] Liu, D. (2012) "Research on PID Parameter Tuning Based on Improved Particle Swarm Optimization Algorithm." Thesis. https://kns.cnki.net/kcms2/article/abstract?v=AGqaFAGiW59z0FVB2MV74begYmOVUfxQGYakeC6BJQ2OIW-h--PJ9M2MuKFlow2tN_kumdiT2DiYwD6ICyqfAsesClwkdmm5AW8TviIFCHMCzd_kXhL5TTgTxC0o-x0dUoXDwg6rFRfZv6CgIro-lk6sdPMNhgZp-utuZ0j4uawlmfS3vsXaAF5obBEQ-hA0_s6-3r720k=&uniplatform=NZKPT&language=CHS.
- [17] Argentim, L. M., Rezende, W. C., Santos, P. E., and Aguiar, R. A.. (2013) "PID, LQR and LQR-PID on a Quadcopter Platform." Journal Article. *2013 International Conference on Informatics, Electronics and Vision (ICIEV)*, 1–6.
- [18] Geng, Z., Meng, Q., Zhao, Y., and Han, H. (n.d.) "PID Control Model of Quad-Rotor Unmanned Aircraft Based Simulink." Conference Proceedings. *The 9th China Command and Control Conference*, 6.
- [19] DJI Innovation Technology Co., Ltd. (n.d.) "Spark Technical Support." [Webpage], n.d. <https://www.dji.com/support/product/spark>.
- [20] Chehadeh, M. S., and Boiko, I. (2019) "Design of Rules for in-Flight Non-Parametric Tuning of PID Controllers for Unmanned Aerial Vehicles." Journal Article. *Journal of the Franklin Institute-Engineering and Applied Mathematics* 356 (1): 474–91.