ITC 1/55
Information Technology and Control
Vol. 55 / No. 1/ 2026
pp. 257-279
DOI 10.5755/j01.itc.55.1.43501

Obstacle Avoidance Path Planning and Robust Tracking Control of Quadrotor UAVs under Uncertain Disturbances

Received 2025/11/03 | Accepted after revision 2026/01/19

HOW TO CITE: Yang, K., Zhu, S., Ren, J., Yang, Z. (2026). Obstacle Avoidance Path Planning and Robust Tracking Control of Quadrotor UAVs under Uncertain Disturbances. *Information Technology and Control*, 55(1), 257-279. https://doi.org/10.5755/j01.itc.55.1.43501

Obstacle Avoidance Path Planning and Robust Tracking Control of Quadrotor UAVs under Uncertain Disturbances

Kaiwen Yang, Shihong Zhu, Jiahui Ren, Zhenchao Yang*
International Engineering College, Xi'an University of Technology, Xi'an, 710048, China

Corresponding author: zcyang@xaut.edu.cn

In this paper, obstacle-avoidance path planning and robust trajectory tracking are addressed for quadrotor UAVs in the presence of model uncertainties and pulse disturbances. By using the Newton–Euler formulation, a 6-DOF nonlinear dynamic model is derived to construct a constrained 3D path-planning problem. To overcome the shortcomings of conventional PSO in terms of premature convergence and poor path smoothness, a hybrid adaptive multi-topology PSO with intelligent strategy learning and cubic B-spline smoothing, termed as HAMT-AISL-PSO, is proposed, which jointly enhances the global search ability and path executability. In the aspect of tracking control, a linear active disturbance rejection control (LADRC) scheme with a linear extended state observer is designed to estimate and compensate the total disturbances in real time. Simulation results on a cluttered 3D environment demonstrate that, compared with the standard PSO, the average cost function and average path length are reduced by 25.20% and 20.45%, respectively. Under a 3D spiral reference and pulse torque disturbances, LADRC achieves faster transient response and near-zero steady-state error with much smaller tracking deviations than PID, hence verifying the effectiveness of the proposed planning-and-control framework.

KEYWORDS: Quadrotor unmanned aerial vehicles, Six-degree-of-freedom dynamic model, HAMT-AISL-PSO algorithm, Linear active disturbance rejection control.

1. Introduction

1.1. Research Background and Significance

In recent years, due to the low cost, strong mobility, and simple structure, quadrotor unmanned aerial vehicles (UAVs) have witnessed a continuous upward trend in application demands in numerous fields, including military and civilian sectors [11]. In search

and rescue missions, UAVs can quickly reach areas that pose significant challenges for human access, such as earthquake-stricken ruins filled with unstable structures and hazardous debris, forest fire scenes with intense heat, thick smoke, and rapidly spreading flames [17], and mountain avalanche events where the terrain is covered with snow-laden debris and prone to secondary disasters. Through the high definition cameras and thermal imaging devices they carry, they can provide timely situational details from affected zones, providing crucial evidence for rescue command and facilitating the implementation of precise rescue operations. For environmental monitoring, UAVs traverse varied terrains and stream real-time indicators — air and water quality as well as vegetation coverage — to ground stations, providing data support for environmental protection and ecological research [30]. In the field of logistics and distribution, UAVs are expected to achieve efficient "last-mile" delivery. Especially in traffic-congested urban areas or remote mountainous regions, they can break through traditional transportation limitations, quickly and accurately deliver goods to users, improving the efficiency of logistics and distribution [14].

However, with the continuous expansion of UAV application scenarios into complex environments, the challenges they encounter are growing increasingly formidable. In such complex environments, the distribution of obstacles is highly intricate. Static obstacles encompass elements such as high-rise buildings, trees, and rocks, while dynamic obstacles include pedestrians and vehicles. It poses extremely high requirements for the path planning of UAVs. Existing path planning methods, such as graph-based algorithms, are prone to getting trapped in the dilemma of local optimal solutions when dealing with complex environments [18]. These methods usually search for paths within a preset search space based on specific search strategies. When the environment contains a relatively complex obstacle distribution, the search process may converge prematurely to a path that seems optimal but is not globally optimal, causing the UAV to face unnecessary risks during actual flight and making it difficult to effectively respond to sudden environmental changes, such as obstacles suddenly appearing or their positions being altered.

At the same time, UAVs are affected by various external disturbances during flight, such as airflow changes and wind forces, and their own systems also have uncertainties, including errors in model parameters and the non-linear characteristics of actuators. Traditional tracking control methods, such as the classic PID control, are mainly designed based on linearized models and have poor adaptability to external disturbances and system uncertainties, lacking sufficient robustness. When subjected to external disturbances, it is difficult to maintain a stable flight attitude and accurate trajectory tracking, which may lead to the failure of the flight mission or even the crash of the UAV [6]. Therefore, studying path-planning and robust tracking control methods for UAVs suitable for complex environments has important practical significance.

1.2. Research Status and Problems

Currently, various algorithms are available in UAV path planning research. As a representative graph-based path-planning method, the A* algorithm directs the search process by means of a heuristic function, thereby enhancing the search efficiency to a discernible extent [27]. However, within intricate and obstacle-laden environments, the search space of the A* algorithm undergoes a rapid expansion. This phenomenon causes an exponential upsurge in computational complexity and a pronounced deceleration in search velocity, ultimately rendering it hard to satisfy the exacting real-time demands. By means of random sampling, the Rapidly-exploring Random Tree (RRT) algorithm builds a search tree and can quickly explore unknown environments, having certain advantages in dealing with complex environments. Nevertheless, the paths generated by this algorithm are often tortuous and not smooth enough, which is not conducive to the efficient flight of the UAV [12]. Moreover, its randomness means that the paths generated each time may vary greatly, and the stability is not satisfactory. Inspired by the foraging behavior of bird flocks, the Particle Swarm Optimization (PSO) algorithm conducts optimization searches and shows clear advantages in handling general optimization problems. However, when applied to path planning, it tends to fall into local optima and exhibits limited environmental adaptability, which restricts its performance in complex and dynamic obstacle scenarios.

Among tracking control approaches, the classic Proportional-Integral-Derivative (PID) controller is

widely applied because of its simple structure and ease of implementation [3]. However, it is strongly dependent on the model. When there is uncertainty in the system or it is affected by external disturbances, the control performance will decline significantly. The Linear Quadratic Regulator (LQR) designs the controller by minimizing the quadratic performance index and can obtain good control effects when dealing with linear systems [2]. Nevertheless, a quadrotor UAV is a highly non-linear system, and the LQR method is challenging to apply directly [23]. Even after linearization processing, the influence of non-linear factors cannot be completely eliminated, and the control accuracy and robustness are limited. The Model Predictive Control (MPC) realizes control by establishing a system model and performing rolling optimization within the prediction time domain. MPC handles constraints well, but its computational load often conflicts with tight real-time budgets. [28]. The Active Disturbance Rejection Control (ADRC) has strong disturbance rejection and robustness. Nevertheless, its parameter tuning is relatively complex and requires careful adjustment according to the specific system. Otherwise, it is difficult to achieve the best performance [31].

Overall, the current research on quadrotor UAVs faces three primary challenges:

1 Developing an efficient path-planning algorithm for complex environments that can rapidly find the global-optimal path, ensure path smoothness, and adapt to environmental changes.
2 Implementing real-time compensation for external disturbances to guarantee the stable flight of quadrotor UAVs under various interference conditions.
3 Strengthening the robustness of system control to enable quadrotor UAVs to track desired trajectories amidst model uncertainties and external disturbances accurately.

To address the above problems, this paper proposes a hybrid planning-and-control framework that integrates HAMT-AISL-PSO path planning with cubic B-spline smoothing and an active disturbance rejection control (LADRC) tracking method. The main contributions are:

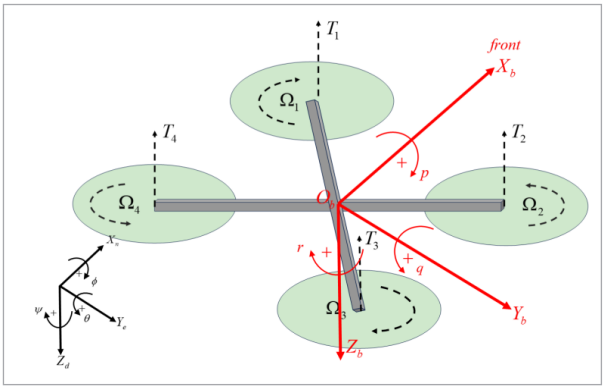
1 A 6-DOF quadrotor dynamic model is derived based on the Newton–Euler formulation, which provides a theoretical foundation for the co-design of planning and control.
2 The proposed HAMT-AISL-PSO algorithm employs multi-topology sub-swarms and adaptive parameter strategies. It is further combined with obstacle-aware fitness evaluation and cubic B-spline smoothing, thereby enhancing global search capability and ensuring path smoothness and feasibility in cluttered three-dimensional environments.
3 An LADRC-based trajectory-tracking method is designed, which incorporates disturbance observation and compensation mechanisms and, under pulse disturbance excitation in simulation, achieves higher tracking accuracy and stronger disturbance-rejection capability than the traditional PID controller.

2. System Modeling

2.1. Quadcopter UAV dynamics model

The quadcopter UAV is driven by four motors, with translational motion along the X-axis, Y-axis and Z-axis in three-dimensional space, as well as three rotational actions of rolling, pitching and yawing, with a total of 6 Degrees of Freedom (6-DOF).

Figure 1
Schematic diagram of a quadcopter drone.



In Figure 1, the rotors are (1, 2, 3, 4), the rotation speeds are (W_1, W_2, W_3, W_4), and the thrust generated by the propeller rotation is (T_1, T_2, T_3, T_4).

Before deriving the model, we make reasonable basic assumptions to simplify the construction of the model within the allowable margin of error:

1 The quadcopter represents a rigid body with uniform symmetry;

2 The mass and moment of inertia of the quadcopter aircraft do not change;

3 The geometric center of the quadcopter coincides with its center;

4 A quadcopter aircraft is influenced solely by propeller thrust and gravity;

5 Counterclockwise rotation is performed by propellers 1 and 3, while clockwise rotation is carried out by propellers 2 and 4.

From the Newton-Euler equation, a rigid body's motion can be viewed as consisting of a translation of its center of mass together with a rotation around that center.

In the body frame, Euler's equation is used to describe the rotation around the center of mass:

$$M = J\dot{w} + w \times Jw\,, \tag{1}$$

where M is the combined moment, J represents the inertial matrix of the UAV, w represents the angular velocity in the body frame, $w \times Jw$ is the moment required to change the direction of its own angular momentum caused by the rotation of the rigid body, which can be derived from Euler's equation.

2.1.1. Dynamic model

Within the dynamic model, the input is composed of the combined external force and moment, whereas the output corresponds to the velocity and angular velocity.

1 **Positional dynamics model**

Based on Newton's second law, there are:

$$\dot{v}^e = g^e - \frac{f^b}{m}, \tag{2}$$

where the acceleration of the UAV's center of mass is denoted by $\dot{v}^e$, G^e is the gravity under the inertial frame, and f^b is the total pull force under the body frame.

Convert the thrust to the representation in the inertial frame and multiply it to the left by the rotation matrix:

$$\dot{v}^e = g^e - R_b^e \frac{f^b}{m}\,, \tag{3}$$

The forces of gravity and pull acting on a quadcopter are directed along the axis $O_e z_e$:

$$\dot{v}^e = ge_3 - \frac{f}{m} R_b^e e_3\,, \tag{4}$$

where e_3 is the unit vector along the central axis $O_e z_e$ of the inertial frame.

Substituting R_b^e and e_3 gives the following:

$$\begin{cases} \dot{v}_x = -\dfrac{f}{m}(C_y S_q C_f + S_y S_f) \\ \dot{v}_y = -\dfrac{f}{m}(S_y S_q C_f - C_y S_f) \\ \dot{v}_z = g - \dfrac{f}{m} C_f C_q \end{cases}. \tag{5}$$

In this paper, C_y denotes cos(y), S_q denotes sin(q), and all similar expressions throughout the paper follow this definition

2 **Establish the attitude dynamics model**

Under the body frame, rotating around the center of mass satisfies the Eulerian equation [20], resulting in:

$$J\dot{w}^b + w^b \times Jw^b = G_a + t\,, \tag{6}$$

where G_a represents the gyroscopic moment. The moment t, generated by the propeller, acts along the fuselage axis and includes the rolling moment t_x about axis $O_b X_b$, the pitch moment t_y around the shaft $O_b X_b$ and the yaw moment t_z around the shaft $O_b X_b$; The angular velocity in the body frame is denoted by ω^b. For the convenience of representation, the three components of w^b on the axis of the body w_x, w_y, w_z are represented by p, q, r:

$$w^b = \begin{bmatrix} w_{xb} \\ w_{yb} \\ w_{zb} \end{bmatrix} = \begin{bmatrix} p \\ q \\ r \end{bmatrix} \tag{7}$$

Regarding the inertial matrix: Based on hypothesis 1 and hypothesis 2, the inertial matrix can be expressed as:

$$J=\begin{bmatrix} J_{xx} & & \\ & J_{yy} & \\ & & J_{zz} \end{bmatrix} \tag{8}$$

Organized:

$$\begin{cases} \dot{p}=\dfrac{t_x+qr\left(I_{yy}-I_{zz}\right)-qWJ_{RP}}{I_{xx}} \\ \dot{q}=\dfrac{t_y+pr\left(I_{zz}-I_{xx}\right)+pWJ_{RP}}{I_{yy}} \\ \dot{r}=\dfrac{t_z+pq\left(I_{xx}-I_{yy}\right)}{I_{zz}} \end{cases} \tag{9}$$

In Equation (9), $W=-v_1-v_2+v_3+v_4$

2.1.2. Kinematic model

The kinematic model takes inputs of velocity and angular velocity, while its outputs are position and attitude.

1 **Positional kinematics model**

$$\dot{p}^e=v^e \tag{10}$$

$$p_e=[a,b,c]^T, \tag{11}$$

where the coordinate position of the quadcopter UAV in the inertial frame is denoted as, and it can be expanded as follows:

$$\begin{bmatrix} \dot{a} & \dot{b} & \dot{c} \end{bmatrix}^T=\begin{bmatrix} v_x & v_y & v_z \end{bmatrix}^T. \tag{12}$$

2 **Attitude kinematics model**

The attitude equation represented by the Euler angle model is:

$$\dot{Q}=W\times w^b, \tag{13}$$

where the W is the conversion matrix from the angular velocity of rotation to the velocity of the Euler angle, Q is the three attitude angles (Euler angles) representing the quadcopter, and there are:

$$\begin{gathered} w^b=\begin{bmatrix} w_{xb} & w_{yb} & w_{zb} \end{bmatrix}^T=\begin{bmatrix} p & q & r \end{bmatrix}^T \\ W=\begin{bmatrix} 1 & T_qS_f & T_qC_f \\ 0 & C_f & -C_f \\ 0 & S_f/C_q & C_f/C_q \end{bmatrix} \end{gathered} \tag{14}$$

When the disturbance is small, that is, when the angle variations are minor, the attitude angle rate can be considered approximately equal to the body's angular velocity of rotation. There are:

$$\begin{bmatrix} \dot{f} \\ \dot{q} \\ \dot{y} \end{bmatrix}=\begin{bmatrix} p \\ q \\ r \end{bmatrix} \tag{15}$$

The flight control rigid body model of the quadcopter is a combination of a dynamic model and a kinematic model [13]. Combine and make that:

$$\begin{bmatrix} U_1 & U_2 & U_3 & U_4 \end{bmatrix}^T=\begin{bmatrix} f & t_x & t_y & t_z \end{bmatrix}^T \tag{16}$$

A quadcopter's non-linear 6-DOF model can be derived as follows:

$$\begin{cases} \ddot{x}=-\dfrac{U_1(\mathrm{C}_y\,\mathrm{S}_q\,\mathrm{C}_f+\mathrm{S}_y\,\mathrm{S}_f)}{m} \\ \ddot{y}=-\dfrac{U_1(\mathrm{S}_y\,\mathrm{S}_q\,\mathrm{C}_f-\mathrm{C}_y\,\mathrm{S}_f)}{m} \\ \ddot{z}=g-\dfrac{U_1\,\mathrm{C}_f\,\mathrm{C}_q}{m} \\ \ddot{f}=\dfrac{I[qr\left(I_{yy}-I_{zz}\right)+U_2-qWJ_{RP}]}{I_{xx}} \\ \ddot{q}=\dfrac{I[pr\left(I_{zz}-I_{xx}\right)+U_3+pWJ_{RP}]}{I_{yy}} \\ \ddot{y}=\dfrac{I\left[pq\left(I_{xx}-I_{yy}\right)+U_4\right]}{I_{zz}} \end{cases} \tag{17}$$

It is this model that completely describes the non-linear mapping between the control input (motor speed) and the motion state (position p_e, attitude f, q, y)of the quadcopter UAV, which is the core basis for subsequent path planning and robust tracking control.

2.2. UAV Path Planning Model

At the constraint processing level, the UAV flight constraint is integrated into the construction of the fitness function. Assuming that the path is characterized by a discrete point sequence $X = \{(X_1, Y_1, Z_1), ..., (X_M, Y_M, Z_M)\}$, assuming that there are C paths, each consisting of n points, and that g hemispherical or dome-cylindrical obstacles coexist in the environment, the fitness function is defined as follows:

$$f(N) = a\,\mathrm{L}(N) + b\,\mathrm{C}(N)\,, \tag{18}$$

where L(N) represents the path length cost, C(N) is the obstacle risk cost, a and b are the weight coefficients. The priority of each target is balanced by normalization.

Among the most important indicators used to evaluate path quality [29] is the path length, given that shorter paths consume less time and energy. The cost function, including the path length, can be expressed as:

$$\mathrm{L}(X) = \sum_{j=0}^{n} \sqrt{(a_{j+1} - a_j)^2 + (b_{j+1} - b_j)^2 + (c_{j+1} - c_j)^2}\,, \tag{19}$$

where L(X) represents the sum of the distances between all adjacent nodes in the path mth, $m \in \{1, 2, \ldots, M\}$, and ($a_j$, b_j, c_j) is the coordinates of the jth node in the path.

2.3. Mission Requirements and Constraints

The danger cost of the obstacle is introduced, so that the adaptability of the path too close to the obstacle is poor, and the final path is kept at a certain distance from the obstacle [8]. For the barrier k,(k=1, 2,..., g), if it is a hemispherical barrier, find the Euclidean distance from the center of the sphere to each path segment, and find the distance l_i^k from the center of the sphere to each path segment, then the minimum distance from the path to the obstacle is $L_k = \min(I_1^k, ..., I_i^k, ..., I_{n-1}^k)$, the path and the xoy obstacle are projected on the planes, and the distance l_i^k from the center of the obstacle to each path segment is found to be $L_k = \min(I_1^k, ..., I_i^k, ..., I_{n-1}^k)$.

Then the cost of the obstacle threat of path m is:

$$\mathrm{dan}_m = \begin{cases} \sum_{k=1}^{g} L_k, & L_k > r_k (k = 1, 2, \ldots, g) \\ \infty, & \text{otherwise} \end{cases}, \tag{20}$$

where is the radius of a spherical barrier or a cylindrical barrier.

Based on the above analysis, the derived 6-DOF nonlinear model describes the full mapping from motor speed to the quadrotor's position and attitude, thereby capturing its actual motion capability and dynamic constraints in 3D space. Further, the path-planning model and mission-related constraints embed these dynamic limits into the cost function through path length, obstacle-threat penalties, and safety margins, so that candidate trajectories are both collision-free and physically feasible. On this basis, this section will design an HAMT-AISL-PSO path-planning algorithm that searches for optimal trajectories within this constrained space, which will give paths suitable for subsequent LADRC-based tracking control.

3. Planning Methods Development

3.1. Introduction to the Standard PSO Algorithm

3.1.1. Core Update Mechanism of the PSO Algorithm

The PSO algorithm, inspired by the collaborative foraging behavior of bird flocks, iteratively refines particle positions in a D-dimensional search space through dynamic updates of velocity and position. It is assumed that within the search space, there exists a swarm consisting of N particles, where the position and velocity of the ith particle are denoted by $X_i = (x_{i1}, x_{i2}, \ldots, x_{iD})$ and $V_i = (v_{i1}, v_{i2}, \ldots, v_{iD})$, respectively, where $i = 1, 2, \ldots, N$. The velocity update rule from generation t to t+1 is given by equation (22) [9].

$$v_{ij}(t+1) = v_{ij}(t) \times w + [p_{ij}(t) - x_{ij}(t)]c_1 r_1(t) + [p_{gj}(t) - x_{ij}(t)]c_2 r_2(t) \quad (21)$$

Where w denotes the inertia weight that balances the particle's global exploration and local exploitation abilities, and its magnitude controls how much of the particle's previous velocity is preserved. c_1 and c_2 are the acceleration constants (usually the value is $c_1 = c_2 = 1.5$), which adjust the step size of the particle to learn the individual optimal position p_{ij} and the global optimal position p_{gj}, respectively r_1 and r_2 are random numbers in the interval [0,1]. The velocity equation consists of three parts: the inertia term $w \times v_{ij}(t)$, the individual cognitive term $\left[p_{ij}(t) - x_{ij}(t)\right]c_1 r_1(t)$ and the social cognitive term $\left[p_{gj}(t) - x_{ij}(t)\right]c_2 r_2(t)$. The coherence of particle motion is preserved by the inertial term, while the individual cognitive component captures how each particle learns from its own experience, and the social cognition component represents its reference to the group's optimal experience.

In the particle swarm optimization algorithm, the position update equation is the key link to realize the iteration of the particle position [24]. Its mathematical expression is as follows:

$$x_{ij}(n+1) = x_{ij}(n) + v_{ij}(n+1), \quad (22)$$

where $X_{ij}(n)$ represents the position of the ith particle at the tth iteration in the j-dimensional space, $v_{ij}(n+1)$ and is the velocity of the particle after it is updated in the j-dimensional space. When the particle completes the update of its own motion direction and velocity according to the velocity equation, it will adjust its position accordingly based on the new velocity [7]. Through this process, the particle realizes the transition that marks the change from the current state to the next state, thereby completing an iteration. Each iteration is an attempt by particles to approximate a better solution in the search space, and the process of continuous iteration of many particles prompts the entire particle swarm to gradually converge to the region in which the global optimal solution resides, which is the focus, providing an effective search strategy for the generation and optimization of the UAV 3D trajectory.

3.1.2. Adaptability Analysis of Standard PSO Algorithms

The standard PSO algorithm continuously explores the optimal path in the solution space through a unique update mechanism. However, in the complex task of UAV 3D trajectory generation, the dimensionality of the solution space increases significantly because of the need to accurately optimize the spatial position coordinates of the UAV, as well as the need to consider various flight constraints such as specific altitude restrictions and complex obstacle avoidance rules. With an increase in dimensionality, the area to be traversed in the process of the algorithm search increases exponentially, the convergence speed is significantly reduced, and hundreds or even thousands of iterations are often required to approximate the feasible solution, which significantly increases the computational cost and time consumption; and significantly restricts the algorithm's applicability in scenarios where high real-time demands are crucial.

In addition, the UAV flight environment is usually highly complex, with a large number of obstacles, such as dense building complexes in cities, rolling mountains, etc., irregular layouts and terrain, which make the fitness function present multimodal characteristics. Local extreme points easily attract the particles of the standard PSO algorithm during the search process. It is challenging to jump out of exploring the global optimal solution in a broader space after falling into the local optimum, and the generated path is only locally optimal.

In simulation experiments, this issue becomes especially apparent, as obtaining the global optimal path is not possible. The path-based control algorithm verification remains incomplete, making it challenging to evaluate the true performance of the control algorithm in complex scenarios, which may lead to incorrect judgments about the algorithm and impact both its practical application and the safety of the UAV system.

3.2. Improved Strategy: Adaptive Inertia Weight and Constraint Handling

In response to the shortcomings of traditional PSO, this study proposes hybrid adaptive multi-topology with intelligent strategy learning PSO (HAMT-AISL-PSO) algorithm, which enhances the search performance through multi-topology switching,

parameter adaptive adjustment and perturbation mechanism, and ensures the feasibility and smoothness of the path in combination with coordinate constraints and B-spline smoothing, the improved algorithm has significant advantages in UAV path planning applications.

3.2.1. Core Architecture of the HAMT-AISL-PSO Algorithm

The HAMT-AISL-PSO algorithm is a hybrid optimization algorithm, which integrates the multi-topology advantages of DAMT-PSO and the multi-strategy characteristics of AISL-PSO. It can achieve efficient optimization of 3D path planning for UAVs in complex environments through the "three-layer collaborative architecture": the top layer is the RL intelligent control layer, which dynamically outputs inertial weights and learning factors based on the system state, and decides on topology switching instructions to avoid the search direction deviating from the optimal target; The middle layer is a multi-topology subpopulation search layer, which constructs a distributed search system with five parallel subpopulations and a total particle count of 180, which greatly broadens the coverage of the algorithm for complex solution space and improves the comprehensiveness and efficiency of path optimization. The bottom layer is the coordinate constraint adaptation layer, which uses the spherical coordinates to represent the position of the particles, directly adapts to the dynamic constraints of the UAV, and ensures that, from the bottom, the path generated by the algorithm conforms to the actual flight capability of the UAV and is physically feasible.

3.2.2. Key Improvement Mechanisms

3.2.2.1 Fusion Speed Update Mechanism

In the iteration process of the traditional PSO algorithm, particle velocity updates rely on a single pattern of individual and global experience for a long time, which limits the balance between exploration breadth and development accuracy. In the traditional PSO velocity update formula, topological neighborhood optimum, quantum behavior global optimum, historical memory experience, and Lévy flight perturbation are introduced to form a multi-information fusion update mechanism to enhance the ability of particles to jump out of the local optimum. Its mathematical expression is as follows:

$$\begin{aligned} v_i^{t+1} &= v_i^t w(t) + r_1 c_1(t)\left(pbest_i - x_i^t\right) \\ &+ r_2 c_2(t)\left(nbest_{topology} - x_i^t\right) \\ &+ c_3(t) r_3\left(gbest_{quantum} - x_i^t\right) \\ &+ \alpha(t) r_4\left(m_t - x_i^t\right) + \beta(t) G_i^t \end{aligned}, \quad (23)$$

where $w(t)$ is the dynamic inertia weight, which maintains the coherence of particle motion; $c_1(t)$ is the adaptive learning factor, which guides the particles to converge to the optimal solution of individual history; $c_2(t)$ is the topological learning factor, which guides the particles to learn to the optimal solution of the neighborhood under the current topology $nbest_{topology}$; $c_3(t)$ is the quantum learning factor, which guides the particles towards the quantum global learning center $best_{quantum}$, which can be equivalent to the mean best or elite average position; $a(t)$ is the memory influence factor, which guides the particles to refer to the historical excellent solution m_t in the elite set to enhance the search diversity and robustness; $b(t)$ is the Lévy perturbation intensity factor, and L_i^t is the random perturbation term following the Lévy distribution, which is used to jump out of the local optimum.

3.2.2.2. Parameter Adaptive Mechanism

In traditional PSO, key parameters such as inertial weights and learning factors often rely on empirical settings or only linear adjustments, making it difficult to adapt to complex search stages. The improved mechanism adopts the Actor-Critic reinforcement learning model, collects the system state in real time, and dynamically outputs parameters such as inertia weight and learning factor according to the iteration stage and environmental complexity, which greatly improves the rationality of parameter configuration and algorithm robustness. The mathematical expression is as follows:

$$\{w(t), d(t), d_2(t), d_3(t), d(t), b(t)\} = \mathrm{F}_{\mathrm{RL}}(S_t), \quad (24)$$

where S_t can contain the distribution information of the particle swarm, the convergence velocity, fitness fluctuations, and other state characteristics, F_{RL} is implemented by a reinforcement learning model.

3.2.2.3. Intelligent Topology Switching Decision-making Mechanism

The traditional PSO population has a fixed topology

and a single information interaction mode, which will lead to premature homogenization of particles and getting stuck in a local optimum. The improved mechanism takes the Q-value function as the basis for decision-making. It adaptively switches among topologies such as ring and star according to population diversity and convergence speed to ensure that the population always maintains a reasonable distribution. The expression of the decision model is:

$$T(t) = \arg\max_{k \in \mathcal{I}} Q(S_t, k), \tag{25}$$

where $Q(S_t,k)$ is the action value function of the combination of different system states and topologies in reinforcement learning covers five topology types: ring, star, fully connected, dynamic adaptive, and grid.

3.2.3. Implementation Steps of the HAMT-AISL-PSO Algorithm

Based on the aforementioned principles (Figure 2), the pseudocode and computational flowchart of HAMT-AISL-PSO are depicted in Algorithm 1 and Figure 3.

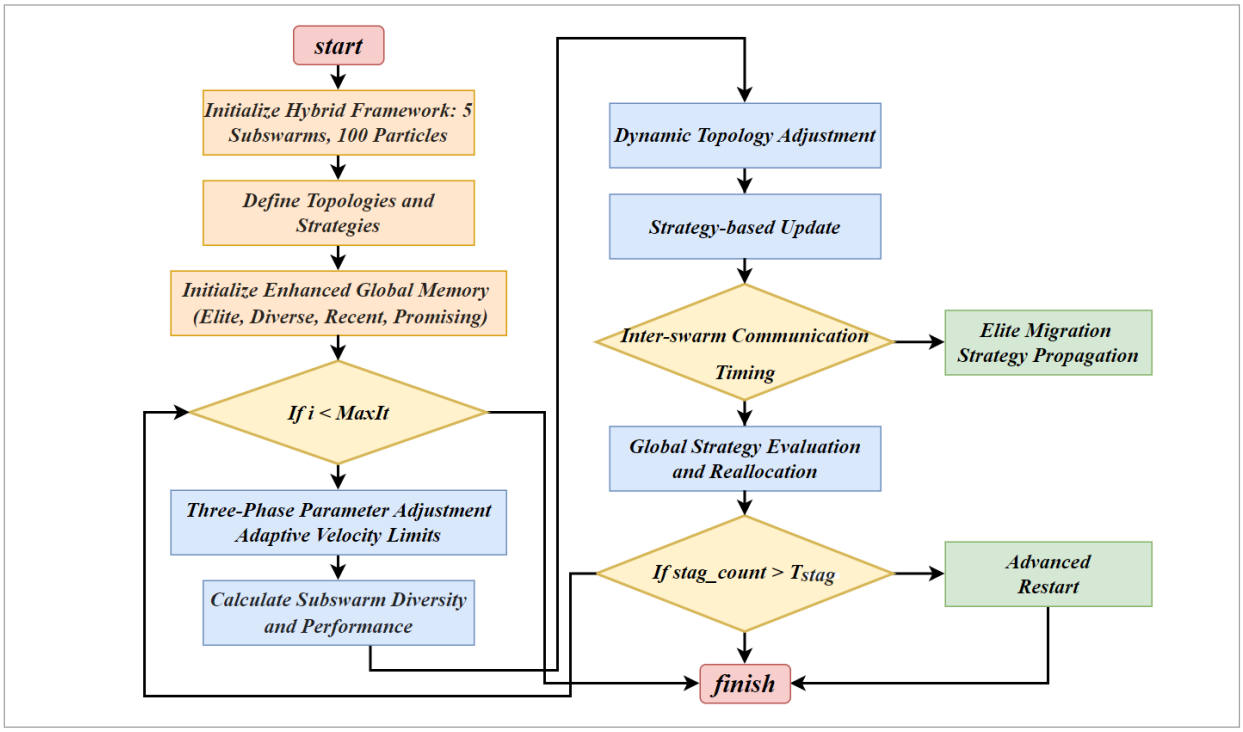
Algorithm 1: HAMT-AISL-PSO (Standard Compact Version)

Input: f: $R^n \rightarrow R$, bounds $[x_{min}, x_{max}]$

Output: gBest $\in R^n$

```
procedure HAMT_AISL_PSO (f, x_min, x_max)
    n_subswarms ← 5; nPop ← 180; subswarm_size ← 36
    T ← {RING, STAR, FULL, DYNAMIC, MESH}
    // Topologies
    S ← {STANDARD, QUANTUM, LEVY, COMPREHENSIVE, HYBRID}// Strategies
    M ← {elite: [], diverse: [], recent: []} // Memory banks
    // Initialize Subswarms
    for s ← 1 to n_subswarms do
        subswarm[s]. topology ← T[s mod 5];
        subswarm[s]. strategy ← S[s mod 5]
        for i ← 1 to subswarm_size do
            p ← CreateParticle(); p. strategy ← Select-Strategy (s, i)
            p. cost ← f(p.x); UpdateBests (p)
        end for
    end for
```

Figure 2
HAMT-AISL-PSO algorithm flowchart.



```
// Main Optimization
for iter ← 1 to MaxIt do
    t ← iter / MaxIt; (w, c1, c2) ← GetParameters (phase, t)
    for s ← 1 to n_subswarms do
        if Diversity(s) > θ then subswarm[s].topology ← RING
        else if Stagnant(s) then subswarm[s].topology ← FULL
        end if
        for each p in subswarm[s] do
            lBest ← LocalBest (p, subswarm[s])
            switch p.strategy do
                QUANTUM: QuantumUpdate (p, lBest, t)
                LEVY: LevyUpdate (p, lBest)
                default: StandardUpdate (p, lBest, w, c1, c2)
            end switch
            p. x ← Bound (p.x, xmin, xmax)
            p. cost ← f(p.x)
            UpdateBests (p)
        end for
    end for
    if iter mod 3 = 0 then MigrateEliteParticles (subswarms)
    end if
    if GlobalStagnation () > 8 then
        RestartWorstParticles (30%, M.elite)
    end if
end for
return gBest
end procedure
```

3.3. Path Smoothing Optimization Based on Cubic B-spline Curves

To enhance the smoothness and continuity of the UAV flight path and meet its dynamic smoothness requirements, a cubic B-spline curve is employed to fit the discrete waypoints generated by Particle Swarm Optimization (PSO), thereby optimizing the initial path. The mathematical formulation of the B-spline is given as follows:

$$B(t)=\sum_{i=1}^{n} P_i B_{i,3}(t), t\in[t_0, t_{n+3}], \quad (26)$$

where P_i represents the control point, $B_{i,3}(t)$ is the cubic B-spline basis function, which is defined by the Cox - de Boor recursive formula:

$$B_{i,k}(t)=\frac{t-t_i}{t_{i+k}-t_i}B_{i,k-1}(t)+\frac{t_{i+k+1}-t}{t_{i+k+1}-t_{i+1}}B_{i+1,k-1}(t). \quad (27)$$

The initial condition is set as follows: when $t_i \le t < t_{i+1}$, $B_{i,0}(t)=1$; In the rest of the cases, $B_{i,0}(t)=0$. In this paper, the B-spline smoothing uses uniform parametrization. All waypoints (n = 12) are mapped to the parameter interval [0, 1] with knots $t_i = i/(n-1)$. A cubic B-spline (k = 3) is then constructed using spline with zero smoothing factor (s = 0), so that the smoothed path exactly interpolates all nodes while maintaining C^2 continuity. The final reference trajectory is obtained by sampling 200 uniformly spaced points along the parameter domain for guidance and plotting.

The core appeal of trajectory smoothing optimization is to reduce the curvature of the path. The formula for calculating the curvature of a spatial curve is:

$$k(s)=\frac{\| r'(s)\times r''(s)\|}{\| r'(s)\|^3}, \quad (28)$$

where is $r(s)$ the parametric path, and s is the arc-length parameter. By adjusting the B-spline control points reasonably, the curvature abruptness in the path can be effectively reduced. In the process simulation experiments, the UAV can effectively avoid the control shock problem caused by the sharp turn of the path, and provide strong support for the stability verification of the control algorithm [21].

4. Robust Controller Design

4.1. Deficiencies of Traditional Trajectory Tracking Control Methods

In the design of quadrotor UAV control systems, PID control plays an important role. Owing to its simple structure and ease of implementation, it offers low development and debugging costs and is highly suitable

for engineering applications. It is robust, can cope with external interference and small-scale changes in model parameters to a certain extent, ensures the basic flight attitude and trajectory tracking of UAVs, and is widely used in many fields. The accumulated experience and parameter tuning methods can be used for reference. However, the dynamic quality margin of PID is limited, and the closed-loop dynamic quality is sensitive to gain changes, and the gain needs to be adjusted frequently when the UAV flight environment is subject to dynamic changes [16]. Moreover, the error handling is an unreasonable control requirement.

In a highly dynamic environment, the trajectory tracking control of quadrotor UAVs is also insufficiently addressed by existing MPC and LQR.MPC relies on model prediction. In a highly dynamic environment, the state of the UAV changes rapidly, and the model is difficult to accurately capture in real time [26], which can easily lead to an increase in the prediction error. LQR is based on linear system design, and the non-linear characteristics of UAVs are significant during high-dynamic flights, the linear assumption of LQR is no longer applicable, which cannot accurately describe the system dynamics, resulting in poor control effects and difficulty in achieving accurate trajectory tracking and attitude control. Therefore, methods such as ADRC need to be introduced to enhance wdwwcontrol performance.

4.2. Principle and Design of ADRC

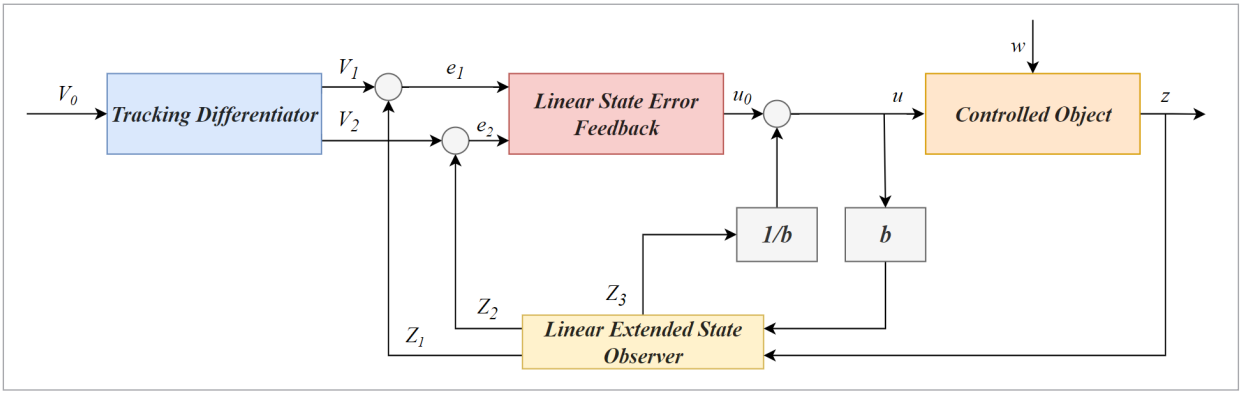
ADRC, as an advanced control method characterized by unique design and principles, is composed of three core elements: the Tracking Differentiator (TD), the Extended State Observer (ESO), and the Non-linear State Error Feedback (NLSEF). The TD addresses the conflict between system response speed and overshoot by shaping the transition process and simultaneously extracting appropriate differential signals, which help smooth the system input and mitigate noise. The ESO incorporates the "total disturbance", including unmodeled dynamics and external interferences, into new state variables that are estimated in real time using the system's input-output data. The NLSEF uses the error between the TD output and ESO estimation as input, applying non-linear functions to map the control action and thereby realize precise regulation of the system state [15].

However, in practical applications, ADRC has problems with many parameters and complex stability analysis due to its non-linear structure. For the purpose of overcoming such shortcomings, Linear Active Disturbance Rejection Control (LADRC) has emerged. LADRC is an improvement over ADRC with a linear structure that simplifies the design and parameter adjustment process.

The LADRC approach demonstrates strong feasibility. From a theoretical perspective, it builds upon modern control theory, allowing real-time estimation and compensation of internal and external disturbances for effective handling of non-linear and uncertain system behaviors. In practical applications, LADRC does not rely on accurate system models, exhibits strong robustness against model parameter perturbations and external interference, and can be widely applied in aerospace, mechanical engineering, power electronics, and many other fields. In the numerical experiments of this paper,

Figure 3

The structure of the second-order linear adaptive disturbance rejection controller.



this disturbance-rejection property is evaluated under pulse disturbances acting on the quadrotor attitude channels. For example, in aircraft control, LADRC can ensure stable control of flight attitude in the face of complex airflow interference and model uncertainty. The motor speed regulation system can cope with disturbances such as load changes and maintain a stable rotational speed.

4.3. Design Based on LADRC UAV Control System

4.3.1. Non-linear Error Feedback

To prevent the system from reaching saturation while ensuring sensitivity to errors, the outer-loop controller adopts a non-linear power fal function [1]:

$$c_0 = k_0 \cdot fal(j_{err}, a_{nl}, d_{nl}), \tag{29}$$

where c_0 is the output signal of non-linear feedback; k_0 is the gain coefficient of non-linear and error feedback; $j_{err} = j_r - j$, that is, the deviation between the desired and actual attitude angles; a_{nl} and d_{nl} are the parameters of the non-linear error feedback link. The fal function is defined as

$$\mathrm{fal}(c,a,d) = \begin{cases} \dfrac{c}{d^{1-a}} & |c| \le \delta \\ |c|^{\alpha}\,\mathrm{sign}(x) & |c| > \delta \end{cases}, \tag{30}$$

where c is the independent variable, a and d are constant values. In this controller, order $a = 0.5$. When the error is less than d, the function fal is a linear function, and the error gain of the outer ring is constant $1/d^{1-a}$, which is linear. When the error is greater than d, the function fal is a power function, and the error gain of the outer ring is $a|c|^{a-1}$, which gradually decreases with the increase of x [4]. Replacing the power function with a linear one near zero prevents the derivative of the power function from approaching infinity as the error tends to zero.

The inner-loop controller takes the output of the outer-loop controller as its error input:

$$e_0 = c_0 - \dot{j}, \tag{31}$$

where is the input signal of the tracking differentiator.

4.3.2. Tracking Differentiator (TD)

In the inner ring control, the tracking differentiator is selected as the controller. The tracking differentiator can be regarded as a PD controller with a filtering function fhan [10], which effectively avoids the occurrence of high-frequency vibration problems in the digital computing process with the help of the fast and optimal control synthesis function fhan of the discrete system, and can obtain a more ideal differential signal. In the process of numerical simulation and physical experiments, the researchers found that the differential signals obtained by using traditional PD controllers have significantly high-frequency jitter. When the step signal is used as input, the resulting differential signal will also have overshoot. Based on the above situation, compared with the traditional PD controller, it is a better choice to use the tracking differentiator as the controller. The algorithm for tracking differentiators is as follows:

$$\begin{cases} fh = fhan(e_1(k-1) - e_0(k), e_2(k-1), r_0, h_0) \\ e_1(k) = e_1(k-1) + he_2(k-1) \\ e_2(k) = e_2(k-1) + hfh \end{cases}, \tag{32}$$

where e_1 is the tracking signal of e_0 output by the tracking differentiator; e_2 is the differential signal of e_0 output by the tracking differentiator; r_0 is the speed factor that determines the twracking speed; h_0 is the filter factor; h is the integral step calculated by the digital controller. Note $fsg(x,d_0) = \dfrac{sign(x+d_0) - sign(x-d_0)}{2}$, then the definition of function fhan is $fh = fhan(x_1, x_2, r, h)$, there is

$$\begin{cases} d_0 = rh^2 \\ a_0 = hx_2 \\ \mathrm{y} = x_1 + a_0 \\ a_1 = \sqrt{d_0(d_0 + 8|y|)} \\ a_2 = a_0 + sign(y)(a_1 - d_0)/2 \\ a = (a_0 + y)fsg(y,d_0) + a_2(1 - fsg(y,d_0)) \\ fh = -r(a/d_0)fsg(a,d_0) - r \cdot sign(a)(1 - fsg(a,d_0)) \end{cases}, \tag{33}$$

where r is the entry parameter of the function fhan.

4.3.3. Linear Expansion State Observer (LESO)

The Linear Extended State Observer (LESO) is the core algorithm component of the ADRC. The basic

principle is to expand the various interference effects that affect the controlled output of the system into a new state variable. Through the control quantity and controlled output, the system is dynamically estimated to be disturbed, and the control amount is compensated accordingly, which is the technology of dynamic estimation and compensation for total disturbances. The algorithm does not rely on a specific mathematical model of interference or directly measure the magnitude of interference using sensors. During the flight of a quadrotor UAV, its attitude control is highly susceptible to moment interference generated by various air currents, and it is difficult to measure or pre-model these interferences in real time effectively. Therefore, it is a suitable strategy to use an expansion state observer to observe and compensate for moment interference.

In view of the fact that among the sensors equipped with a quadrotor UAV, the gyroscope measures the attitude angular velocity with the highest accuracy, in the expansion state observer of the attitude angle control channel, $\dot{j}$ is directly used as the input quantity of the observer, and its observed value is recorded as z_1, then $\dot{z}_1$ corresponds to the attitude angular acceleration. Since the moment interference will affect the angular acceleration of the system, the observed value of the moment interference is recorded as z_2, and the following LESO can be constructed based on this:

For each attitude channel, the second-order dynamics are rewritten in the extended-state form by defining x_1 as the angular position, x_2 as its rate, and collecting all internal and external moments into an additional state x_3. This yields the canonical structure $\dot{x}_1=x_2$, $\dot{x}_2=bu+x_3$, $\dot{x}_3=a(x,t)$, which is used to construct the LESO in Equation (34).

$$\begin{cases} e(t)=z_1(t)-x_1(t) \\ \dot{z}_1(t)=z_2(t)-b_{01}\times e(t) \\ \dot{z}_2(t)=z_3(t)-b_{02}\times fal(e(t),a_1,d)+bu(t) \\ \dot{z}_3=-b_{03}\times fal(e(t),a_2,d) \end{cases}. \quad (34)$$

$$\rightarrow \begin{cases} \dot{z}_1=z_2-\beta_1(z_1-y) \\ \dot{z}_2=z_3-\beta_2(z_1-y)+bu \\ \dot{z}_3=-\beta_3(z_1-y) \end{cases}$$

To clarify the tuning of the LESO, let $e=y-z_1$ denote the output estimation error. From (34) the observer error dynamics form a third-order linear system. In this paper the gains β_1, β_2, β_3 are designed by pole placement: the characteristic polynomial of the error dynamics is assigned to $(s+w_0)^3$, so that all observer poles are located at $s=-w_0$ and the observer bandwidth is w_0. Matching coefficients of this polynomial yields the relationships between β_1, β_2 and β_3; once w_0 is chosen, the LESO parameters are uniquely determined and the same tuning principle is applied to all attitude channels.

$$\beta_1=3\omega_0,\beta_2=3\omega_0^2,\beta_3=\omega_0^3,\omega_0>0, \quad (35)$$

where β_1 β_2 and β_3 are observer parameters, selected by the pole placement method.

Figure 4

The structure of control system based on LADRC.

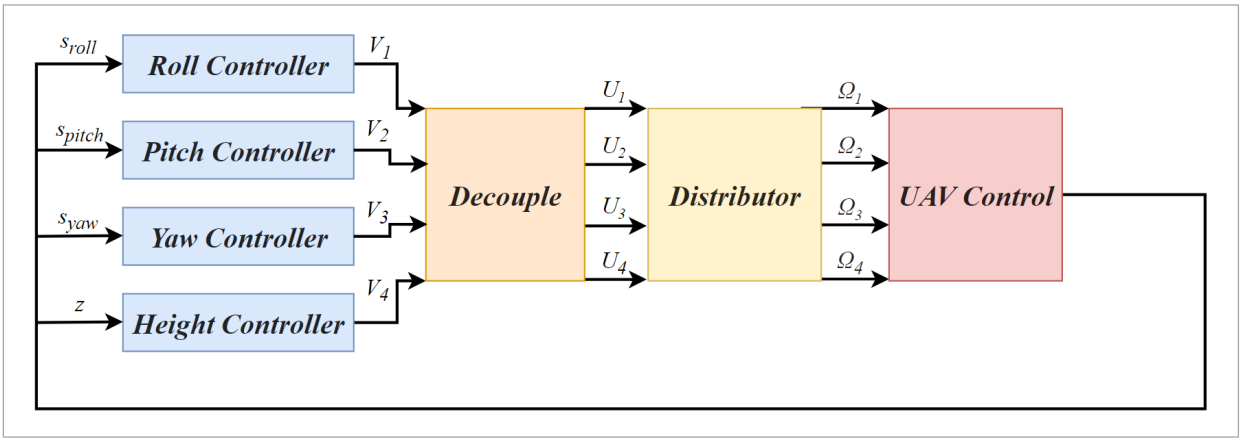


Figure 4 shows the control system structure based on LADRC, including roll, pitch, yaw, and altitude controllers, as well as decoupling, allocation, and execution modules. By configuring the observer poles, the estimated states quickly converge to the actual states, allowing disturbance compensation and improving system stability and dynamic performance.

4.3.4. Pitch Angle Controller Design

In aerospace and related fields, the pitch angle serves as a key parameter to describe an aircraft's attitude. In aerodynamics, the angle formed between the aircraft's longitudinal axis and the horizontal plane is defined as the pitch angle. When the aircraft nose rises, tilting the fuselage axis upward with respect to the horizontal plane, the pitch angle becomes positive. Accurate regulation of the pitch angle is crucial for stable flight, trajectory tracking, and mission performance, as its variation directly influences the motion state and aerodynamic properties of the aircraft in the vertical direction.

The pitch channel model is rewritten as $\ddot{q} = f_2(j, \dot{j}, q, \dot{q}, y, \dot{y}) + w_2 + b_2 u_2$, and it is further deduced that there are $\dot{q} = x_{q1}, \dot{x}_{q1} = f_2 + b_{1_q} u_q$ orders $x_{q2}=f_2$ (x_{q2}) is the pitch channel's expansion state variable), and the model of the pitch channel can be represented as [25]

$$\begin{cases} \dot{x}_{q1} = x_{q2} + b_{q1} u_q \\ \dot{x}_{q2} = h \end{cases}, \tag{36}$$

where h is the differentiation of f_2.

The observation equation of the LESO is

$$\begin{cases} \dot{z}_{q1} = z_{q2} - b_{1q} e_{q1} + b_{1q} u_q \\ \dot{z}_{q2} = -b_{2q} e_{q1} \end{cases}, \tag{37}$$

where $z_{\theta 1}$ and $z_{\theta 2}$ observes $x_{\theta 1}$ and $x_{\theta 2}$ respectively, and $e_{q1}=z_{q1}-q, b_{1q}, b_{2q}$ is the observer parameter.

The control law based on linear state error feedback is

$$\begin{cases} u_{0q} = k_{pq}(q_d - z_{q1}) + k_{dq} z_{2q} \\ U_2 = u_{0q} - z_{3q} / b_{0q} \end{cases}. \tag{38}$$

5. Results Analysis

5.1. Construction of Simulation Environment

In order to verify the effectiveness of the HAMT-AISL-PSO algorithm, comparative experiments were conducted on the traditional PSO, DAMT-PSO, AISL-PSO, GA, ACO, and the HAMT-AISL-PSO proposed in this paper. To reduce the contingency of experimental results, each of the six algorithms performed 100 experiments independently.

The simulation experiment was performed on a computer configured with Windows 11, 32.00 GB of running memory, and an Intel (R) Core (TM) i9-14900HX processor using MATLAB and Simulink.

5.2. Comparative Experiment of Path Planning Algorithms

Tables 1-2 give the statistical results of the six algorithms in terms of cost function value and path length,

Table 1

Statistics of cost function simulation results of six algorithms.

Algorithm	Mean Cost	Std Cost	Min Cost	Max Cost
traditional PSO	7015.12	664.86	6350.26	7679.98
DAMT-PSO	5726.41	27.23	5699.18	5753.64
AISL-PSO	6156.34	8.00	6148.34	6164.34
HAMT-AISL-PSO	5247.42	7.75	5239.67	5255.17
GA	6738.46	611.85	6126.61	7350.30
ACO	8023.53	192.83	7830.70	8216.36

Table 2
Statistics of path length simulation results of six algorithms.

Algorithm	Mean Path	Std Path
traditional PSO	1206.17	174.52
DAMT-PSO	978.87	17.20
AISL-PSO	1092.40	66.41
HAMT-AISL-PSO	959.48	6.18
GA	1131.89	79.68
ACO	1087.16	40.74

respectively. It can be seen that HAMT-AISL-PSO is consistently better than the other methods in both indicators. The average cost is 5247.42, which is 25.20% lower than that of traditional PSO, and the average path length is 959.48 m, shortened by 20.44%. At the same time, its standard deviation is also the lowest, indicating that the algorithm has obvious advantages in terms of stability and robustness. The mean cost and path length of Genetic Algorithm (GA) were inferior to HAMT-AISL-PSO, with noticeably poorer stability.

From the path planning results (Figures 5-6), it can be seen that the path of traditional PSO in the obstacle-dense area is too tortuous, and some paragraphs are even infeasible. DAMT-PSO and AISL-PSO have improved search capabilities to some extent, but there

Figure 5
3D path comparison.

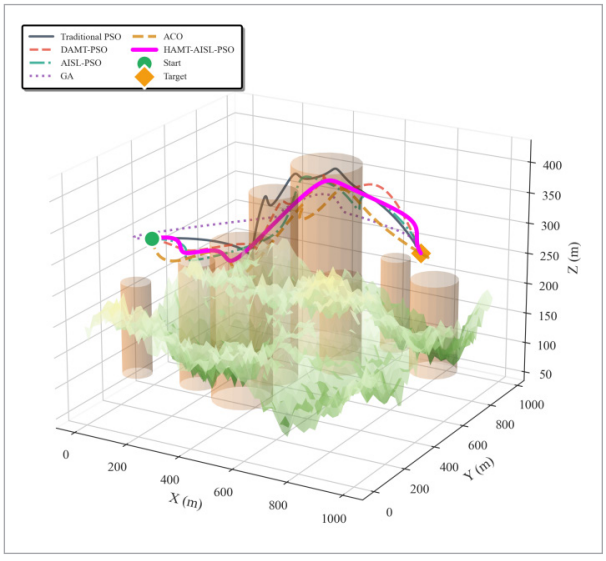


Figure 6
Top view.

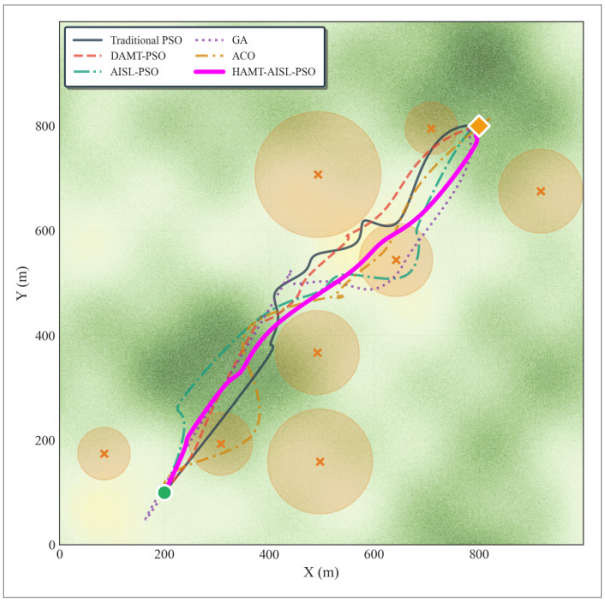
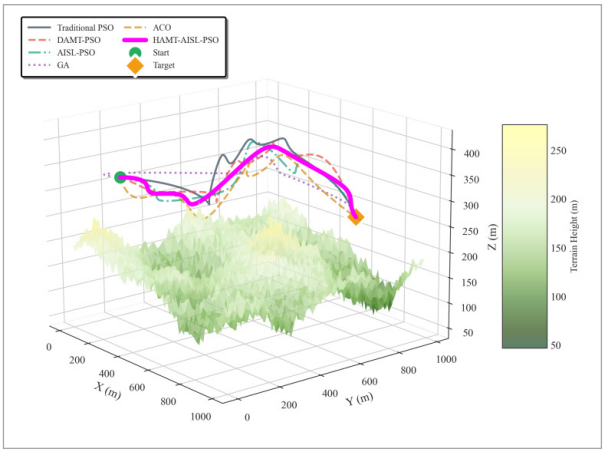


Figure 7
3D terrain with optimal paths.



are still problems of long and insufficiently smooth paths [22]. In contrast, the path generated by HAMT-AISL-PSO is simpler and smoother, which can effectively shorten the range while avoiding obstacles. Further observation of the optimal path comparison in the three-dimensional environment (Figure 7) shows that the trajectory generated by HAMT-AISL-PSO is more stable in altitude variation, indicating that the improved spherical coordinate constraint and B-spline smoothing mechanism effectively improve the executability of the trajectory.

Figure 8 further confirms that the improved algorithm has advantages in performance. Here, the horizontal axis is the iteration number and the vertical axis is the value of the cost function before normalization. All four algorithms converge within fewer iterations, but the speed at which they achieve this goal is different: traditional PSO converges slowly, while DAMT-PSO and AISL-PSO can reduce the value of the cost function faster in the early stage, whereas HAMT-AISL-PSO converges fastest, obtaining the lowest objective value within the fewest iterations. This shows that the algorithm proposed in this paper has higher convergence efficiency and optimization performance [66]. From this, it can be seen that not only does the proposed algorithm accelerate convergence but also it maintains a superior balance between exploration and exploitation throughout the run. Thus, the solutions obtained after the optimization process become consistently better with fewer iterations, reflecting the algorithm's high efficiency and robustness.

Figure 9 presents the comparison of altitude profile, showing the differences in the flight altitude dimension of each algorithm's paths. Within this figure, the altitude curves of traditional PSO, DAMT-PSO, AISL-PSO and HAMT-AISL-PSO are plotted for the same horizontal ground track. The altitude curve generated by traditional PSO fluctuates violently and results in high flight risks. Although DAMT-PSO and AISL-PSO have improved, the overall variation is still not smooth enough. Therefore, the height curve generated by HAMT-AISL-PSO is the smoothest, with the smallest fluctuation amplitude, which meets the dynamic characteristics and mission requirements of UAV flights.

From the point of view of computational complexity, the four PSO-based algorithms in Table 3 (Traditional PSO, DAMT-PSO, AISL-PSO and HAMT-AISL-PSO) have the same asymptotic time complexity per run. According to our analysis, the total cost of PSO-based 3D path planning is O(MaxIt × nPop × n × (D + T_cost)); in our experiments D = 3 is fixed and the cost function meets T_cost = O(n × num_threats) ≈ O(n), so the leading term reduces to O(MaxIt × nPop × n^2). DAMT-PSO adds an extra low-order term O(-MaxIt/5 × n_subswarms × subswarm_size), and AISL-PSO as well as HAMT-AISL-PSO introduce multi-topology and strategy-learning modules with about 1.5–2.5× constant overhead, but these modifications only change the constant factor and do not change the leading order O(MaxIt × nPop × n^2).

Figure 8
Convergence curves.

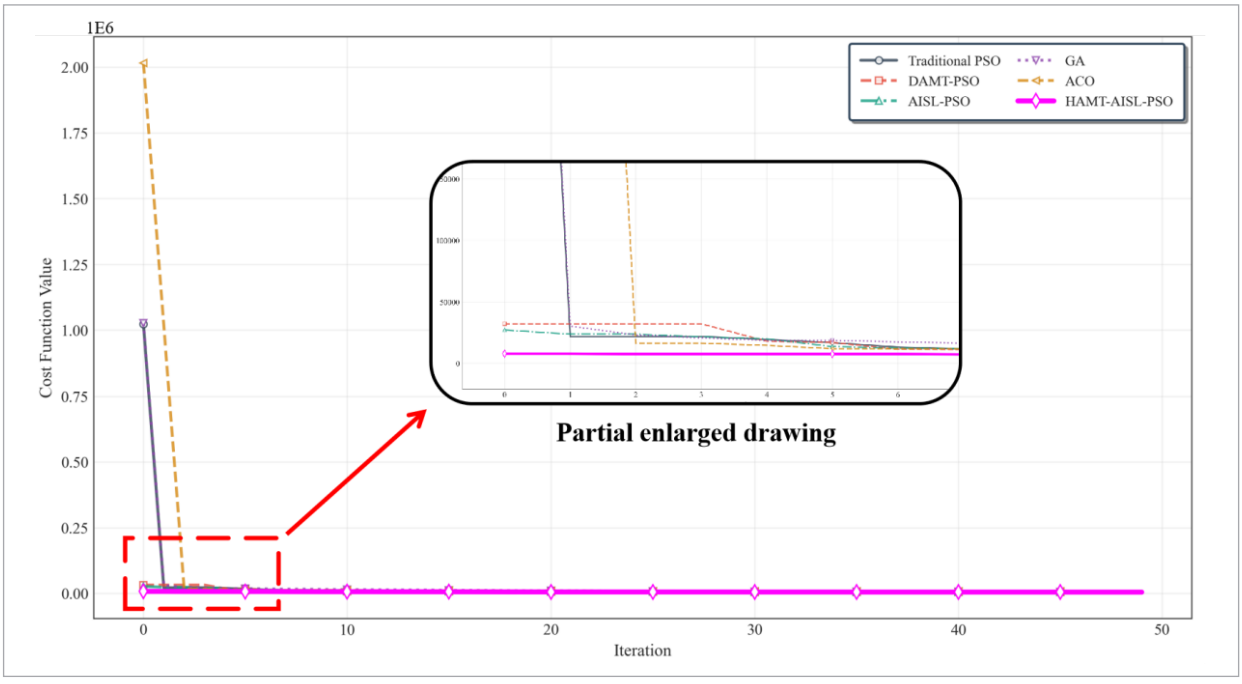


Figure 9
Altitude Profiles.

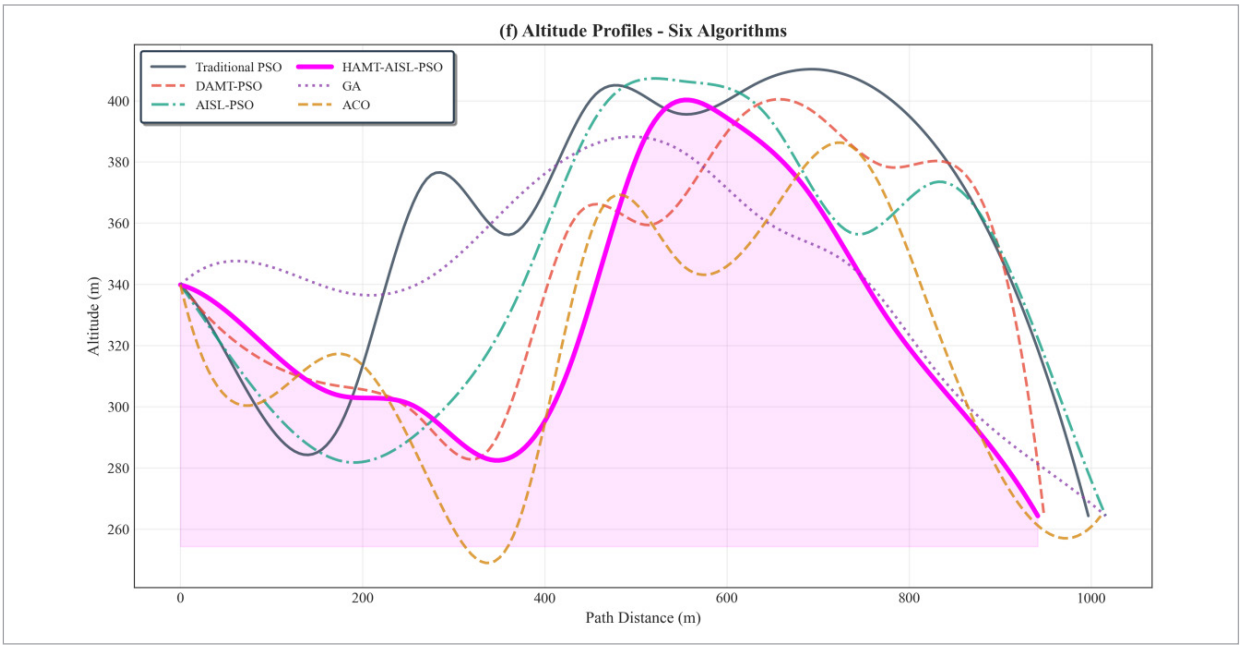


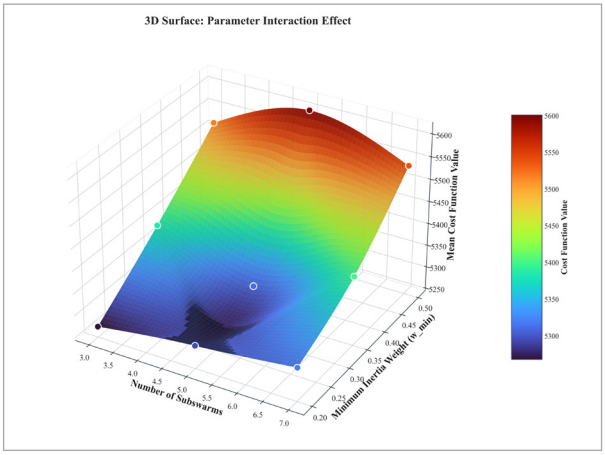
Table 3
Main complexity-related parameters and asymptotic time complexity.

Algorithm	Main iteration settings	Asymptotic time complexity per run
Traditional PSO	MaxIt = 50, nPop = 120	$O(MaxIt \times nPop \times n^2)$
DAMT-PSO	MaxIt = 50, nPop = 120	$O(MaxIt \times nPop \times n^2)$
AISL-PSO	MaxIt = 50, nPop = 150	$O(MaxIt \times nPop \times n^2)$
HAMT-AISL-PSO	MaxIt = 50, nPop = 180	$O(MaxIt \times nPop \times n^2)$
GA	MaxGen = 50, PopSize = 120	$O(MaxGen \times PopSize \times (\log(PopSize) + n^2))$
ACO	MaxIt = 50, nAnt = 80	$O(MaxIt \times nAnt \times n \times n\_discrete^3)$

Specifically, the time complexity for the GA baseline is $O(MaxGen \times (PopSize \times \log(PopSize) + PopSize \times n \times D + PopSize \times T\_cost))$, which simplifies to $O(MaxGen \times PopSize \times n^2)$ if $T\_cost = O(n)$; hence, the asymptotic order of GA equals that of the PSO-based methods, while its actual computational load is slightly higher owing to the sorting and tournament-selection operations, $O(PopSize \times \log(PopSize))$. On the contrary, ACO presents the largest complexity, $O(MaxIt \times nAnt \times n \times n\_discrete^3)$, due to the cubic dependence on the discretization of 3D space. Yet, the proposed HAMT-AISL-PSO eventually secures smaller cost values after fewer effective iterations, so its overall convergence speed remains competitive, and the enhancements in path cost, path length, and stability remain significant under comparable wall-clock runtimes, as depicted from the convergence curves in Figure 8.

From the obtained experimental results, it can be observed that the hybrid HAMT-AISL-PSO algorithm proposed in this paper is significantly better than the comparison algorithm in terms of path planning quality, convergence performance, and algorithm stability. By integrating multiple topological structures and intelligent learning strategies, the algorithm effectively overcomes the shortcomings

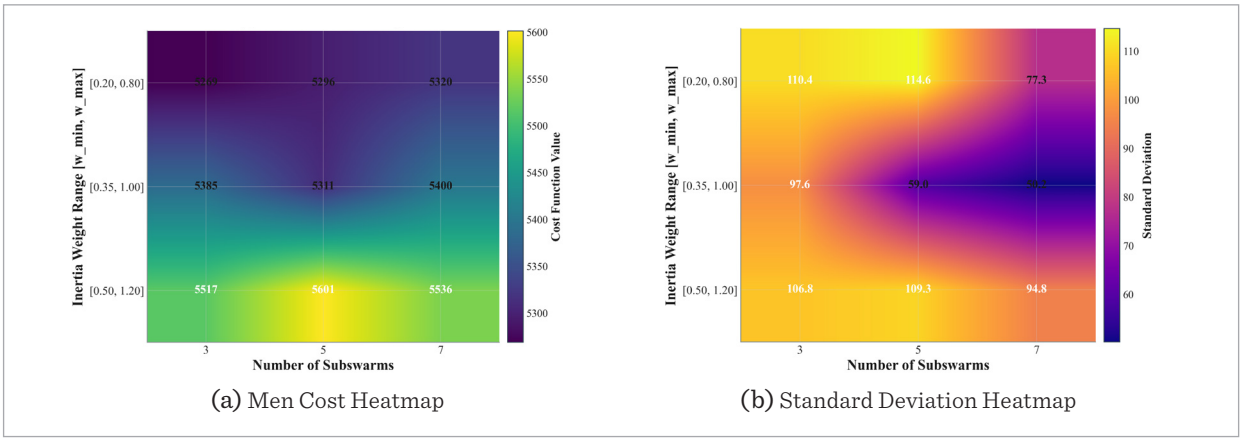
Figure 10
3D Surface: Parameter Interaction Effect.



of traditional particle swarm algorithms, such as easy precocious convergence and low search accuracy. It offers a practical solution to UAV 3D path planning under complex environments. Meanwhile, this algorithm demonstrates a significant advantage in convergence speed, achieving the optimal solution within a relatively small number of iterations, which reflects a dual improvement in its global search ability and optimization efficiency.

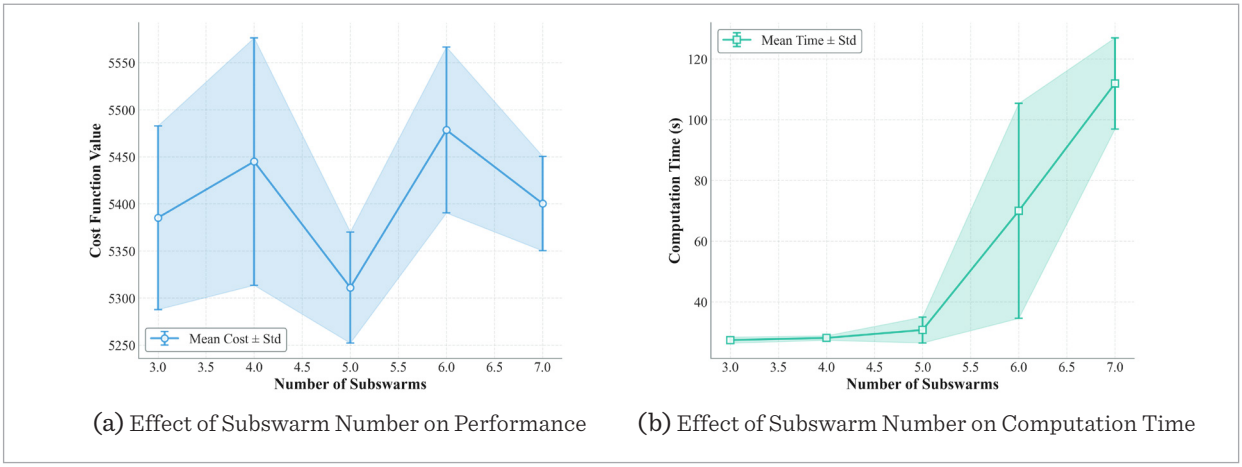
A parameter sensitivity analysis is then conducted for the subswarm number n_subswarms and the inertia-weight range [w_{min}, w_{max}] to evaluate the robustness of HAMT-AISL-PSO. From Figures 10-12, it can be seen that a clear low-cost region appears around $n_{subswarms}$ = 5 with a relatively small w_{min}, while

Figure 11
Heatmaps of the mean cost.



(a) Men Cost Heatmap (b) Standard Deviation Heatmap

Figure 12
Effect of subswarm number.



(a) Effect of Subswarm Number on Performance (b) Effect of Subswarm Number on Computation Time

too few or too many subswarms and extreme inertia weights lead to higher cost and larger variance. As summarized in Figures 13 and 14, $n_{subswarms}$ = 5 and a medium inertia-weight range such as [0.20, 0.80] provide a favourable trade-off: the mean cost and path length are close to the minimum, the standard deviation remains small, and the computation time does not increase sharply. Therefore, this setting is adopted as the default configuration of HAMT-AISL-PSO in the subsequent experiments.

5.3. Comparison Experiment of the Trajectory Tracking Controller

For the verification of the proposed controller's trajectory tracking performance, the above quadcopter UAV control system modules were established by using the modules in MATLAB/Simulink, and the comparison experiments were conducted using both PID and LADRC controllers, respectively. For verifying the controller's performance in complex space missions, the 3D spiral is selected as the reference trajectory, which not only involves continuous circular motion but also has vertical-direction climbing superimposed on it, thus posing higher requirements for the decoupling ability and robustness of the controller. Specifically, the reference trajectory consists of three spiral turns with a radius of 5 meters, while the altitude increases linearly from 3 meters to 10 meters.

Position tracking curves of PID, LQR and LADRC are shown in the three tracking figures of this subsection. All of the three controllers can track the 3D spiral basically. However, at turning segments with large curvature and at height-change sections, the trajectories under PID and LQR have obvious lag and offset. Particularly, at instants when a disturbance occurs, the LQR trajectory has larger oscillation and overshoot, which means weaker disturbance-rejection capability. In contrast, the LADRC trajectories are smooth and closely follow the reference, almost without overshooting, and demonstrate better dynamic response, stability, robustness to trajectory curvature and changes of altitude.

Figure 16 in this subsection plots error curves of PID, LQR and LADRC, respectively. Around the four injected disturbance segments, errors of both PID and LQR increase sharply, and the LQR controller presents the biggest spikes on all three axes, indicating its relatively poor disturbance-rejection ability. The PID

Figure 13
PID position tracking.

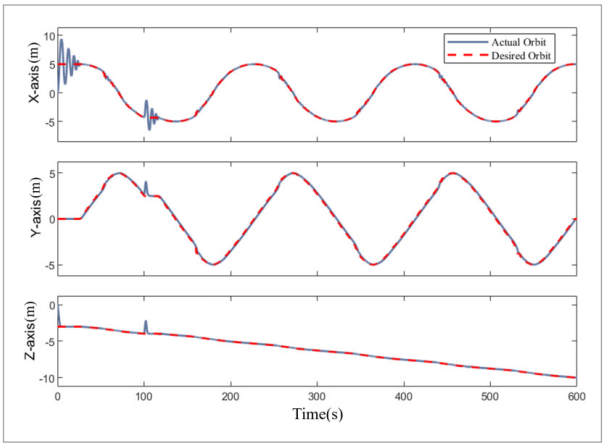


Figure 14
LQR position tracking.

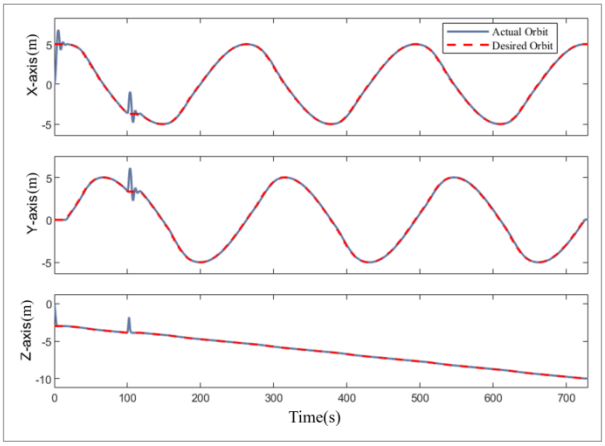


Figure 15
LADRC position tracking.

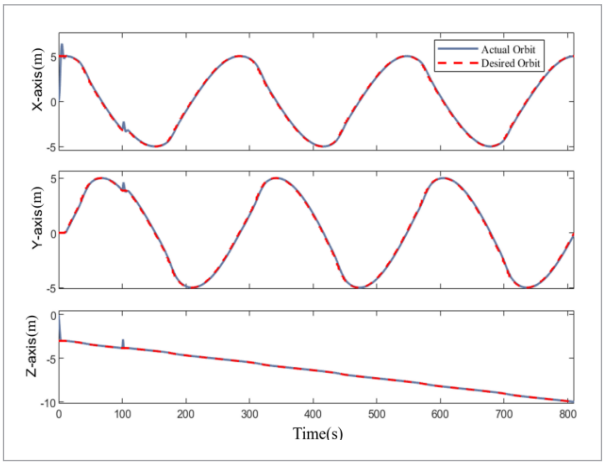
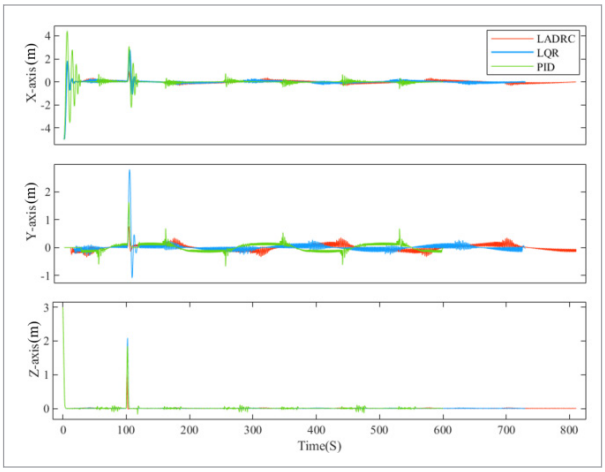


Figure 16
Error curves of PID, LQR and LADRC.



controller also suffers from considerable transient oscillation and non-negligible steady-state error. By contrast, LADRC error stays within a small band and quickly converges to nearly zero after each disturbance, confirming stronger robustness and faster recovery for LADRC under pulse-type disturbances.

Besides PID, LQR and MPC have also been widely used as baseline controllers for quadrotor tracking. It is difficult for LQR to achieve satisfactory performance except for linearized models under nominal conditions; more seriously, its robustness against strong nonlinearities and disturbances is rather poor. Although MPC can handle constraints and model uncertainties to some degree, its on-line optimization results in higher computational cost, possibly unfavorable to a real-time embedded implementation. In contrast, the method adopted in this paper, based on LADRC, maintains a relatively simple structure with strong pulse-disturbance rejection and high tracking accuracy in the considered spiral-trajectory scenario.

Table 4
Mean absolute tracking error of PID, LQR and LADRC in each axis.

Axis	MAE			Improvement Rate	
	PID	LQR	LADRC	LADRC vs PID	LADRC vs LQR
z	0.0255	0.0210	0.0143	43.92%	31.90%
y	0.1065	0.0830	0.0789	25.92%	4.94%
x	0.1551	0.1126	0.1035	33.27%	8.08%

To give a quantitative comparison among the three controllers, the mean absolute tracking errors (MAEs) in each axis are computed over the whole simulation horizon, summarized in Table 4. The MAEs of the PID controller in the z, y and x directions are 0.0255, 0.1065 and 0.1551, respectively, while LADRC reduces them to 0.0143, 0.0789 and 0.1035, corresponding to error-reductions of about 43.92%, 25.92% and 33.27%. For the LQR controller, the MAEs are 0.0210, 0.0830 and 0.1126 on the z, y and x axes, and replacing LQR with LADRC yields further reductions of 31.90%, 4.94% and 8.08%, with an average improvement of about 14.97%. These quantitative indices confirm that LADRC achieves the smallest tracking errors and the best overall accuracy among the three controllers, consistent with the trajectory and error plots.

In summary, in the tracking task of a complex 3D spiral trajectory, the LADRC controller shows significantly better performance advantages than the PID and LQR controllers. Under the designed pulse-disturbance scenario, its higher accuracy, faster convergence speed, and stronger disturbance-rejection capability make it a better candidate for practical deployment in actual flight environments. Overall, considering both planning metrics and tracking metrics, the LADRC delivers the best comprehensive performance in terms of convergence, stability, and robustness.

6. Conclusion

6.1. Research Conclusions

This paper studies the path planning and trajectory tracking of a quadcopter UAV in complex environments, and forms a relatively complete scheme. In the path planning part, the proposed HAMT-AISL-PSO algorithm improves the global search ability and trajectory smoothness. It effectively avoids issues of traditional algorithms, such as being prone to falling into local optimal solutions and generating trajectories with large curvature. In terms of controller design, a trajectory tracking controller based on LADRC is constructed based on the principle of active disturbance rejection control. It can compensate for the system disturbances and uncertainties in real time using the extended state observer, such

that the stability and tracking accuracy of the UAV are guaranteed in a scenario of rapid change. The simulation results, considering applied pulse disturbance torques, are compared with the PID control in this paper. LADRC exhibits more advantages with regard to dynamic response, steady-state error, and disturbance immunity, especially in the stages of turning and changing altitude during its spiraling trajectory.To conclude, the proposed planning and control methods in this paper effectively enhance the autonomous flight performance of UAVs operating in complex environments.

These conclusions are obtained under the modeling assumptions and simulation settings adopted in this study. The quadrotor is modeled as a rigid, symmetric body with constant mass and inertia, actuated only by rotor thrust and gravity, and small-angle approximations are used in the attitude kinematics. Path planning and obstacle avoidance are evaluated in a single-UAV, static three-dimensional environment with hemispherical and cylindrical obstacles, while tracking control is validated on a 3D spiral reference trajectory with pulse-type moment disturbances and ideal sensors and actuators [19]. Therefore, the proposed framework is mainly applicable to medium-scale multirotor platforms performing missions in structured but cluttered environments under bounded disturbances. Its performance in scenarios with strong aerodynamic coupling, dynamic or densely time-varying obstacles, significant sensor noise, or large modeling errors has not yet been verified and thus remains a limitation of the present work.

6.2. Future Research Directions

Although the study in this paper verifies the effectiveness of the proposed method, there is still room for further development. First, more real-time optimization mechanisms can be incorporated into the path planning algorithm, for example by designing incremental or receding-horizon variants of the proposed HAMT-AISL-PSO planner, so as to better meet the rapid decision-making requirements of UAVs in highly dynamic environments [5]. Second, an important research direction is to extend the HAMT-AISL-PSO–based path planner and the LADRC-based trajectory-tracking controller to multi-UAV cooperative and formation-control scenarios. In this context, the algorithm needs to be adapted to handle inter-UAV collision avoidance, communication and sensing constraints, and dynamic task allocation, so that safe and coordinated 3D paths can be generated for multiple quadrotors while maintaining the robustness and tracking performance demonstrated in the single-UAV case. Finally, flight experiments on real quadrotor platforms, possibly combined with hardware-in-the-loop simulations, are needed to further evaluate the reliability and engineering applicability of the proposed framework in real environments.

Acknowledgement

This work was supported by the ”College Students’ Innovative Entrepreneurial Training Plan Program” of China under the project titled ”A Driver Monitoring System Based on Human Posture Assessment”(202510700033).

References

1. Bingul, Z., Karahan, O. Real-Time Trajectory Tracking Control of Stewart Platform Using Fractional Order Fuzzy PID Controller Optimized by Particle Swarm Algorithm. Industrial Robot: The International Journal of Robotics Research and Application, 2022, 49(4), 708-725. https://doi.org/10.1108/IR-07-2021-0157
2. Cavanini, L., Ippoliti, G., Camacho, E. F. Model Predictive Control for a Linear Parameter Varying Model of an UAV. Journal of Intelligent & Robotic Systems, 2021, 101(3), 57. https://doi.org/10.1007/s10846-021-01337-x
3. Chen, C. C., Chen, Y. T. Feedback Linearized Optimal Control Design for Quadrotor with Multi-Performances. IEEE Access, 2021, 9, 26674-26695. https://doi.org/10.1109/ACCESS.2021.3057378
4. Chen, Z. W., Zhang, Z. Z., Cao, Y. J. Fal Function Improvement of ADRC and Its Application in Quadrotor Aircraft Attitude Control. Control and Decision, 2018, 33(10), 1901-1907.
5. Cheng, Z., Wang, E., Tang, Y., Wang, Y. Real-Time Path Planning Strategy for UAV Based on Improved Particle Swarm Optimization. Journal

of Computers, 2014, 9(1), 209-214. https://doi.org/10.4304/jcp.9.1.209-214

6. Elmeseiry, N., Alshaer, N., Ismail, T. A Detailed Survey and Future Directions of Unmanned Aerial Vehicles (UAVs) with Potential Applications. Aerospace, 2021, 8(12), 363. https://doi.org/10.3390/aerospace8120363
7. Gad, A. G. Particle Swarm Optimization Algorithm and Its Applications: A Systematic Review. Archives of Computational Methods in Engineering, 2022, 29(5). https://doi.org/10.1007/s11831-021-09694-4
8. Golabi, M., Ghambari, S., Lepagnot, J., Jourdan, J., Brevilliers, M., Idoumghar, L. Bypassing or Flying Above the Obstacles? A Novel Multi-Objective UAV Path Planning Problem. 2020 IEEE Congress on Evolutionary Computation (CEC). IEEE, 2020, 1-8. https://doi.org/10.1109/CEC48606.2020.9185695
9. Guan, Z., Zihao, F., Yingxin, H., Jintao, C., Jinyu, R. An Adaptive Strategy Quantum Particle Swarm Optimization Method Based on Intuitionistic Fuzzy Entropy and Evolutionary Game Theory. Applied Soft Computing, 2025, 113654. https://doi.org/10.1016/j.asoc.2025.113654
10. Ibraheem, I. K., Bdul-Adheem, W. R. A Novel Second-Order Non-Linear Differentiator with Application to Active Disturbance Rejection Control. 2018 1st International Scientific Conference of Engineering Sciences-3rd Scientific Conference of Engineering Science (ISCES). IEEE, 2018, 68-73. https://doi.org/10.1109/ISCES.2018.8340530
11. Ishihara, Y., Hazama, Y., Suzuki, K., Yokono, J. J., Sabe, K., Kawamoto, K. Improving Wind Resistance Performance of Cascaded PID Controlled Quadcopters Using Residual Reinforcement Learning. arXiv Preprint arXiv:2308.01648, 2023.
12. Kothari, M., Postlethwaite, I. A Probabilistically Robust Path Planning Algorithm for UAVs Using Rapidly-Exploring Random Trees. Journal of Intelligent & Robotic Systems, 2013, 71(2), 231-253. https://doi.org/10.1007/s10846-012-9776-4
13. Kurak, S., Hodzic, M. Control and Estimation of a Quadcopter Dynamical Model. Periodicals of Engineering and Natural Sciences, 2018, 6(1), 63-75. https://doi.org/10.21533/pen.v6i1.164
14. Li, A., Hansen, M., Zou, B. Traffic Management and Resource Allocation for UAV-Based Parcel Delivery in Low-Altitude Urban Space. Transportation Research Part C: Emerging Technologies, 2022, 143, 103808. https://doi.org/10.1016/j.trc.2022.103808
15. Li, K., Wang, G., Bai, Y. A Second-Order LADRC-Based Control Strategy for Quadrotor UAVs Using a Modified Crayfish Optimization Algorithm and Fuzzy Logic. Electronics, 2025, 14(15), 3124. https://doi.org/10.3390/electronics14153124
16. Lopez-Sanchez, I., Moreno-Valenzuela, J. PID Control of Quadrotor UAVs: A Survey. Annual Reviews in Control, 2023, 56, 100900. https://doi.org/10.1016/j.arcontrol.2023.100900
17. Lu, K., Xu, R., Li, J., Lv, Y., Lin, H., Liu, Y. A Vision-Based Detection and Spatial Localization Scheme for Forest Fire Inspection from UAV. Forests, 2022, 13(3), 383. https://doi.org/10.3390/f13030383
18. Ma, H. Graph-Based Multi-Robot Path Finding and Planning. Current Robotics Reports, 2022, 3(3), 77-84. https://doi.org/10.1007/s43154-022-00083-8
19. Meng, Q., Chen, K., Qu, Q. PPSwarm: Multi-UAV Path Planning Based on Hybrid PSO in Complex Scenarios. Drones, 2024, 8(5), 192. https://doi.org/10.3390/drones8050192
20. Najm, A. A., Ibraheem, I. K. Non-Linear PID Controller Design for a 6-DOF UAV Quadrotor System. Engineering Science and Technology, an International Journal, 2019, 22(4), 1087-1097. https://doi.org/10.1016/j.jestch.2019.02.005
21. Oersted, H., Ma, Y. Review of PID Controller Applications for UAVs. arXiv Preprint arXiv:2311.06809, 2023.
22. Phung, M. D., Ha, Q. P. Safety-Enhanced UAV Path Planning with Spherical Vector-Based Particle Swarm Optimization. Applied Soft Computing, 2021, 107, 107376. https://doi.org/10.1016/j.asoc.2021.107376
23. Saccani, D., Cecchin, L., Fagiano, L. Multitrajectory Model Predictive Control for Safe UAV Navigation in an Unknown Environment. IEEE Transactions on Control Systems Technology,

2022, 31(5), 1982-1997. https://doi.org/10.1109/TCST.2022.3216989

24. Sengupta, S., Basak, S., Peters, R. A. Particle Swarm Optimization: A Survey of Historical and Recent Developments with Hybridization Perspectives. Machine Learning and Knowledge Extraction, 2018, 1(1), 157-191. https://doi.org/10.3390/make1010010
25. Song, J., Hu, Y., Su, J., Zhao, M., Ai, S. Fractional-Order Linear Active Disturbance Rejection Control Design and Optimization Based Improved Sparrow Search Algorithm for Quadrotor UAV with System Uncertainties and External Disturbance. Drones, 2022, 6(9), 229. https://doi.org/10.3390/drones6090229
26. Sun, W., Sun, P., Ding, W., Zhao, J., Li, Y. Gradient-Based Autonomous Obstacle Avoidance Trajectory Planning for B-Spline UAVs. Scientific Reports, 2024, 14(1), 14458. https://doi.org/10.1038/s41598-024-65463-w
27. Wang, H., Qi, X., Lou, S., Jing, J., He, H., Liu, W. An Efficient and Robust Improved A* Algorithm for Path Planning. Symmetry, 2021, 13(11), 2213. https://doi.org/10.3390/sym13112213
28. Xu, L., Tian, B., Wang, C., Lu, J., Wang, D., Li, Z., Zong, Q. Fixed-Time Disturbance Observer-Based MPC Robust Trajectory Tracking Control of Quadrotor. IEEE/ASME Transactions on Mechatronics, 2024. https://doi.org/10.1109/TMECH.2024.3503062
29. Yu, X., Li, C., Zhou, J. F. A Constrained Differential Evolution Algorithm to Solve UAV Path Planning in Disaster Scenarios. Knowledge-Based Systems, 2020, 204, 106209. https://doi.org/10.1016/j.knosys.2020.106209
30. Yuan, S., Li, Y., Bao, F., Xu, H., Yang, Y., Yan, Q., Zhong, S., Yin, H., Xu, J., Huang, Z., Lin, J. Marine Environmental Monitoring with Unmanned Vehicle Platforms: Present Applications and Future Prospects. Science of The Total Environment, 2023, 858, 159741. https://doi.org/10.1016/j.scitotenv.2022.159741
31. Zhang, Q., Fu, M., Zhai, C., Wang, S., Ning, K., Wang, M. Incremental Non-Linear Dynamic Inversion Control for Quadrotor UAV with an Angular Accelerometer. 2023 42nd Chinese Control Conference (CCC). IEEE, 2023, 657-662. https://doi.org/10.23919/CCC58697.2023.10240476