A New Method of Solving UAV Trajectory Planning Under Obstacles and Multi-constraint

SHIKAI SHAO, WEILONG SHI, YUANJIE ZHAO, AND YUN DU
School of Electrical Engineering, Hebei University of Science and Technology, Shijiazhuang 050018, China

Corresponding author: Shikai Shao (e-mail: kdsks@126.com).

This work was supported in part by the National Natural Science Foundation of China under Grant 61903122, in part by Natural Science Foundation of Hebei Province under Grant F20211208015, in part by the Science and Technology Project of Hebei Education Department under Grant BJ2021003, and in part by the National Natural Science Foundation of China under Grant U20A20198.

ABSTRACT Multi-constraint trajectory planning for unmanned aerial vehicles (UAVs) has been widely used in military and civil fields. The existing path planning methods, such as swarm intelligence algorithm and graph-based algorithm, cannot incorporate the flying time and UAV kinematic model into evolution. To overcome such disadvantage, a method of solving trajectory planning under obstacles and multi-constraint is investigated in this paper. Firstly, the flying time is discretized as a certain number of Chebyshev points which are the optimized moments of control variable, and they can reduce the computational burden. The process of solution is divided into multi-phase, i.e., two points as a phase to generate the trajectory. Then, angular velocity is taken as control variable, and function of angular velocity is solved by cubic spline interpolation. Besides, functions of angle and position are obtained by integration. The results are substituted into the model consisted by particle swarm optimization (PSO) and the UAV kinematic model to optimize. On this basis, the angular velocity, angle and position are calculated according to the allocated moments. Finally, Monte-Carlo simulation and comparison with existed method are carried out in obstacle environment. All the results illustrate that the multi-phase method can calculate the kinematic parameters of UAV accurately and plan smooth trajectories. Meanwhile, the proposed method is easier to meet the complicated constraints than the single-phase method. In addition, the dimensionality of solution is also enriched effectively.

INDEX TERMS Unmanned aerial vehicles, trajectory planning, kinematic model, particles warm optimization, cubic spline interpolation

I. INTRODUCTION

Unmanned system is becoming more common as it can complete complicated missions without human intervention. Unmanned aerial vehicle (UAV) is an aircraft without human pilot onboard. With the development of technology, UAV has been widely applied to both military and civilian tasks, such as surveillance, reconnaissance, rescue and commercial performance [1]–[3]. As one of the important directions of application, UAV trajectory planning has attracted much attention. Besides, UAV with trajectory planning capability can guarantee its own safety and reduce the requirements and dependence on professional operators. When UAV carries out mission, trajectory planning system can provide route guidance for flight system, which is important for UAV to realize autonomous flight and to enhance the autonomy level.

In general, path or trajectory planning of UAVs can be regarded as an optimization problem which aims to find an optimal or near-optimal solution from the starting position to the destination position under certain criteria and mission constraints. The main challenge is to avoid collisions with environmental obstacles and other members and to provide a capable guidance for the tracking control system. In [4], a newest review on path planning of unmanned robot system was presented and some categories about existing results have been made. For the path planning issue, the most used methods and algorithms include Voronoi diagram algorithm [5], A* algorithm [6], probabilistic roadmaps algorithm [7], rapidly-exploring random trees based algorithm [8], and so on. But the UAV kinematic model and dynamic model are seldom considered, thus they are usually not appropriate for practical situations. The field-based method [9] and fluid based method [10] are another two candidate methods for path planning. But they are easy to trap into local minimum and sometimes no feasible path could be
generated when the target and obstacles are near. In the past two years, the machine learning technique was also utilized for UAV trajectory planning [11], [12], but the kinematic model of UAV was also not considered in the calculation process. In addition, all the aforementioned methods can only provide path information other than trajectory information, which means the flying time is usually not involved and the results cannot be tracked directly by control system.

For UAV trajectory planning problem, it is also usually modeled as a nonlinear optimal control (NOC) problem with multiple state and control constraints. Many gradient based numerical methods have been adopted to solve the problem [13], [14]. Generally, these methods are mainly classified into two branches: indirect method and direct method [15]. Both methods attempt to minimize objective function and constraint violations by using discrete approximations. But, there are still some limitations. When the dimension of the problem and the complexity are large, the numerical method will have problems such as long running-time, slow convergence rate and even inability to obtain the feasible solution. In [16], a Chebyshev pseudo-spectral method (GPM) was introduced, and thereby trajectory optimization for hypersonic glide vehicle with flexible initial conditions was addressed. GPM is able to maintain relative faster convergence and accurate solutions during the optimization process. However pseudo-spectral method also inherent the drawbacks of sensitive to initial values and lack of a fast global searching capability.

Besides numerical methods, the swarm intelligence algorithms, such as ant colony optimization (ACO), particle swarm optimization (PSO) and plant growth algorithm, have also been investigated for path planning by researchers [17]–[19]. Among the existing swarm intelligence algorithms, PSO maintains the advantages of simple implementation and being able to access the optimal or sub-optimal solution, thus it is often utilized for path planning and other optimization tasks. In [20]–[21], a comprehensive analysis on the factors that can affect the performance of PSO was carried out and it was pointed out that parameter selection, topology structure and combination with other methods were the main aspects to improve the efficiency and feasibility of PSO. In [22], PSO was adopted to solve optimal scheduling for a swarm robotic system. In [23], the dynamic hybrid PSO combined with the trim-maneuver library technology was proposed for UAV motion planning. In [24], the authors considered the drawback of basic PSO that some high-quality waypoints in previous candidate paths might not be exploited for further evolution, and proposed an idea of separately evaluating and evolving waypoints for two dimensional UAV path planning. Similar idea was extended by [25], where the authors introduced a strategy about optimal path competition strategy between the single waypoint selection and the integrated waypoints selection, and completed a three dimensional UAV path planning demonstration. In order to overcome the defects of local minimum and premature of PSO, the authors in [26] proposed an improved PSO combining with an adaptive decision operator for three dimensional UAV path planning, and results illustrated the advantages of the proposed strategy. Apart from the utilization in UAV path planning, some researchers have also focused on PSO theoretical analysis and improvements.

Taking both advantages of swarm intelligence algorithm and numerical methods, it may provide a meaningful and effective solution for UAV trajectory planning by involving UAV kinematic model with swarm intelligence algorithm. The newest results on robot trajectory planning also enriched the relevant research. In [27], graph-based multi-agent path planner was involved with nonlinear optimization to generate smooth trajectories for non-holonomic mobile robots. The resulting trajectories indicated the superiority of hybrid optimization. In [28], in order to minimize energy consumption and running time while ensuring the smoothness of collaborative welding robot with multiple manipulators, a methodology for optimal trajectory planning was proposed. The trajectory was interpolated in each joint space by means of cubic B-spline and motion time nodes were optimized based on particle swarm optimization (PSO) algorithm. In [29], the authors designed a time-optimal trajectory planning method for the manipulator by searching the optimal path simultaneously, and they proposed a cubic uniform B-spline interpolation algorithm to derive the motion curve expression of the manipulator joint with unknown path points. In [30], an adaptive cuckoo search (ACS) algorithm with high efficiency and excellent stability was proposed to minimize the total motion time under strict dynamic constraints. Depending on the key nodes of the joint angles calculated by a hybrid inverse kinematics algorithm, the continuous quantic B-spline curve algorithm was utilized for planning smooth trajectories of the feeding motion from peer to peer. But, these literatures all focus on trajectory planning for robot system. Besides, the curve fitted by B-spline does not pass through the data points, which will affect the accuracy of data.

Cubic polynomial interpolation trajectory planning is widely used due to its simplicity in calculation, but trajectory planning based on polynomial interpolation is difficult to optimize under traditional methodology due to its high order and lack of convex hull. Using intelligent algorithms to achieve time optimal trajectory planning is a worthwhile research objective to this effect. In [31], the authors introduced a new idea that polynomial interpolation function was combined with improved cuckoo search algorithm to generate time-optimal trajectory for UR robot. In [32]–[34] a hybrid approach of control parametrization and time discretization (CPTD) and PSO was proposed to solve the optimization problem.
A. PROBLEM FORMULATION

The target strike mission specified in this paper is defined as: UAV flies with multi-constraint to search for the target point quickly, and it can avoid the obstacles in the process of flying. Once the distance between UAV and target point is smaller than the threshold, the target will be hit, then multi-phase method is adopted to compute the trajectory, head for the target. According to the above description, the trajectory planning problem can be shown in Fig. 1.

According to the computing analysis, the position and angle information can provide instructions for control system, which can generate feasible trajectory for actual flight.

B. KINEMATIC MODEL OF UAV

In general, the UAV is regarded as a particle in trajectory planning problem, and a particle motion model is used. In some literature, only the kinematic model is considered [36]. This is because that compared to control module, trajectory planning module is an outer loop to provide the important command signal, which requires a fast computation speed. For a rigid body, the motion model which includes a set of nonlinear differential equations with variable coefficients is used. It will cost much time to solve such complicated equations. Thus, in order to further save the computing time, a simplified UAV kinematic model is introduced as follows [35]

\[
\begin{align*}
\dot{x} &= V \cos \theta \cos \varphi \\
\dot{y} &= V \cos \theta \sin \varphi \\
\dot{z} &= V \sin \theta \\
\dot{\theta} &= q \\
\dot{\varphi} &= r
\end{align*}
\]

where (1) denotes the UAV kinematic model. \(x, y, z, V\) are position and speed of UAV. \(q, r\) are trajectory inclination velocity and heading angular velocity respectively. \(\theta, \varphi\) are trajectory inclination and heading.
angle. Therefore, the control variable can be defined as 
\[ u = [q \, r]^T \], and the state variable is designed as 
\[ x = [x_1 \, y \, z \, V \, \theta \, \phi]^T \]. The motion performance parameters of UAV are calculated according to (1) after control variable is determined.

III. PSO AND DESIGN OF FITNESS FUNCTION

In the following, the formula of PSO is firstly introduced in detail. Then, the design of fitness function is discussed and analyzed.

A. DESCRIPTION OF FORMULA

PSO is a population-based optimization algorithm. It starts from random initializations with candidate solutions and could find an optimal or near-optimal solution with particle position and velocity updating [20]. For each particle, the current velocity represents the searching direction and is dynamically updated by its previous value, the personal best position and the global best position. For trajectory planning problem, each particle corresponds to a possible trajectory and is usually consisted by a series of collocation points. In this paper, the number of collocation points is set as \( n \). The number of total particles, namely the swarm size, is defined as \( N \).

Then the position and velocity vector for the \( i \)th particle can be written as

\[
\begin{align*}
v_i^{t+1} & = \omega v_i^t + c_1 r_1 (p_b^t - x_i^t) + c_2 r_2 (p_g^t - x_i^t) \quad (2) \\
x_i^{t+1} & = x_i^t + v_i^{t+1} \quad (3)
\end{align*}
\]

where \( t \in 1...T \), is the current iteration with \( T \) denoting the total iterations. \( \omega \) is the inertia weight for balancing global and local searching, and \( c_1, c_2 \) are acceleration coefficients reflecting the ability of learning from local best particle and personal best particles. \( p_b \) is personal best value, \( p_g \) is global best value, and \( r_1, r_2 \) are two random values which is distributed in \( (0,1) \). In order to accelerate the convergence of algorithm, some researchers have adopted linear-varying inertia weight, which is set as [36]

\[
\omega_i = \omega_{\text{max}} - \frac{(\omega_{\text{max}} - \omega_{\text{min}})t}{T} \quad (4)
\]

where \( \omega_{\text{max}} \) and \( \omega_{\text{min}} \) are defined as maximum value and minimum value of inertia weight respectively, and the values are \( \omega_{\text{min}} = 0.4 \) and \( \omega_{\text{max}} = 0.9 \) in this paper.

B. CHAOS-BASED PARTICLE INITIALIZATION

As is well known, PSO is initialized with a population of random distributed solutions, and the distribution quality of particle initialization plays a critical role in the searching for optimal solution. Thus, in order to enrich the diversity of particle initialization in the process of UAV trajectory planning, the Logistic map, which is one of the simplest and the most widely used maps, is introduced clearly as follows [20]

\[
x_{p+1} = \mu x_p (1 - x_p) \quad (5)
\]

where \( \mu \) is defined as the bifurcation coefficient, \( x_p \) is set as the \( p \)th chaotic variable to calculation. When \( \mu \in [3.5, 4] \), the whole system gradually evolves into the chaotic state. When \( \mu = 4, x_0 \in (0, 1) \), the proposed system will produce chaotic signal \( x_p \in (0, 1) \) uniformly, which will be utilized for particle initialization.

C. ADAPTIVE PARAMETER ADJUSTMENT

Apart from inertia weight, acceleration coefficients \( c_1, c_2 \) are another two critical parameters. Usually, they indicate the weight of stochastic acceleration terms that pull each particle to local best positions and global best positions, and play important roles in adjusting the convergence speed and searching direction. Because of the randomness which is reflected in the searching process, it is difficult to carry out precise quantitative analysis between coefficient values and iterative times. However, we can analyze and discuss the mapping relationship between them in a qualitative way. For PSO evolution, searching and convergence are the two main objectives during the process. Searching means each particle should always enrich the diversity of the swarm and convergence means the algorithm should finally achieve a same value that is the global optimal solution. Theoretically, it is reasonable to conclude that searching process should predominate at the evolution at the first half stage while convergence process should predominate at the last half stage. Therefore, the proposed acceleration coefficients are adaptively designed as [36]

\[
c_1 = c_{\text{min}} + \frac{(c_{\text{max}} - c_{\text{min}})t}{T} \quad (6)
\]

\[
c_2 = c_{\text{max}} + \frac{(c_{\text{max}} - c_{\text{min}})t}{T} \quad (7)
\]

where \( T \) is the total iterative times and \( t \) is the current iterative time, \( c_{\text{min}} \) and \( c_{\text{max}} \) are constant values with \( c_{\text{max}} > c_{\text{min}} > 0 \). In the simulation, \( c_1 \) and \( c_2 \) for conventional PSO are set as same constant value, while \( c_{\text{max}} \), \( c_{\text{min}} \) for the proposed PSO are set according to the principle \( (c_{\text{max}} + c_{\text{min}}) / 2 = c_1^e = c_2^e \), where \( c_1^e = c_2^e \) are usually set as 2. In this paper, we set \( c_{\text{max}} = 3.5 \), \( c_{\text{min}} = 0.5 \). This principle guarantees that the average \( c_1 \) and \( c_2 \) for the proposed PSO are the same as those for conventional PSO, and makes the simulation comparisons much more fair and well-founded.

It is obvious that \( c_1 \) and \( c_2 \) are respectively linearly decreasing and linearly increasing, and the sum satisfies
\( c_{\text{max}} + c_{\text{min}} = c_1 + c_2 \) meaning that the ability for particle searching and convergence is constant. For Eqs. (6) ~ (7), it is satisfied that \( c_1 > c_2 \) at the first half stage, thus the searching diversity is strengthened, which would be benefit for fast obtaining global optimal solution and avoiding local minimum. \( c_1 < c_2 \) is satisfied at the last half stage and has the benefit of fast convergence to the same global optimal solution. In a word, the adaptive coefficients would improve the optimality and rapidity of PSO.

**D. CONSTRAINTS**

To achieve of goal of cooperative trajectory planning, several constraints should be taken into consideration. They mainly include performance constraint, minimal flying length of trajectory, terminal constraint, mountain terrain constraint, and radar threat constraint. The details are given in the following. In this paper, the collision avoidance is reflected by the mountain constraint and radar constraint. When the trajectory points pass through mountain or radar, the fitness function value will be given a large penalty value. Therefore, the points passing through the mountain or radar should be eliminated.

1) PERFORMANCE CONSTRAINT

Maneuverability is an important index to reflect the ability of position and motion direction for a vehicle. The constraints on the maneuverability of vehicle must be met, or the obtained trajectory is unfeasible and useless. The goal for UAV trajectory planning is to find a feasible and optimal trajectory to the target under complex environment and state constraints. Performance constraints mean that the performance of trajectory planning should meet engineering requirements. Generally, the performance constraints mainly include the position constraint, angle constraint and angular velocity constraint, which are formulated as

\[
\begin{align*}
x_{\text{min}} & \leq x \leq x_{\text{max}} \\
y_{\text{min}} & \leq y \leq y_{\text{max}} \\
z_{\text{min}} & \leq z \leq z_{\text{max}} \\
q & \leq q_{\text{max}}, \ r \leq r_{\text{max}} \\
\theta & \leq \theta_{\text{max}}, \ \varphi \leq \varphi_{\text{max}}
\end{align*}
\]

The corresponding fitness function can be expressed as

\[
f_x = \sum_{i=1}^{n} \sum_{j=1}^{\varphi} A_{ij} = \begin{cases} P_{ij}^c & \text{if } d_{ij}^p > |d_{ij}'| \\ 0 & \text{if } d_{ij}^p \leq |d_{ij}'| \end{cases}
\]

where \( x_{\text{min}}, x_{\text{max}}, y_{\text{min}}, y_{\text{max}}, z_{\text{min}}, z_{\text{max}} \) are the lower and upper bounds for position coordinates of UAV respectively, \( q_{\text{max}}, r_{\text{max}}, \theta_{\text{max}}, \varphi_{\text{max}} \) are set as the boundaries of angular velocity, and \( \theta_{\text{max}}, \varphi_{\text{max}} \) are set as the boundary values of angle. Besides, \( P_{ij}^c (j = x, y, z, q, r, \theta, \varphi) \) is the cost value of performance parameter corresponding to \( i \)th trajectory point, for example, \( P_{ix}^c \) is regard as the cost value of the \( x \)-axis constraint corresponding to the first trajectory point, and the specific value is determined by the scenes of simulation. \( f_x \) is set as the total cost of performance constraints. \( A_{ij} \) is set as the cost value after careful comparison and judgment. \( d_{ij}^p \) represents the actual numerical value of the performance parameter corresponding to the \( i \)th trajectory point, thus \( d_{ix}^p \) represents the value of \( x \)-axis coordinate corresponding to the first trajectory point. \( d_{ij}' \) is defined as the threshold value of the performance parameter corresponding to the \( j \)th trajectory point.

2) MINIMAL FLYING LENGTH OF TRAJECTORY

For flying missions, shorter trajectory is always preferred than longer trajectory length, because shorter trajectories usually result in less fuel consumption and can reduce the probability of encountering an unexpected threat, which is given as follows

\[
f_L = \sum_{i=2}^{n} \sqrt{(x_i - x_{i-1})^2 + (y_i - y_{i-1})^2 + (z_i - z_{i-1})^2}
\]

(11)

where \( n \) represents the number of total collocation points in this paper, mainly including the starting waypoint \((x_1, y_1, z_1)\), target waypoint and the designed waypoints \((x_i, y_i, z_i)\) with \( i = 2, \ldots, n-1 \).

3) TERMINAL CONSTRAINT

As is well known, the purpose of UAV trajectory planning is to plan a trajectory, reach the destination, and avoid the obstacles successfully in the process of forward. On this basis, the trajectory of calculation can appropriately provide guidance and instruction for actual situation. Thus, in order to achieve the desired effect of flight, and get the feasible trajectory, UAV should hit the target at the end of the moment. When the distance between UAV and target is less than a small value, the target is considered to have been attacked. It is expressed as follows

\[
\begin{align*}
x_i &= |x_{\text{end}} - x_{\text{tar}}| \\
y_i &= |y_{\text{end}} - y_{\text{tar}}| \\
z_i &= |z_{\text{end}} - z_{\text{tar}}| \\
d_{w-t} &= \sqrt{x_i^2 + y_i^2 + z_i^2}
\end{align*}
\]

(12)

The corresponding fitness function can be expressed as
where \((x_{\text{end}}, y_{\text{end}}, z_{\text{end}})\) represents the position coordinate of UAV and \((x_{\text{tar}}, y_{\text{tar}}, z_{\text{tar}})\) is the position of target. \(d_{w-i}\) is the distance between end position and target, \(d_f\) is the distance threshold between end position and target. \(f_k\) is the terminal constraint. \(P^T\) is set as the cost value after comparison and judgement, and the specific value is determined by the simulation.

4) MOUNTAIN TERRAIN CONSTRAINT

For UAV safe flying, the planned trajectory must avoid all the mountains in 3-D environment. In common applications, accurate and updatable terrain maps and terrain constraints can be known in advance. In this paper, we suppose mountains as the main terrain feature and UAV cannot fly through mountains. The simplified model of mountains is described as [25]

\[
z_e^m = h_e \ast \exp \left( \frac{-(x_e^m - x_e^e)^2}{x_e^k} - \frac{(y_e^m - y_e^e)^2}{y_e} \right)
\]

where \(x_e^m, y_e^m, z_e^m\) are set as the three-dimensional coordinate of the \(i\)th mountain, \((x_e^e, y_e^e)\) is the horizontal center of the mountain, and \(h_e, x_e, y_e\) are the designed parameters, which can affect directly the height and slope of a mountain. The corresponding fitness function can be expressed as

\[
f_m = \sum_{i=1}^{K} \sum_{k=1}^{E} C_{ik} \]

\[
C_{ik} = \begin{cases} P_e^{M} & \text{if } z_i \leq H_e^{M} \\ 0 & \text{if } z_i > H_e^{M} \end{cases}
\]

\[
H_e^{M} = h_e \ast \exp \left( \frac{-(x_e - x_e^m)^2}{x_e^k} - \frac{(y_e - y_e^m)^2}{y_e} \right)
\]

where \(f_m\) is the total cost of the mountain constraints. \(P_e^{M} (e = 1, 2, ..., E)\) is the threat cost value of the \(e\)th mountain corresponding to \(i\)th collocation point, and the specific value depends on the simulation. \(H_e^{M}\) is the height reference value of the \(i\)th mountain, \(C_{ik}\) is the cost value after comparison, \((x_i, y_i, z_i)\) is the position coordinate of \(i\)th collocation point.

5) RADAR THREAT COST

For a flying UAV under penetration condition, it may face high risk threat. In this paper, the radar is regard as the risk threat. The mathematical model of the threat is set as a geometric sphere. The simplified mathematical model for a radar threat is introduced as

\[
d_{ik}^R = \sqrt{(x_i - x'_e)^2 + (y_i - y'_e)^2 + (z_i - z'_e)^2}
\]

The corresponding fitness function can be expressed as:

\[
f_R = \sum_{i=1}^{K} \sum_{k=1}^{E} B_{ik}
\]

\[
B_{ik} = \begin{cases} P^R & \text{if } d_{ik}^R \leq R_k \\ 0 & \text{if } d_{ik}^R > R_k \end{cases}
\]

where \((x_i, y_i, z_i)\) is set as the position coordinate of the \(i\)th collocation point. \(P^R (k = 1, 2, ..., K)\) is set as the cost value of the \(k\)th radar corresponding to \(i\)th collocation point, and the specific value is determined by the simulation. \(B_{ik}\) is the cost value after comparison and judgement. \(d_{ik}^R\) is the distance between the collocation point and the radar center, and \(f_k\) is the total cost of radar constraint.

E. OBJECTIVE

To establish clearly an optimization problem, the objective for UAV trajectory planning should also be determined and designed. In this paper, length of trajectory, performance parameter and collision avoidance are considered in fitness function. The specific objective function about trajectory planning can be written as follows

\[
f = f_X + f_L + f_E + f_M + f_R
\]

where \(f_X\) is the total cost of performance parameter, \(f_L\) is the flying trajectory length cost, \(f_E\) is terminal constraint, \(f_M\) is the mountain cost, namely the safe flying cost around the mountain, \(f_R\) is the radar detection cost of UAV in the process of flight. In this paper, we aim to find an optimal or near optimal trajectory under performance constraint, terminal constraint, mountain constraint and radar constraint. In order to meet these four constraints, the penalty terms are added in the objective, which are respectively denoted by \(f_X, f_L, f_M, f_R\). When the constraint is satisfied, the corresponding penalty term is set as 0. When any constraint is not satisfied, the corresponding term will be set a penalty value. Too large penalty value will make the length term \(f_L\) meaningless. Too small penalty value will make the four constraints meaningless. Thus in this paper, the penalty values are set as constants with the same magnitude but smaller than the trajectory length. In the simulation, the penalty value for \(f_X, f_L, f_M, f_R\) are set as 30.

IV. SOLVING STRATEGY OF TRAJECTORY PLANNING

In order to obtain efficient trajectory of UAV, a trajectory planning strategy which considers the characteristic and
performance of the established model must be developed. In this section, we introduce the distribution of collocation points firstly. Then we describe the model consisted by PSO and UAV kinematic model. Besides, cubic spline interpolation and the method based on integral and cubic spline interpolation are given in Sections IV-C and IV-D respectively. Finally, we propose the detailed process of the proposed method.

A. DISTRIBUTION OF COLLOCATION POINTS
In order to reduce the burden of calculation, the flying time between the initial moment of UAV (denoted as \( t_0 \)) and the moment of end is discretized as a certain number of
Chebyshev points, and they are the moments where the control variable is optimized. The moments of Chebyshev points are calculated by the following formula

\[
z_m = \cos \left( \frac{(n-1-m) \cdot \pi}{n-1} \right), m = 0, 1, 2, \ldots, n-1 \tag{19}\]

where the number of collocation points is set as \( n \) and the two endpoints, i.e., \( t_0 \) and \( t_{\text{end}} \) are also included, \( z_m \) is the moment of \((m+1)\)th collocation point, and the Chebyshev points are dense at both ends and sparse in the middle. This feature is just suitable for meeting the terminal constraints easily that there are more control variables to be operated at the end of phase [35]. As \( z_m \in [-1, 1] \) in (19), a conversion is needed to make the value of \( z_m \) fall into the interval \([0, t_{\text{end}}]\).

\[
t_m = \frac{t_{\text{end}} - t_0}{2} \cdot z_m + \frac{t_{\text{end}} + t_0}{2} \tag{20}\]

where \( t_m (m=0, 1, \ldots, n-1) \) is the real moment of collocation point, and the distribution of Chebyshev points is shown in Fig. 2.

In the implementation, the selection of the number of trajectory point \( n \) is very important. While increasing \( n \) will exponentially increase the computational burden, decreasing \( n \) will cause the loss of accuracy. After analysis and comparison, we set \( n = 22 \), which can guarantee trajectory feasibility and algorithm efficiency.

B. KINEMATIC MODEL COMBINED WITH PSO
In many previous studies, each particle in PSO algorithm is a collection of discrete points, which presents a possible trajectory of UAV. In this paper, as the kinematic model is considered, each particle of the algorithm represents a set of control variables, and the corresponding trajectory can be obtained by solving the kinematic equations. The scale of particle swarm is set as \( N \), and the number of collocation points is defined as \( n \) in 3-D environment. The formulas of updating solutions are presented as

\[
\Delta u_i (t) = \omega \cdot \Delta u_i (t-1) + c_1 \cdot r_1 \cdot (u_{ib} (t) - u_i (t)) + c_2 \cdot r_2 \cdot (u_{gb} (t) - u_i (t)) \tag{21}\]

\[
u_i (t + 1) = u_i (t) + \Delta u_i (t) \tag{22}\]

where \( u_i (t) \) is the control variable of the \( i \)th particle in the \( t \)th iteration, and \( u_{ib} (t) \) and \( u_{gb} (t) \) are the control vectors corresponding to the individual best solution and the global best solution. \( \omega \) is set as the inertia weight, \( c_1 \) and \( c_2 \) are acceleration constraints, and \( r_1 \) and \( r_2 \) are random number obeyed uniform distribution in interval \([0, 1]\). One time of iteration is finished after the solutions and results are updated according to (21) and (22) Note that if any control variable in \( u_i (t + 1) \) goes out of range, it will be set to the closest threshold, which is used to ensure the normal operation of algorithm.

C. CUBIC SPLINE INTERPOLATION
The concept of spline is originated from conscripting technique of using a flexible strip known as spline to draw the smooth curve through a set of points. To achieve more smoothness and less rate of error in the curve, the cubic spline interpolation method is adopted which is the most common and useful version in the field of engineering and science [37].

Define \( s(x) \), \( a \leq x \leq b \), as the cubic spline function, it should meet the following conditions:

(a) Approximate \( s(x) \) as a third-order polynomial in each interval \([x_i, x_j] (i = 1, \ldots, n - 1)\).

(b) The second derivative should be continuous.

(c) \( s(x) = y_i (i = 0, 1, \ldots, n - 1) \), where \( y \) is set as the original sampling data points.

According to (a), \( s_i (x) \) can be defined as follows which is assumed without loss of generality.
where, $a_i, b_i, c_i, d_i$, are set as the undetermined coefficients, and the total number of these coefficients is $4(n-1)$. Then, according to condition (b), we can obtain (25), the number of equations is calculated as $4(n-1) - 2$.

$$
\begin{align*}
\begin{cases}
    s_i(x_i) = s_{i+1}(x_i) \\
    s_i'(x_i) = s_{i+1}'(x_i) & i = 1, 2, \ldots, n-2 \\
    s_i''(x_i) = s_{i+1}''(x_i)
\end{cases}
\end{align*}
$$

In order to determine all undetermined coefficients in $s(x)$, we need to find other two conditions to calculate. The most common constraint which is used widely is the so-called natural boundary conditions, that is

$$s_i''(x_i) = s_{i-1}''(x_{i-1}) = 0 \quad (26)$$

On this basis, combining (23), (24), (25), with (26), we can reconstruct $s(x)$ that can be determined by solving linear equations uniquely. The important thing should be noted is that the cubic spline interpolation has been shown to have good convergence and fine smoothness, and the algorithm commonly used to calculate approximately the function in the neighborhood of its maximum. In order to make the effect more clear, the original cross-correlation sampling data are shown in Fig.3 and Fig.4, we can see clearly the distribution about the data of sample points and the result by using cubic spline interpolation.

**FIGURE 3. Sampling data points**

**FIGURE 4. Cubic spline interpolation results**

**D. SOLUTION OF CUBIC SPLINE COMBINED WITH INTEGRATION**

In this paper, the model consisted by PSO and kinematic model for UAV is used to optimize the trajectory planning problem. In the progress of optimizing the problem, cubic spline interpolation and algebraic integration are used to solve and calculate. The detailed steps are as follows

**STEP 1:** According to the kinematic model of UAV, the control variable is set as follows

$$u_i = [q_i, r_i]^T (i = 1, 2, \ldots, N) \quad (27)$$

where $q_i$ and $r_i$ are the trajectory inclination velocity and heading angular velocity of the $i$th collocation point, and control variable is distributed according to collocation points.

**STEP 2:** The cubic spline interpolation principle is used to interpolate all of the collocation points including the starting point and the ending point. Then, the function of control variable combined with time is obtained, which is presented as follows

$$
\begin{align*}
\begin{cases}
    F_{qi}^k = f_{qi}^k(q_i, t) \\
    F_{ri}^k = f_{ri}^k(r_i, t)
\end{cases} \quad i = 1, 2, \ldots, N
\end{align*}
$$

where $F_{qi}^k, F_{ri}^k (k = 1, \ldots, n-1)$ are defined as the function of trajectory inclination velocity and heading angular velocity with time in the $k$th phase respectively. Coefficients of the function are obtained by cubic spline interpolation, on this basis, the method of constructor function is adopted to establish the specific function, such as the function of trajectory inclination velocity is introduced as follows

$$F_{qi}^k = a_0 \cdot t^3 + a_1 \cdot t^2 + a_2 \cdot t + a_3 \quad (29)$$

where $a_0, a_1, a_2, a_3$ are the polynomial coefficients.
\[ F_{\theta i}^k = \int F_{\theta i}^k dt \quad i = 1, 2, \ldots, N \]  
\[ F_{\phi i}^k = \int F_{\phi i}^k dt \]  
\[ F_{ao}^k = \frac{1}{4} b_0 \cdot t^4 + \frac{1}{3} b_1 \cdot t^3 + \frac{1}{2} b_2 \cdot t^2 + b_3 \cdot t + c_0 \]  
\[ F_{ao}^1, F_{ao}^2, F_{ao}^3 (k = 1, \ldots, n-1) \]

STEP 3: According to the principle and theory of physics, the angular velocity is integrated to get the angle. After the function of angular velocity is obtained, the function of angle with time are calculated by integration, and they are expressed as follows

\[ F_{\theta i}^k = \int F_{\theta i}^k dt \quad i = 1, 2, \ldots, N \]  
\[ F_{\phi i}^k = \int F_{\phi i}^k dt \]

where \( F_{\theta i}^k, F_{\phi i}^k (k = 1, \ldots, n-1) \) are the functions of the trajectory inclination and heading angle in the \( k \)th phase respectively. Taking the function of trajectory inclination as an example, which is presented as follows

\[ F_{\theta i}^k = \frac{1}{4} a_0 \cdot t^4 + \frac{1}{3} a_1 \cdot t^3 + \frac{1}{2} a_2 \cdot t^2 + a_3 \cdot t + c \]  

where \( c \) is determined by the initial value of the trajectory inclination in the \( k \)th phase.

STEP 4: According to the formula of angle, and the UAV kinematic model, the values of positional derivative in collocation points are obtained. Then the co-efficient matrix of positional derivative is obtained by cubic spline interpolation. The functions are introduced as

\[ F_{xi}^k = f_i (V, \theta, \phi, t) \]  
\[ F_{yi}^k = f_i (V, \theta, \phi, t) \quad i = 1, 2, \ldots, N \]  
\[ F_{zi}^k = f_i (V, \theta, \phi, t) \]

where \( F_{xi}^k, F_{yi}^k, F_{zi}^k (k = 1, \ldots, n-1) \) are the functions of \( \dot{x}, \dot{y}, \dot{z} \) respectively.

Because the function of positional derivative is complex to calculate. In order to overcome the problem, the value of positional derivative in each collocation point is calculated firstly. Based on these discrete points, the function of positional derivative is obtained by cubic spline interpolation, and taking the function of \( \dot{x} \) as an example, it is introduced as

\[ F_{xi}^k = b_0 \cdot t^4 + b_1 \cdot t^3 + b_2 \cdot t^2 + b_3 \cdot t + b_4 \]  

where \( b_0, b_1, b_2, b_3, b_4 \) are the polynomial coefficients.

STEP 5: Using the algebraic integration again to get the expression of the positional function as follows

\[ F_{xi}^k = \int F_{xi}^k dt \]  
\[ F_{yi}^k = \int F_{yi}^k dt \quad i = 1, 2, \ldots, N \]  
\[ F_{zi}^k = \int F_{zi}^k dt \]

where \( F_{xi}^k, F_{yi}^k, F_{zi}^k (k = 1, \ldots, n-1) \) are the functions of \( x, y, z \) respectively.

Taking the function expression of \( x \) as an example, and it is as follows

\[ F_{ao}^k = \frac{1}{4} b_0 \cdot t^4 + \frac{1}{3} b_1 \cdot t^3 + \frac{1}{2} b_2 \cdot t^2 + b_3 \cdot t + c_0 \]  

where \( c_0 \) is determined by the initial numerical value of the position in the \( k \)th phase.

Algorithm 1 Pseudo code of the proposed method.

//PSO initialization
1: Set starting position and destination position for UAV;
2: Set the swarm parameters, \( N, n, \omega_{\max}, \omega_{\min}, c_{\max}, c_{\min}, T \);
3: Set chaos parameter \( \mu = 4 \);
4: Initialize particles using (5);
5: Set a relative large value for fitness value;
6: Calculate particle fitness value and select gbest and pbest;

//Main loop
7: For \( i = 1 \) to \( T \);
8: Update PSO parameters by (4) and (6)–(7); 
9: Generate new \( u_i \) and \( \Delta u_i \) of particle in the group by (21)–(22); 
10: Calculation of angles and positions by (27)–(35);
11: Sort fitness value of all particles;
12: Select gbest and pbest;
13: Remember the value and the integration for gbest;
14: \( i = i + 1 \)
15: End
16: Output the planned trajectory for UAV

E. PROCEDURE OF TRAJECTORY PLANNING

The procedures of solving the trajectory planning problem for UAV are shown in Algorithm 1. The detailed procedures are introduced as follows:

STEP 1: Construct the mission area, terrain and threat model. Set the starting and destination point for UAV.

STEP 2: PSO parameters initialization for each UAV, including swarm scale \( N \), the number of collocation points \( n \), acceleration coefficients \( c_{\min}, c_{\max} \), maximum iterative time \( T \), inertia weight \( \omega_{\max}, \omega_{\min} \), and swarm initialization by (5).

STEP 3: Angular velocity is set as the control variable, the function of angular velocity is solved by cubic spline interpolation, and the functions of position and angle are obtained by integration. Construct the fitness function for UAV, and carry out the algorithm for one time.

STEP 4: Calculate the fitness function, figure out the best position and swarm best position.

STEP 5: Update parameters by given strategies, including acceleration coefficients swarm initialization, and inertia weight.

STEP 6: Update \( u_{ib}, u_{ib}, u_{sb} \) and according to the rules in PSO algorithm.

STEP 7: Output the optimal control variable for UAV at collocation points. Obtain the state of UAV at collocation.
points. Finish the evolution and plot the optimal trajectory for trajectory planning.

V. SIMULATION ANALYSIS
In this section, a variety of numerical simulations are carried out to demonstrate the feasibility and effectiveness of the proposed method in detail. The algorithm is implemented in MATLAB environment and is run on a PC with 2.1GHz CPU, 8.00GB of RAM and 64-bit operating system.

A. PARAMETER SETTING
In the simulation scenario, UAV flies from the starting point to the target point under the given battlefield environment. The environment parameters of radars and mountains are shown in Table 1, and the starting point and target point are (0,150,20) m and (200,50,30) m respectively. The swarm size is \( N = 500 \), the maximum iteration is \( T = 150 \), and the number of collocation points is \( n = 22 \). The inertia weight parameters are \( w_{\text{max}} = 0.9 \), \( w_{\text{min}} = 0.4 \), and the acceleration coefficient parameters are \( c_{\text{max}} = 3.5 \), \( c_{\text{min}} = 0.4 \). Then, the initialization parameter of chaos is set as \( \mu = 4 \). The flight speed is 8.5m/s, and the flying time is set as 30s. The initial values and ranges of control variable and state variable are shown in Table 2.

| Name of model threat parameters | Parameter setting (m) |
|---------------------------------|-----------------------|
| Mountain1                       | \( x^0 = 60, y^0 = 140, h = 70, x' = 19, y' = 12 \) |
| Mountain2                       | \( x^0 = 60, y^0 = 40, h = 70, x' = 19, y' = 19 \) |
| Mountain3                       | \( x^0 = 175, y^0 = 40, h = 70, x' = 19, y' = 19 \) |
| Radar1                          | \( x'_1 = 160, y'_1 = 140, z'_1 = 0, R = 40 \) |
| Radar2                          | \( x'_2 = 160, y'_2 = 140, z'_2 = 0, R = 40 \) |

| Table 2. Setting of initial states and boundary values | Value |
|------------------------------------------------------|-------|
| \( q(t_0) \)                                        | 0 \text{ rad/s} |
| \( r(t_0) \)                                        | 0 \text{ rad/s} |
| \( \theta(t_0) \)                                   | 0 \text{ rad} |
| \( \phi(t_0) \)                                     | 0 \text{ rad} |
| \( q_{\text{max}} \)                                | 0.2 \text{ rad/s} |
| \( r_{\text{max}} \)                                | 0.2 \text{ rad/s} |
| \( \theta_{\text{max}} \)                           | 90 \text{ deg} |
| \( \phi_{\text{max}} \)                             | 90 \text{ deg} |
| \( d_{\text{f}} \)                                  | 1 \text{ m} |

B. RESULTS AND COMPARISONS IN 3-D ENVIRONMENT
In the simulation, the trajectory of UAV should not exceed the given ranges. Therefore, the parameters are constrained to ensure the effectiveness of trajectory. The ranges of environment are shown as follow

\[
0 \leq x \leq 200m \\
0 \leq y \leq 200m \\
0 \leq z \leq 200m
\] (36)

![FIGURE 5. Results of trajectory planning, (a) Fitness value of proposed method. (b) 3-D trajectory. (c) Top view](image-url)
Fig. 5 shows the fitness value evaluations and 3-D trajectory generated by the proposed method. The fitness value is measured by (18) and a penalty term is added when UAV conflicts with obstacles or other parameters surpass the ranges of threshold. Fig. 5(a) shows the evolution process for fitness value, which is vital to evaluate the proposed method performance, and the fitness value achieves convergence after about 50 times of iterations. Figs. 5(b) and (c) are the planned trajectories, it's easy to see that the strategy can guarantee feasible trajectory. After calculation, the flight distance of UAV is about 264.18 m. Besides, the distance between the end point of the trajectory and the target point is 0.14 m, it is smaller than \( d_f \), thus the results meet the preset requirements and verify the feasibility of the algorithm.

Fig. 5 shows the fitness value evaluations and 3-D trajectory generated by the proposed method. The fitness value is measured by (18) and a penalty term is added when UAV conflicts with obstacles or other parameters surpass the ranges of threshold. Fig. 5(a) shows the evolution process for fitness value, which is vital to evaluate the proposed method performance, and the fitness value achieves convergence after about 50 times of iterations. Figs. 5(b) and (c) are the planned trajectories, it's easy to see that the strategy can guarantee feasible trajectory. After calculation, the flight distance of UAV is about 264.18 m. Besides, the distance between the end point of the trajectory and the target point is 0.14 m, it is smaller than \( d_f \), thus the results meet the preset requirements and verify the feasibility of the algorithm.

**FIGURE 6.** Angular velocity. (a) Trajectory inclination velocity. (b) Heading angular velocity.

Fig. 6 introduces the information of angular velocity. In this paper, angular velocity is regarded as the control variable, which is optimized by the proposed method. Figs. 6(a) and (b) are the trajectory inclination velocity and heading angular velocity respectively. The black asterisks indicate the collocation points, the cubic spline interpolation represent the reference values, and the two straight lines are the preset boundaries of angular velocity. One can see that the curve of cubic spline interpolation is smooth and can provide an effective reference for the flight of UAV. The numerical values of angular velocity are also within the preset ranges, meaning that they can meet the given constraints. Thus, the proposed method can guarantee that optimized angular velocity can be used as the reference of control variable.

**FIGURE 7.** Information of trajectory inclination and heading angle. (a) Trajectory inclination. (b) Heading angle.

Beside angular velocity, the states of angle are also provided and shown in Fig. 7. The information of angle can reflect the flight directions of UAV in 3-D environment. Fig. 7(a) is the curve of trajectory inclination, and Fig. 7(b) is the curve of heading angle. It can be seen that each angle satisfies the boundary conditions which are expressed by two straight lines. Meanwhile, we can also see that the curve of cubic spline interpolation is consistent with the curve of discrete
points, which indicates the discrete collocation points can offer good guidance.

The above results demonstrate that the proposed multi-phase method of solving trajectory planning is effective in generating optimal trajectory for UAV, and the solution can meet all of the constraints proposed in the trajectory planning model.

To further verify the feasibility of the proposed method, different case is selected for simulation; the coordinates of starting point and target point are changed. On this basis, UAV can strike different target from different starting point.

The coordinates of starting point and target point are shown in Table 3. Herein, we give the related results and show them in Fig. 8. Fig. 8(a) is the fitness value convergence curve, it can be seen that the fitness value decreases from 730 to 223.25 and becomes convergent after about 40 times of iterations. Figs. 8(b) and (c) are the planned trajectory in 3-D environment. It can be found that the distance from starting point to target point is about 223.25m, besides, UAV can plan effective trajectory and the results can satisfy all the constraints in the mathematical model. Thus, the feasibility of the proposed multi-phase method is verified.

**TABLE 3. Data of trajectory planning**

| Item                  | Value       |
|-----------------------|-------------|
| Starting point /m     | (0.75,20)   |
| Target point /m       | (200,100,30) |
| $L / m$               | 223.25      |
| $d_{min} / m$         | 0.063       |
| $V / m / s$           | 8.5         |

**C. MONTE-CARLO SIMULATION IN 3-D ENVIRONMENT**

Apart from the feasibility, effectiveness and success rate of trajectory are also important factors in evaluating the performance. Due to the randomness of PSO, it is not enough to evaluate the performance by running only one time. In order to better show the advantages of the proposed method, we also carry out Monte-Carlo simulation. In the following, the Monte-Carlo simulation times are set as 20 and the related results are shown from Figs. 9–11.
As a comprehensive indicator for solution optimality, the fitness values for UAV are firstly calculated in Fig. 9(a). From the result, one can see that curves of fitness function converge to a same value after 150 times of iterations, which means that almost the same final value is obtained. Figs. 9(b) and (c) indicate the trajectories in 3-D environment. The starting point, target point, flight speed and other parameters are the same as Fig. 5(b). It is obvious that the planned trajectories are effective and smooth, which illustrates that UAV can reach the target point accurately and avoid obstacles successfully in the 20 times of simulations.

Fig. 10 shows the results for trajectory inclination velocity and heading angular velocity. Fig. 10 (a) is the curve of trajectory inclination velocity after Monte-Carlo simulation, and Fig. 10(b) is the curve of heading angular velocity. In Figs. 10(a) and (b), the discrete points represent control variable at the collocation points, the cubic spline interpolations represent the reference command for the actual flight of UAV. One can see that the discrete points are in agreement with the curves of cubic spline interpolation, and all of the values are within the ranges of constraints. Thus the proposed strategy can provide effective reference for the UAV system, and all the optimized solutions are feasible.
To make a better understanding, the trajectory inclination and heading angle are presented in Fig. 11. Firstly, it can be seen that the values of discrete points are consistent with the values of cubic spline interpolation, which illustrates that the proposed multi-phase method can also provide effective guidance for the angles. Besides, the value of each angle is within the two straight lines, showing that the results meet the constraint range.

To better indicate the stability and performance of the proposed method, the runtime and objective function value are shown in Fig. 12. After calculation the mean of running time is 88.1575s, and the mean of objective value is 254.2834m. Besides, the values of standard deviation are 5.68518 and 0.141842, respectively. By analyzing these data, we can get a conclusion that the proposed multi-phase method is feasible and stable.

**D. COMPARISON WITH CPTD**

To evaluate our proposed method, we adopt two different methods to make the simulation. One is the method we proposed which has been introduced, and the other is the method in literatures [32]-[34] (CPTD). The related results of comparison are shown in Figs. 13–14. Figs. 13(a)-13(b) are the trajectory planning results by CPTD and proposed multi-phase method respectively, where the first figure is the evolution process for fitness value. It is straightforward that the fitness value of proposed method converges faster than that of CPTD, thus the proposed method maintains faster convergence than CPTD. Fig. 13(b) shows the flying trajectory of the proposed method is smoother than that of CPTD.
To investigate the reason, the detailed information of angular velocity and angle by two methods are presented in Fig. 14. It can be seen from Fig. 14(a) that the time is divided equally, the control input of CPTD is close to the piecewise constant value. However, according to the method we proposed, the flying time is discretized as a certain number of Chebyshev points, and the control input is a curve. Comparing with the constant input, the control input in our research is closer to the real scene. Besides, in Fig. 14(b) the connecting line of the angle generated by CPTD is a broken line, which results in the trajectory not smooth. But, we can observe that the angle value in our research is close to a curve. Therefore, a much smoother trajectory can be generated by using our method.

E. RESULTS AND COMPARISONS IN 2-D ENVIRONMENT

In order to highlight the effect of the proposed method in complex environment, the environment of trajectory planning is changed to 2-D, namely UAV flies with a constant height, and the circles are adopted as the obstacles. The center coordinates and radius of the obstacles are shown in Table 4, the given parameters for control variable and state variable are the same with those in 3-D environment. In this section, multi-phase method is used to obtain the results of 2-D trajectory planning and the results are shown in Figs. 15-16.

| Obstacles | Center coordinates /m | Radius /m |
|-----------|------------------------|-----------|
| Obstacle1 | (50,80)                | 20        |
| Obstacle2 | (100,115)              | 20        |
| Obstacle3 | (110,60)               | 16        |
| Obstacle4 | (150,100)              | 16        |
| Obstacle5 | (50,130)               | 16        |
| Obstacle6 | (50,20)                | 20        |
| Obstacle7 | (100,170)              | 16        |
| Obstacle8 | (150,30)               | 20        |

Fig. 15(a) is the convergence curve of fitness function, as can be seen from the figure, the fitness value is about 225.62 after about 70 times of iterations. Fig. 15(b) shows the planned trajectory. In the figure, the starting point is set as (0,100)m, and the target point is set as (0,200)m. It is obvious that the trajectory is smooth and effective. UAV reaches the target point and avoids the obstacles successfully in the process of flying. After calculation, the distance between the target point and the end position is 0.067m which is smaller than 1m.
In Fig. 16, heading angular velocity and the heading angle are introduced in detail. One can see all the values are within the preset range and the curves are smooth. Because UAV is assumed to fly with a constant height in 2-D environment, so the trajectory inclination and the velocity is not considered.

The above results illustrate that the proposed multi-phase method can guarantee that the trajectory is feasible, and the solution can meet all of the constraints.

In order to verify the feasibility again, the coordinates of starting point and target point are changed, the starting point is set as (0,40), the target point is set as (200,140). The related results are shown in Figs. 17-18. From Figs. 17(a) and (b) we can see that the fitness value becomes the convergence after about 26 times of iterations and the planned trajectory is also smooth and effective. In Figs.18(a) and (b), it can be seen that the values of angular velocity and angle can meet all of the constraints, thus the proposed method can still obtain feasible solution after changing the parameters, and the feasibility of the proposed method is verified.
Similarly, in order to verify the effectiveness, Monte-Carlo simulation for 2-D environment is also presented. In the following, the simulation times of Monte-Carlo are set as 20 and the related results are shown in Figs. 19-20.

![Figure 17](image1.png)

**FIGURE 17.** Results of trajectory planning in 2-D environment. (a) Fitness value of function. (b) 2-D trajectory.

![Figure 18](image2.png)

**FIGURE 18.** Information of angular velocity and angle. (a) Heading angular velocity. (b) Heading angle.

![Figure 19](image3.png)

**FIGURE 19.** Results of Monte-Carlo simulation. (a) Fitness value of function. (b) 2-D trajectory.

Fig. 19 introduces the results of trajectory planning for Monte-Carlo simulation. The fitness values are shown in Fig. 19(a), one can see that all of the fitness values can reach the convergent values after about 20 times of simulations, and the final fitness values are between 235m-250m, the difference of values is due to the randomness of PSO. After calculation, the fitness values are stable in the end, which demonstrates that the multi-phase method can obtain the feasible solution in each simulation. Fig. 19(b) shows the planned trajectories for Monte-Carlo simulation. Although some different trajectories are obtained, it is ensured that UAV can also reach the target point as expected and all trajectories are feasible and effective. UAV avoids all obstacles in the
process of flying successfully. To further investigate the effectiveness of the multi-phase method, the heading angular velocity and heading angle are also presented in Fig. 20. Clearly, the constraints of angle and angular velocity are satisfied, and discrete points are in agreement with the curves of cubic spline interpolation.

Thus, the results of Monte-Carlo simulation demonstrate that UAV can reach the target successfully, and the planned trajectory can meet all of the constraints. The effectiveness of the proposed multi-phase method in 2-D environment is verified.

FIGURE 20. Information of angular velocity and angle. (a) Heading angular velocity. (b) Heading angle.

**F. COMPARISON WITH SINGLE-PHASE METHOD**

In order to verify the advantage of the proposed multi-phase method, the single-phase method (in this method the process of solving is treated as a single process) is proposed for comparison and verification. The difference between the multi-phase method and single-phase method is that the single-phase method uses curve fitting and integral to solve the problem of trajectory planning, and all of the collocation points are regarded as a whole to calculate. The related results of the single-phase method are shown in Figs. 21-22. In order to achieve fair comparison, the flight and environmental parameters of the single-phase method are the same as the multi-phase method.

![Image of Fig. 21](image-url)

**FIGURE 21.** Results of trajectory planning for single-phase method. (a) Fitness value of function. (b) 2-D trajectory.

In Fig. 21(a), one can see that the fitness value does not converge after about 150 times of iterations, which indicates the solution may not be feasible. Fig. 21(b) shows the planned trajectory, we can observe that UAV does not hit the target and deviates from the target greatly under the single-phase method.

Compared to Fig. 15(b), it’s obvious that the multi-phase can guarantee that UAV obtains feasible trajectory, but the single-phase method is failed. Besides, compared to Fig. 15(a) it is clear that the fitness value reaches the feasible value after about 70 times of iterations in multi-phase method. However, the fitness value of single-phase method does not reach the feasible solution in Fig. 21(a).
The results of simulation mean that UAV may not finish the task under the single-phase method.

![Heading angular velocity and heading angle](image)

**FIGURE 22.** Information of angular velocity and angle. (a) Heading angle velocity, (b) Heading angle.

To further investigate the reasons for the failure of the single-phase method, the detailed information of angular velocity and angle is presented in Fig. 22. Fig. 22(a) introduces the information of heading angular velocity, the discrete points are the calculated control variables at collocation points, and the curve fitting represent reference values of the actual flight for UAV. In Fig. 22(a), it’s easy to see that the fitted curve is not consistent with the curve of discrete points, which shows that there are some errors between the theoretical calculated values and the reference values. The reason is that fitted curve only reflects entire change trend of all the collocation points. Compared to Fig. 16(a), we can also conclude that the results of the multi-phase method are better than the single-phase method.

Besides, the results of Monte-Carlo simulation for the single-phase method are shown in Fig. 23. It is obvious that the fitness values are different in the end, and most of the values are inefficient or fall into local optimum, and UAV cannot hit the target point as expected. Thus the planned trajectories are ineffective, and no one is successful. On the contrary, Fig. 16 shows the feasible results with the multi-phase method. Therefore, compared to the single-phase method, the proposed multi-phase method decomposes the complicated constraints into many subparts, i.e., the process of solving the trajectory planning problem is divided into multi-phase, which makes it easier to obtain satisfactory solution. In a word, the above results demonstrate and verify the superiority of the proposed multi-phase method.

![Monte-Carlo simulation for single-phase method](image)

**FIGURE 23.** Monte-Carlo simulation for single-phase method. (a) Fitness value of function, (b) 2-D trajectory

**VI. CONCLUSION**

In this paper, we proposed a new method of solving UAV trajectory under obstacles and multi-constraint. The PSO, UAV kinematic model, Chebyshev points, cubic spline interpolation, and multi-segment strategy are respectively introduced and a smooth and feasible can be obtained. The detailed contributions are as follows:

1. To solve the established trajectory planning model, a PSO-based algorithm is developed to generate the feasible trajectories. Collocation points are set to reduce the computation load of search and improve the solution quality, the collocation points are distributed by moment.
Cubic spline interpolation and integration are adopted to acquire the functions of control variable and state variable. By proposing the multi-phase method extending the traditional path planning based on PSO with only position information to four-dimensional trajectory planning with flying time and three-dimensional (3-D) position.

(2) In the simulation studies, the superiority of the proposed multi-phase method is verified by comparisons. Comparing with single-phase, the proposed multi-phase method can better address the complicated constraints and result is very close to the theoretical optimal solution. Comparing with CPTD the multi-phase method can obtain more accurate and closer to the actual trajectory.

In the future, we will continue to research on the issues, the dynamic model of quadrotor UAV will be established, cubic spline interpolation and integration is also adopted to generate the trajectory. Besides, other applications of the UAV system need to be discovered to expand its military and civil use.

REFERENCES
[1] R.-J. Wai and A. S. Prasetia, “Adaptive neural network control and optimal path planning of UAV surveillance system with energy consumption prediction,” IEEE Access, vol. 7, pp. 126137–126153, 2019.
[2] Y. Wang, P. Bai, X. Liang, W. Wang, J. Zhang, and Q. Fu, “Reconnaissance Mission Conducted by UAV Swarms Based on Distributed PSO Path Planning Algorithms,” IEEE Access, vol. 7, pp. 105086–105099, 2019.
[3] K. Harikumar, J. Saradhi, and S. Sundaram, “Multi-UAV ocyhynx marina-inspired search and dynamic formation control for forest firefighting,” IEEE Trans. Autom. Sci. Eng., vol. 16, no. 2, pp. 863–873, Apr. 2019.
[4] A. Madridano, A. Al-Kaff and D. Martin, “Trajectory planning for multi-robot systems: Methods and applications,” Expert Syst. Appl., vol. 173, Jul. 2021, Art. no. 114660.
[5] Y. V. Pehlivanoglu, “A new vibrational genetic algorithm enhanced with a Voronoi diagram for path planning of autonomous UAV,” Aerosp. Sci. Technol., vol. 16, no. 1, pp. 47-55, Jan. 2012.
[6] S. Bayili and F. Polat, “Limited-Damage A*: A path search algorithm that considers damage as a feasibility criterion,” Knowl.-Based Syst., vol. 24, no. 4, pp. 501-512, May 2011.
[7] M. A. Brubaker, A. Geiger, and R. Urtsasun, “Map-based probabilistic visual self-localization,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 38, no. 4, pp. 652-665, Apr. 2016.
[8] C.-B. Moon and W. Chung, “Kinodynamic planner dual-tree RRT (DT-RRT) for two-wheeled mobile robots using the rapidly exploring random tree,” IEEE Trans. Ind. Electron., vol. 62, no.2, pp. 1080-1090, Feb. 2015.
[9] Y. Li and J. Zheng, "Real-time collision avoidance planning for unmanned surface vessels based on field theory." ISA Trans., vol. 106, pp. 233-242, Nov. 2020.
[10] J. Wu, H. Wang, M. Zhang, and Y. Yu, "On obstacle avoidance path planning in unknown 3D environments: A fluid-based framework." ISA Trans. Vol. 111, pp. 249-264, May 2021.
[11] U. Challita, W. Saad, and C. Bettstetter, "Interference management for cellular-connected UAVs: A deep reinforcement learning approach." IEEE Trans. Wireless Commun., vol. 18, no. 4, pp. 2125-2140, Apr. 2019.
[12] Y. Hu, M. Chen, W. Saad, H. Vincent Poor, and S. Cui, "Distributed Multi-agent Meta Learning for Trajectory Design in Wireless Drone Networks," 2020, arXiv:2012.03158. [Online]. Available: http://arxiv.org/abs/2012.03158.
[13] B.-J. Zhu, Z.-X. Hou, and H-J Ouyang, "Trajectory optimization of unmanned aerial vehicle in dynamic soaring." Proc. Inst. Mech. Eng., G, J. Aerosp. Eng., vol. 231, no. 10, pp. 1779-1793, Aug. 2017
Algorithm." *IEEE Trans. Autom. Sci. Eng.*, vol.18, no.3, pp. 1263–1276, Jun. 2020.

[35] Y. Wu, "Coordinated path planning for an unmanned aerial-aquatic vehicle (UAAV) and an autonomous underwater vehicle (AUV) in an underwater target strike mission." *Ocean. Eng.*, vol. 182, pp. 162-173, Jun. 2019.

[36] S. Shao, Y. Peng, L. He, Y. Du, "Efficient path planning for UAV formation via comprehensively improved particle swarm optimization." *ISA Trans.* vol. 97, pp. 415-430, Feb. 2020.

[37] Z. Zheng, Y. Wang, L.-Z. Lv, Y. Zhou, H. Zhang, H.-B. Zhang and H.-Y. Wang, "A new time-delay estimation with high precision based on Cubic spline interpolation and Area bary center algorithm" Present at 2014 12th International Conference on Signal Processing(ICSP). Available: https://ieeexplore.ieee.org/document/7014971.