IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received March 11, 2021, accepted April 9, 2021, date of publication April 15, 2021, date of current version April 27, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3073420

Efficient Trajectory Planning for UAVs Using Hierarchical Optimization

SHIKAI SHAO[1,2], CHENGLONG HE[1], YUANJIE ZHAO[2], AND XIAOJING WU[2]

[1]State Key Laboratory of Satellite Navigation System and Equipment Technology, Shijiazhuang 050081, China
[2]School of Electrical Engineering, Hebei University of Science and Technology, Shijiazhuang 050018, China

Corresponding author: Shikai Shao (kdssk@hebust.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61903122, 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 and Grant 62003129.

ABSTRACT Automatic generation of feasible trajectory is one of the key technologies for autonomous flying of unmanned aerial vehicles (UAVs). The existing path planning methods, such as swarm intelligence algorithm and graph-based algorithm, cannot incorporate the flying time and UAV dynamic model into evolution. To overcome such disadvantages, a hierarchical trajectory optimization scheme consisted by improved particle swarm optimization (PSO) and Gauss pseudo-spectral method (GPM) is investigated in this paper. Firstly, considering that traditional GPM is sensitive to initial values, we design an improved PSO for path planning in the first layer. By introducing adaptive parameter adjustment strategy and position mutation updating strategy, the rapidity and optimality of the improved PSO is enhanced. Then in the second layer, a fitted curve based on the path waypoints generated by improved PSO is constructed and served as the initial values for GPM. Comparing with random initial values, the designed curve can significant improve GPM efficiency. A multi-segment strategy is also put forward to further improve the efficiency. Finally, with the consideration of dynamic model and state constraints, the time minimum trajectory planning for quadrotor UAVs is solved. Plenty of simulations are carried out and the results illustrate that the proposed scheme guarantees much better efficiency.

INDEX TERMS Unmanned aerial vehicles, trajectory planning, particle swarm optimization, Gauss pseudo-spectral method.

I. INTRODUCTION

Unmanned aerial vehicles (UAVs) are aircrafts without human pilots onboard. With the development of technology, UAVs have been widely applied to both military and civilian tasks, such as intelligence, surveillance, reconnaissance, rescue and commercial performance [1]–[3]. In order to enhance the autonomy level and operating efficiency, automatic generation of flyable trajectory for UAVs has becoming a crucial technology.

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 UAV kinematic and dynamic models 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 dynamic model of UAV was also not considered. In addition, all the aforementioned methods can only provide path information other than

The associate editor coordinating the review of this manuscript and approving it for publication was Seyedali Mirjalili.

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 classified into two branches: indirect method and direct method [15]. Both methods attempt to minimize objective function and constraint violations by using discrete approximations. Comparing with indirect method, the direct one removes the Pontryagin minimum principle and is more convenient to implement. For the utilization of direct method, the continuous NOC problem is firstly transformed into a nonlinear programming (NLP) problem, then it could be solved numerically via some well-developed toolboxes. With maintaining the property of dynamic feasibility, pseudo-spectral optimization method [16] has proved to be an effective direct numerical method and attracted much attention. A well-known application of the pseudo-spectral method is the attitude maneuver of NASA's space telescope, by which almost zero fuel consumption was achieved and millions of dollars were saved [17]. A recent review [18] on optimization techniques in spacecraft flight trajectory design also showed the effectiveness of pseudo-spectral method. In [19], a Chebyshev pseudo-spectral method (CPM) was introduced and reentry trajectory optimization for hypersonic glide vehicle with flexible initial conditions was also addressed. Among the pseudo-spectral methods, GPM has been proved to be able to maintain relative faster convergence and more accurate solutions during the optimization process. However, like other numerical methods, all pseudo-spectral methods also inherent the drawbacks of sensitive to initial values and lack of a fast global searching capability, indicating that poor initial value may weaken the effectiveness or even make no feasible solution for a NOC problem.

Besides numerical methods, the bio-inspired swarm intelligence algorithms, such as ant colony optimization (ACO), particle swarm optimization (PSO) and plant growth algorithm, have also been widely investigated for path planning by researchers [20]–[22]. These algorithms not only possess the advantage of being able to achieve global optimal or near-optimal solution fast but also are insensitive to initial values. Among the existing swarm intelligence algorithms, PSO maintains the advantages of simple implementation and being able to access global optimum, thus it is often utilized for path planning and other optimization tasks. Although PSO may trap into local minimum and premature convergence in some situations [23], one can always refine the PSO performance by some appropriate improvements. In [24], [25], a new idea of separately evaluating and evolving path waypoints for UAV planning was proposed. In [26], the authors proposed an adaptive decision operator for PSO path planning and illustrated the advantages. In [27], [28], 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 of PSO.

Taking both advantages of swarm intelligence algorithm and numerical methods, it may provide a meaningful and effective solution for UAV trajectory planning by involving pseudo-spectral methods with swarm intelligence algorithm. In accordance with this idea, Modares and Sistani combined PSO and successive quadratic programming (SQP) together and proposed an integration optimization method in [29]. The PSO was activated at the beginning to find a near-optimal solution quickly, then the SQP was adopted to accelerate the search process and to find an accurate solution in the final. Appealed by the result, some researchers extended the idea to optimal trajectory planning for under-actuated spacecraft and autonomous underwater vehicles [30], [31]. The newest results on robot trajectory planning also enriched the relevant research. In [32], polynomial interpolation function was combined with improved cuckoo search algorithm to generate time-optimal trajectory for UR robot. In [33], graph-based multi-agent path planner was involved with nonlinear optimization to generate smooth trajectories for non-holonomic mobile robots. The resulting trajectories showed the superiority of hybrid optimization.

It is to be noted that although some results based on pseudo-spectral methods or other numerical methods have been reported, the results are not always satisfactory for kinds of unmanned robots or vehicles with different models. The in-depth improvements, such as to further reduce computation complexity and to further improve algorithm efficiency, are still needed. To our best knowledge, no one has investigated the time-efficient trajectory planning for quadrotor UAVs with the consideration of six degree-of-freedom (6-DOF) dynamic model. Besides, it is concluded from the analysis that the generation of favorable initial values for numerical optimization algorithm plays a key role in improving the efficiency and optimality of the trajectory planning. Motivated by these aspects, a hybrid hierarchical optimization scheme combined by improved PSO (IPSO) and Gauss pseudo-spectral method (GPM) is proposed for multi-UAV trajectory planning in the paper. The IPSO and improved GPM are respectively designed in the first and second layer to carry out path planning and trajectory planning. The generated path waypoints by IPSO act as the bridge to combine the two layers. The details and contributions of the paper are as follows.

Firstly, an improved PSO algorithm is designed for path planning in the first layer. The convergence speed and solution optimality are improved by designing adaptive parameter varying strategy and position mutation updating strategy. Secondly, the hierarchical optimization scheme combined by IPSO and GPM is designed for the 6-DOF quadrotor UAVs. A fitted curve based on IPSO path waypoints was designed as the initial values of GPM and the resultant solving time is greatly reduced. Thirdly, an adaptive multi-segment strategy was designed for the hybrid optimization. By dividing the

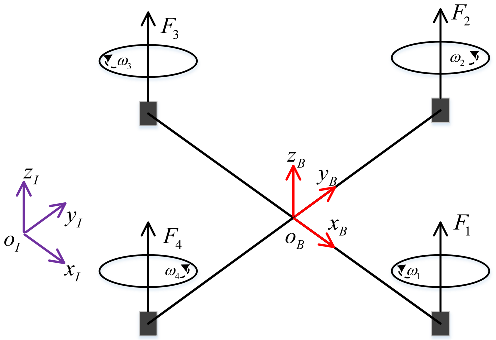


FIGURE 1. **Configuration of quadrotor UAV.**

whole trajectory into several segments according to the curvature of fitted curve, the solving time can be further reduced and near online trajectory planning can be achieved.

The rest of the paper is organized as follows: the upcoming section introduces the mathematical model of quadrotor UAV trajectory planning. The IPSO is proposed and analyzed in the next section. Then the detailed design of IPSO-GPM planning is discussed. Simulations and comparison results are carried out in a later section. Finally, conclusion is given in the last section.

II. PROBLEM FORMULATION

In this paper, the quadrotor UAV is researched for trajectory planning. Comparing with fixed-wing UAVs, the quadrotors can maintain hovering flight and are easier to execute some missions, such as cargo delivery, photography, etc. Different from some results with only 2D plane considered, the 3D trajectory planning with 6-DOF dynamic model is investigated in this research. In the following, the dynamic model for quadrotor UAV is firstly introduced. Then UAV state and environment constraints are presented and the NOC problem for UAV formation trajectory planning is formulated.

A. MODEL OF QUADROTOR UAV

Let n be the number of quadrotor UAVs in the formation denoted by $F = \{F_1, F_2, \ldots, F_n\}$. Assume all UAVs are the same in the formation. For dynamic principle illustration, the body-fixed frame $O_B x_B y_B z_B$, the earth-fixed inertial frame $O_I x_I y_I z_I$ and the configuration of quadrotor UAV are shown in Fig. 1.

In this research, the 6-DOF UAV model is taken into consideration. Under the assumption of symmetric rigid body and low-speed flying without disturbances, a simplified dynamics of quadrotor UAV involving translational and rotational motions is introduced as follows [34], [35]

$$\begin{cases} \dot{x} = v_x \\ \dot{y} = v_y \\ \dot{z} = v_z \\ \dot{v}_x = \left[(\cos\phi \sin\theta \cos\psi + \sin\phi \sin\psi)\, U_1\right]\big/ m \\ \dot{v}_y = \left[(\cos\phi \sin\theta \sin\psi - \sin\phi \cos\psi)\, U_1\right]\big/ m \\ \dot{v}_z = \left[-mg + (\cos\phi \cos\theta)\, U_1\right]\big/ m \end{cases} \tag{1}$$

$$\begin{cases} \dot{\phi} = p \\ \dot{\theta} = q \\ \dot{\psi} = r \\ \dot{p} = \left[U_2 + qr\,(I_y - I_z)\right]\big/ I_x \\ \dot{q} = \left[U_3 + pr\,(I_z - I_x)\right]\big/ I_y \\ \dot{r} = \left[U_4 + pq\,(I_x - I_y)\right]\big/ I_z \end{cases} \tag{2}$$

where (1) denotes the translational model and (2) denotes the rotational model. (x, y, z) is the absolute position in inertial frame, (v_x, v_y, v_z) is the corresponding velocity. (ϕ, θ, ψ) is the Euler angle of body-fixed frame with respect to inertial frame, (p, q, r) is the angular velocity in body-fixed frame. U_1 is the total thrust by rotors and (U_2, U_3, U_4) is the control torque vector. (I_x, I_y, I_z) is the moment of inertia along each axis and m is the mass.

B. CONSTRAINT

To achieve of goal of cooperative trajectory planning, several constraints should be taken into consideration. They mainly include state constraint, control constraint, boundary constraint and no-fly zone constraint. The details are given in the following.

1) STATE CONSTRAINT

State constraint means that the state of each UAV should meet engineering requirements. Generally, the state constraint mainly include position constraint, velocity constraint, Euler angle constraint and angular velocity constraint, which is formulated as

$$\begin{cases} \underline{x} \le |x| \le \bar{x},\ \underline{y} \le |y| \le \bar{y},\ \underline{z} \le |z| \le \bar{z} \\ |v_x| \le \bar{v}_x,\ \left|v_y\right| \le \bar{v}_y,\ |v_z| \le \bar{v}_z \\ |\phi| \le \bar{\phi},\ |\theta| \le \bar{\theta},\ |\psi| \le \bar{\psi} \\ |p| \le \bar{p},\ |q| \le \bar{q},\ |r| \le \bar{r} \end{cases} \tag{3}$$

where $\left(\underline{x}, \underline{y}, \underline{z}\right)$ and $(\bar{x}, \bar{y}, \bar{z})$ are respectively the lower and upper bounds for UAV position, $(\bar{v}_x, \bar{v}_y, \bar{v}_z)$, $(\bar{\phi}, \bar{\theta}, \bar{\psi})$, $(\bar{p}, \bar{q}, \bar{r})$ are the maximum bounds for UAV velocity, Euler angle and angle velocity.

2) CONTROL CONSTRAINT

Due to physical limit and control stability, UAV control thrust and torque should be limited within a reasonable range, which is

$$\begin{cases} |U_1| \le \bar{U}_1 \\ |U_2| \le \bar{U}_2 \\ |U_3| \le \bar{U}_3 \\ |U_4| \le \bar{U}_4 \end{cases} \tag{4}$$

where $\bar{U}_1$ and $\left(\bar{U}_2, \bar{U}_3, \bar{U}_4\right)$ are the maximum bounds of control vector.

3) BOUNDARY CONSTRAINT

Boundary constraint is usually used to denote UAV state limit in both endpoints of a trajectory. In this research, the position boundary for starting and ending points is denoted as (x_0, y_0, z_0) , $\left(x_f, y_f, z_f\right)$, the velocity boundary for the two points is denoted as $\left(v_{x0}, v_{y0}, v_{z0}\right)$, $\left(v_{xf}, v_{yf}, v_{zf}\right)$, the Euler angle boundary for the two points is denoted as $(\phi_0, \theta_0, \psi_0)$, $\left(\phi_f, \theta_f, \psi_f\right)$ and the angular velocity boundary is denoted as (p_0, q_0, r_0), $\left(p_f, q_f, r_f\right)$.

4) COLLISION AVOIDANCE CONSTRAINT

Each UAV should not conflict with others during the flying, namely, a relative safe distance should be maintained at a same time moment among all UAVs. In this research, a three-axis constraint is added to avoid collision, which is formulated as

$$\begin{cases} \left|x_i(t) - x_j(t)\right| \geq d_x \\ \left|y_i(t) - y_j(t)\right| \geq d_y \quad (i \neq j, i, j \in F) \\ \left|z_i(t) - z_j(t)\right| \geq d_z \end{cases} \tag{5}$$

where the subscripts i, j respectively denote the ith and jth UAV and $\left(d_x, d_y, d_z\right)$ is the safe distance among UAVs.

5) NO-FLY ZONE CONSTRAINT

Due to environment obstacle or threat, each UAV should avoid flying through some specific areas. For simplicity, the no-flying zone in this research is modeled as hemisphere with a finite radius, which can be described as

$$\sqrt{(x - X_i)^2 + (y - Y_i)^2 + (z - Z_i)^2} \geq d_i \ \ (i \in 1, 2, \ldots, N_{no}) \tag{6}$$

where N_{no} is the number of no-fly areas, (X_i, Y_i, Z_i) and d_i are respectively the center and radius of hemisphere.

C. OBJECTIVE

To establish an optimization problem, the objective for multi-UAV trajectory planning should also be determined. During the voyage of UAV, flying time is an important factor to be considered. Shorter flying time usually means reduced energy consumption and less probability of being discovered or attacked, thus the minimum flying time is selected as the optimization objective in this paper, which is given as follows

$$\min J = t_f \tag{7}$$

where t_f is the objective time to be optimized.

D. OPTIMIZATION STRATEGY DESIGN

In the following research, the multi-UAV trajectory planning is converted to a mathematical optimization problem, which aims to minimize the flying time t_f under equality constraints (1)~(2) and inequality constraints (3)~(6).

A two-layer hybrid optimization scheme is proposed in this paper to achieve the efficient trajectory planning for multi-UAV, as shown in Fig.2. The two layers are consisted by IPSO and GPM and are respectively designed for path planning and trajectory planning. In the first layer, by selecting path length as the fitness function, a feasible path with possible shortest length is obtained by IPSO. With the consideration of UAV velocity, the total flying time for the whole path and the moment for each path waypoint are calculated. Then in the second layer, a fitted curve by using generated path waypoints and time moments is constructed. Under the consideration of UAV dynamic model (1)~(2) and different constraints (3)~(6), the time minimum trajectory planning by using IPSO-GPM is accomplished. To further reduce procedure running time, a multi-segment strategy is also adopted in the following research.

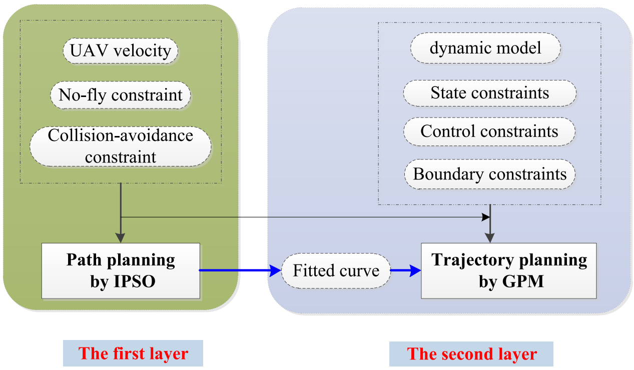


FIGURE 2. Optimization scheme for IPSO-GPM.

III. PATH PLANNING BY IPSO

In the following, the traditional PSO is firstly introduced. Then to improve the efficiency and optimality of traditional PSO, the IPSO is proposed and analyzed.

A. TRADITIONAL PSO

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. For path planning problem, each particle corresponds to a possible path and is usually consisted by a series of path waypoints. The number of path waypoints equals to the dimension of the particle. In this paper, the number of path waypoints is set as D. The number of total particles, namely the swarm size, is set as S. Then the position and velocity vector for the ith particle in three-dimensional space can be written as

$$\begin{aligned} \mathbf{p}_i &= \left(\mathbf{p}_{i,1}, \ldots, \mathbf{p}_{i,D}\right)^T \\ &= \left(\left(p_{i,1}^x, p_{i,1}^y, p_{i,1}^z\right), \ldots, \left(p_{i,D}^x, p_{i,D}^y, p_{i,D}^z\right)\right)^T \end{aligned} \tag{8}$$

$$\begin{aligned} \mathbf{v}_i &= \left(\mathbf{v}_{i,1}, \ldots, \mathbf{v}_{i,D}\right)^T \\ &= \left(\left(v_{i,1}^x, v_{i,1}^y, v_{i,1}^z\right), \ldots, \left(v_{i,D}^x, v_{i,D}^y, v_{i,D}^z\right)\right)^T \end{aligned} \tag{9}$$

where $\left(p_{i,j}^x, p_{i,j}^y, p_{i,j}^z\right)$, $\left(v_{i,j}^x, v_{i,j}^y, v_{i,j}^z\right)$, $j \in \ 1, \ldots, D$ are the jth path waypoint's position and velocity respectively of the ith particle. Then the whole swarm can be expressed as

$$\left(\left(\mathbf{p}_1, \mathbf{v}_1\right), \left(\mathbf{p}_2, \mathbf{v}_2\right), \ldots, \left(\mathbf{p}_S, \mathbf{v}_S\right)\right) \tag{10}$$

For PSO with S particles, there are also S personal best positions and one global best position, which are

$$\begin{aligned}\mathbf{P}_{i,best} &= \left(\mathbf{p}_{i,1,best}, \ldots, \mathbf{p}_{i,D,best}\right)^T \\ &= \left(\left(p^x_{i,1,best}, p^y_{i,1,best} p^z_{i,1,best}\right), \right. \\ &\quad \left. \ldots, \left(p^x_{i,D,best}, p^y_{i,D,best} p^z_{i,D,best}\right)\right)^T \qquad (11)\\ \mathbf{G}_{best} &= \left(\mathbf{g}_{1,best}, \ldots, \mathbf{g}_{D,best}\right)^T \\ &= \left(\left(p^x_{1,best}, p^y_{1,best} p^z_{1,best}\right), \right. \\ &\quad \left. \ldots, \left(p^x_{D,best}, p^y_{D,best} p^z_{D,best}\right)\right)^T \qquad (12)\end{aligned}$$

where $i \in \;\; 1, \ldots, S$ is the particle label. Then the velocity and position of each particle are updated according to the following formula [27], [28]

$$\begin{cases}\mathbf{v}_{i,j}(t+1) = w \cdot \mathbf{v}_{i,j}(t) + c_1 \cdot r_1 \cdot \Delta_1 + c_2 \cdot r_2 \cdot \Delta_2 \\ \mathbf{p}_{i,j}(t+1) = \mathbf{p}_{i,j}(t) + \mathbf{v}_{i,j}(t+1)\end{cases} \qquad (13)$$

where $\Delta_1 = \left(\mathbf{p}_{i,j,best}(t) - \mathbf{p}_{i,j}(t)\right)$, $\Delta_2 = \left(\mathbf{g}_{j,best}(t) - \mathbf{p}_{i,j}(t)\right)$ and $t \in \;\; 1, \ldots, T$ is the current iteration with T denoting the total iterations. w is the inertia weight for balancing global and local searching, c_1, c_2 are two constant acceleration coefficients reflecting the ability of learning from global best particle and personal best particles. r_1, r_2 are two random values distributed in $(0, 1)$. The updating velocity is usually restricted by $[-V_{\max}, V_{\max}]$. In order to accelerate convergence, the following linear-varying inertia weight is usually adopted [25]

$$w = w_{\max} - \frac{(w_{\max} - w_{\min})\, t}{T} \qquad (14)$$

B. PERFORMANCE ANALYSIS

The traditional PSO possesses the convenience of few parameters and easy implementation. However, some drawbacks also accompany with traditional PSO, such as it may fall into local minimum and premature convergence. Besides, the total iterations needed to achieve global satisfactory solution is usually quite large and the whole evaluation process can be improved.

In this paper, the path waypoints generated by PSO are adopted as initial values for GPM, thus the quality of the generated path waypoints would have significant influence on the performance of GPM. Theoretically, better PSO outputs will result in better planned trajectory by GPM. How to generate PSO path waypoints with better optimality and less iterations would be a key to improve the efficiency of GPM trajectory.

During PSO evolution, searching and convergence are the two main objectives. Searching means each particle should enrich the diversity of the swarm and convergence means all particles should finally achieve a same optimal or near-optimal solution after certain iterations. In the design of PSO, it is reasonable to divide the whole process into two phases, the first half phase should focus on searching and maintain diversity while the latter half phase should focus on enhancing convergence and optimality. From (13), it is easy to see that the acceleration coefficients c_1, c_2 and the maximum velocity $V_{\max}$ are key parameters to influence the evolution process. At the first half phase, the searching range would be enlarged and the swarm diversity would be benefitted if relatively large c_1 and $V_{\max}$ are chosen. At the latter half phase, the convergence to global optimal or near-optimal solution and the convergence speed would be enhanced if relatively large c_2 and relatively small $V_{\max}$ are chosen.

Apart from above parameters, appropriate modification on position updating topology can also improve PSO performance. For traditional PSO, all particles would be involved into the position updating process. However, there are always some particles inheriting poor performance during the process. Taking such particles into position updating would increase total iterations and decrease algorithm efficiency. So it is meaningful and beneficial to abandon such particles and take advantage of the good particles only, which would theoretically improve the solution property.

C. IMPROVED PSO

In order to achieve the global optimal or near-optimal solution in a more quick and accurate way, the IPSO is proposed in this section. Three improvements for IPSO are given as follows:

1) ADAPTIVE PARAMETER ADJUSTMENT

Acceleration coefficients c_1, c_2 are two critical parameters for PSO. They indicate the weights of acceleration terms that pull each particle to local and global best positions and play important roles in adjusting convergence speed and searching direction. From the view of swarm evaluation, the PSO would maintain better performance if the coefficients are adaptively adjusted with the evaluation process.

According to PSO searching and evaluation process, it is reasonable to conclude that searching should predominate the first half phase while convergence should predominate the latter half phase. Thus, c_1, c_2 are adaptively designed as

$$c_1 = c_{\max} - \frac{(c_{\max} - c_{\min})\, t}{T} \qquad (15)$$

$$c_2 = c_{\min} + \frac{(c_{\max} - c_{\min})\, t}{T} \qquad (16)$$

where $c_{\max}, c_{\min}$ are constant values with $c_{\max} > c_{\min} > 0$. In simulations, $c_{\max}, c_{\min}$ are set according to the equation $(c_{\max} + c_{\min})/2 = c_1^c = c_2^c$, where $c_1^c = c_2^c$ denote the values in traditional PSO. This principle guarantees that the average of c_1 and c_2 for IPSO are the same as they are for traditional PSO, and makes the comparison much fair and well-founded.

For (15)$\sim$(16), it is obvious that c_1 and c_2 are respectively linearly decreasing and increasing. The sum satisfies $c_1 + c_2 = c_{\max} + c_{\min}$, meaning that the ability sum for particle searching and convergence is constant during the evolution. It is also obvious that $c_1 > c_2$ is satisfied at the first half phase, thus the searching diversity is strengthened and it is beneficial for fast obtaining optimal solution and avoiding local minimum. While $c_1 < c_2$ is satisfied at the latter half phase, it has the benefit of fast convergence to a same global optimal solution. In a word, the adaptive coefficients would improve the optimality and rapidity of PSO.

2) MAXIMUM VELOCITY DESIGN

Maximum velocity $V_{\max}$ is also an important factor in influencing PSO performance, which has been seldom paid attention to by researchers. Generally, constant $V_{\max}$ is not sufficient to precisely reflect the searching diversity and accuracy in different evolution stages. During PSO evolution, it is always expected that searching range should be large enough to enrich solution diversity at the beginning and it should be relatively small to improve the accuracy of optimal solution at the end. Thus, a simple but effective linear-varying $V_{\max}$ is designed as follows

$$V_{\max} = V_1 - \frac{(V_1 - V_2)\,t}{T} \tag{17}$$

where $V_1 > V_2 > 0$ are constant values. In the following simulations, V_1 and V_2 are selected under the equation $(V_1 + V_2)/2 = 2V_{\max}^c$, where $V_{\max}^c$ is the constant value for traditional PSO.

It is obvious that $V_{\max} = V_1$ when $t = 0$ and $V_{\max} = V_2$ when $t = T$. The designed $V_{\max}$ can linearly decrease from V_1 to V_2 during the evolution. This strategy takes both searching range and accuracy into consideration. At the beginning stage, the solution diversity is enriched and the probability of meeting global optimal solution is increased. At the end stage, the velocity with smaller value can help to refine optimal solution by smaller searching range and remove unnecessary searching of non-optimal solutions. It is reasonable to conclude that the adaptive adjustment of maximum velocity guarantees better performance than constant maximum velocity.

3) POSITION UPDATING STRATEGY

Position updating strategy is also a main factor to improve PSO. Some researchers have introduced adaptive sensitivity decision operator to update the best position, but additional models would be brought in and computation complexity would be increased [26]. In order to improve convergence speed in a simple way, a new position updating strategy is proposed in the following.

To implement the strategy, we should firstly carry out PSO for one time and calculate fitness values of all particles. Then rearrange all particles in a descending order according to the fitness values. The position updating scheme is designed as

$$\begin{cases} \mathbf{p}_{l\arg e} = \mathbf{p}_{small} + a \cdot r_3 \\ \mathbf{v}_{l\arg e} = \mathbf{v}_{small} \end{cases} \tag{18}$$

where $\mathbf{p}_{l\arg e}$ and $\mathbf{p}_{small}$ respectively represent the half particles with larger and smaller fitness values, r_3 is a random number in $[0, 1]$, a is a constant value, reflecting position offset of the newly obtained mutant particles with respect to the particles with smaller fitness value. Ideally, a should be relatively large at the beginning and becomes increasingly smaller as the evolution proceeds. For brevity, the value in simulation is set constant.

The equation (18) means that particles with larger fitness values will be replaced by those with smaller fitness values using an offset value, and particle velocities with larger fitness values would remain unchanged. The reasonability for adopting such scheme can be explained as follows.

The particles with large fitness values means that they distribute far from the expected path or some of the waypoints are within the threat or collision areas, thus it would be meaningless and time-consuming for subsequent evolution, so we propose the position updating strategy to accelerate the convergence. Meanwhile, in order to keep the diversity of the swarm, an offset value a is added for position replacement. This proposed strategy would be effective in improving convergence speed and maintaining swarm diversity.

REMARK 1: For traditional PSO and IPSO, it is obvious that the main structures are the same, but the IPSO refines parameter settings using (15)~(18). By qualitative analysis, the IPSO enhances both convergence speed and solution optimality. Meanwhile, the procedure running time for the two PSOs is almost the same in theory because they maintain the same updating structure.

Based on above improvements, the IPSO can be adopted for path planning of multi-UAV. To evaluate the process, the length of the path is set as the fitness function. The collision avoidance with obstacles and other UAVs are taken as constraints. The implementation process for path planning by IPSO is the same as that by PSO, which is simple and omitted here. For more details on PSO path planning, one can refer to [24], [25].

IV. TRAJECTORY PLANNING BY IPSO-GPM

In this section, the IPSO-GPM trajectory planning for multi-UAV is proposed. Firstly, the GPM principle is introduced. Then, the procedures for the hierarchical algorithm are presented. Finally, a multi-segment strategy is also proposed.

A. GPM PRINCIPLE

In GPM, the continuous NOC is firstly converted into a NLP problem by approximating states and controls using Lagrange interpolating polynomials, then it is solved by an appropriate numerical solver. The procedures for different pseudo-spectral methods are similar and the readers can refer to [17] and [34] for details.

1) TIME DOMAIN TRANSFORMATION

The UAV flying time from t_0 to t_f is transformed into $\tau \in [-1, 1]$ in the GPM domain with the transformations as follows

$$\tau = \frac{2t}{t_f - t_0} - \frac{t_f + t_0}{t_f - t_0} \tag{19}$$

2) APPROXIMATION OF SYSTEM VARIABLES

GPM chooses the Kth-order Legendre-Gauss points and $\tau_0 = -1$ as the collocation points, and constructs

$(K+1)$ Lagrange interpolating polynomials denoted by $L_i(\tau)$ $(i = 0, 1, 2, \ldots, K)$. Then the state and control variables are approximated by

$$\begin{cases} x(\tau) \approx X(\tau) = \sum_{i=0}^{K} L_i(\tau) X(\tau_i) \\ u(\tau) \approx U(\tau) = \sum_{i=1}^{K} L_i^*(\tau) X(\tau_i) \end{cases} \tag{20}$$

where $L_i(\tau)$ and $L_i^*(\tau)$ are the interpolating basis functions and denoted as

$$\begin{cases} L_i(\tau) = \prod_{j=0, j\neq i}^{K} \dfrac{\tau - \tau_j}{\tau_i - \tau_j} \\ L_i^*(\tau) = \prod_{j=1, j\neq i}^{K} \dfrac{\tau - \tau_j}{\tau_i - \tau_j}, \end{cases} \quad i = 0, 1, \ldots, K \tag{21}$$

3) TRANSFORMATION OF EQUATION CONSTRAINT

Differentiating (20) yields

$$\dot{x}(\tau) \approx \dot{X}(\tau) = \sum_{i=0}^{K} \dot{L}_i(\tau) X(\tau_i) = \sum_{i=0}^{K} D_{ki} X(\tau_i) \tag{22}$$

where $D_{ki} \in R^{K\times(K+1)}$ is a differential approximation matrix and expressed as follows

$$D_{ki}(\tau) = \begin{cases} \dfrac{(1+\tau_K) P_K(\tau_k) + P_K(\tau_k)}{(\tau_K - \tau_i)\left[(1+\tau_i) P_K(\tau_i) + P_K(\tau_i)\right]}, & i \neq k \\ \dfrac{(1+\tau_i) \ddot{P}_K(\tau_i) + 2P_K(\tau_i)}{2\left[(1+\tau_i) P_K(\tau_i) + P_K(\tau_i)\right]}, & i = k \end{cases} \tag{23}$$

So the differential equation can be transformed into the following algebraic constraint

$$\sum_{i=0}^{K} D_{ki} X(\tau_i) - \frac{t_f - t_0}{2} f(X(\tau_k), u(\tau_k), \tau_k) = 0 \tag{24}$$

4) TRANSFORMATION OF TERMINAL STATE

The final point when $\tau_f = 1$ is not concluded in the Legendre-Gauss collocations, so we calculate it as follows

$$x(\tau_f) = x(\tau_0) + \int_{-1}^{1} f(X(\tau), U(\tau), \tau) d\tau \tag{25}$$

By using Gauss integration, it yields

$$x(\tau_f) = x(\tau_0) + \frac{t_f - t_0}{2} \sum \omega_k f(X(\tau_k), U(\tau_k), \tau_k) \tag{26}$$

where $\omega_k = \int_{-1}^{1} L_i(\tau) d\tau$ is the Gauss weight.

According to (19)~(26), the NOC problem can finally be converted to a NLP and can be solved by an appropriate solver.

B. PROCEDURE FOR IPSO-GPM

The key of multi-UAV trajectory planning by hierarchical IPSO-GPM is to introduce a fitted curve according to IPSO to serve as the initial values of GPM. Thus, the IPSO is firstly regarded as a start engine to generate the fitted curve. Because time domain is not included in IPSO, the shortest path distance is regarded as IPSO objective. A feasible flying time is then chosen with the consideration of UAV velocity and the length sum of generated path. With the feasible time and a series of best path waypoints, the curve fitting operation is implemented to generate a time-related curve for GPM. Then, the GPM carries out the time minimum optimization problem modeled by (7) until all the constraints (1)~(6) are satisfied. The detailed procedures for IPSO-GPM are as follows:

STEP 1: Construct the 3D mission area. Set the initial and final position for UAVs.

STEP 2: Design IPSO parameters, including swarm size S, waypoint number D, inertia weight parameters $w_{\max}$, $w_{\min}$, acceleration coefficient parameters $c_{\max}$, $c_{\min}$, velocity parameters V_1, V_2 and maximum iterations T.

STEP 3: Set UAV state and control constraints, which include path constraint (3), control constrain (4), collision avoidance constraint (5), no-fly constraint (6) and boundary constraint.

STEP 4: Choose the path length by connecting generated path waypoints as the IPSO fitness function.

STEP 5: Calculate the fitness function and figure out the personal best position and global best position.

STEP 6: Sort the fitness values of all particles and update particle position according to (18).

STEP 7: Evolve the IPSO and repeat step 4~ step 6 until the maximum iterations is satisfied.

STEP 8: Output the path waypoints of IPSO and choose feasible flying time to establish a fitted curve.

STEP 9: Switch the searching process to GPM. Set the collocation number N_G. Convert the trajectory planning problem to NLP and choose SNOPT as the solver.

STEP 10: Observe the output of GPM. If the result shows "Optimality conditions satisfied", then stop; otherwise modify N_G and go to step 9.

STEP 11: Finish the evolution and plot the trajectory.

C. MULTI-SEGMENT STRATEGY

Besides the initial values for GPM, the number of collocation points is another key factor that can influence the solution accuracy and convergence rapidity. Generally, more collocation points will contribute to more accurate solutions, but more running time is needed accordingly. How to reduce running time and maintain solution quality is vital for practical applications.

In order to reduce the running time and maintain the quality of output solutions, a multi-segment strategy for IPSO-GPM is proposed in this section. By dividing the fitted curve of IPSO into several segments, the IPSO-GPM is then adopted to each segment repeatedly. The whole trajectory would be

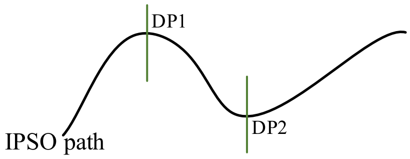


FIGURE 3. Illustrations of dividing the curve into 3 segments.

generated by connecting neighboring segments. To implement the strategy, the number of segments and the dividing point position should be chosen appropriately. Actually, how to choose the dividing point position and segment number is an NP optimization problem. The solving of this problem is even more difficult than that of the trajectory planning problem itself, thus a simple but feasible operation for choosing the two parameters is given as follows.

Firstly, a fitted curve should be obtained on the basis of IPSO path waypoints and feasible flying time. Then, the dividing point position is mainly dependent on the curvature of the fitted curve. With the aid of parametric equation in X-Y-Z axis, the curvature of the fitted curve could be easily conducted. When the curvature of some point is larger than a preset constant, the corresponding point is regarded as the dividing point. Finally, if the total number of dividing points is n, then the number of segments would be $n+1$. Fig. 3 gives an example of dividing the fitted curve into three segments with DP1 and DP2 denoting dividing points.

The reason why adopt curvature to divide the fitted curve is explained as follows. In GPM, the collocation points are distributed densely on the both sides of the designed trajectory and sparsely in the middle. When the curvature of the fitted curve is large in some specific points, more collocation points are needed theoretically so as to improve the accuracy of the planned trajectory. By choosing dividing points according to curvature, more collocation points would be distributed near the dividing points.

REMARK 2: Practically, to avoid the calculation of curvature, the dividing points and the number of segments can also be chosen according to the path length or complexity of the optimization problem. Comparing with one-segment GPM, the multi-segment GPM would theoretically reduce total running time with the same number of collocation points. To weaken the possible influence on solution optimality by multi-segment GPM, a balance should also be established between solution optimality and the number of curve segments. By appropriately choosing the number of curve segments, the multi-segment strategy can even achieve near-online results. Taking a long-flying-time mission as an example, the first segment can be generated via offline planning, while all other following segments can be generated online during the period that UAV tracks the previous segment trajectory, which would make the planning time-saving and can also accommodate moving obstacles simultaneously.

V. SIMULATION ANALYSIS

In this section, numerical simulations are carried out to illustrate the effectiveness of the proposed strategy. The algorithms are implemented in MATLAB environment and are run on a PC with 3.6GHz CPU, 4.00GB of RAM and 64-bit operating system. The SNOPT solver is adopted for solving NLP in the simulation.

TABLE 1. Obstacle parameters.

Label	Center position(m)	Radius(m)
Obstacle 1	(120,100,0)	37
Obstacle 2	(60,70,0)	30

TABLE 2. Boundary constraints for each UAV.

Label (m)	SP (m)	FP (m)	SV (m/s)	FV (m/s)	SA (rad)	FA (rad)	SAV (rad/s)	FAV (rad/s)
UAV 1	(8,70,2)	(200,70,30)	**0**	**0**	**0**	N/A	**0**	N/A
UAV 2	(8,100,2)	(200,60,20)	**0**	**0**	**0**	N/A	**0**	N/A
UAV 3	(8,130,2)	(200,70,10)	**0**	**0**	**0**	N/A	**0**	N/A

where SP, SV, SA and SAV respectively represent starting position, starting velocity, starting attitude and starting angular velocity, FP, FV, FA and FAV respectively represent final position, final velocity, final attitude and final angular velocity.

Firstly, the parameters for flying environment and UAV are introduced. Three identical quadrotor UAVs are supposed to fly within the area of 200 × 200 × 100m, and there are two hemisphere obstacles inside. The mass for each UAV is $m = 0.625kg$ and moment of inertia along each axis is $2.3 \times 10^{-3} kg \cdot m^2$. The detailed parameters for obstacles are shown in Table 1.

To specify the optimization problem, the necessary UAV constraints are given in the following. Firstly, the boundary constraints are presented in Table 2.

Besides boundary constraints, the state constraints for UAV are given as

$$\begin{cases} 8 \le x \le 200,\ 20 \le y \le 200,\ 2 \le z \le 40\,(m) \\ -5 \le v_x \le 5,\ -5 \le v_y \le 5,\ -5 \le v_z \le 5\,(m/s) \\ |\phi| \le \pi/2,\ |\theta| \le \pi/2,\ |\psi| \le \pi/2\,(rad) \\ |p| \le 2,\ |q| \le 2,\ |r| \le 2\,\left(rad/s\right) \end{cases}$$

The control constraints for UAV are set as

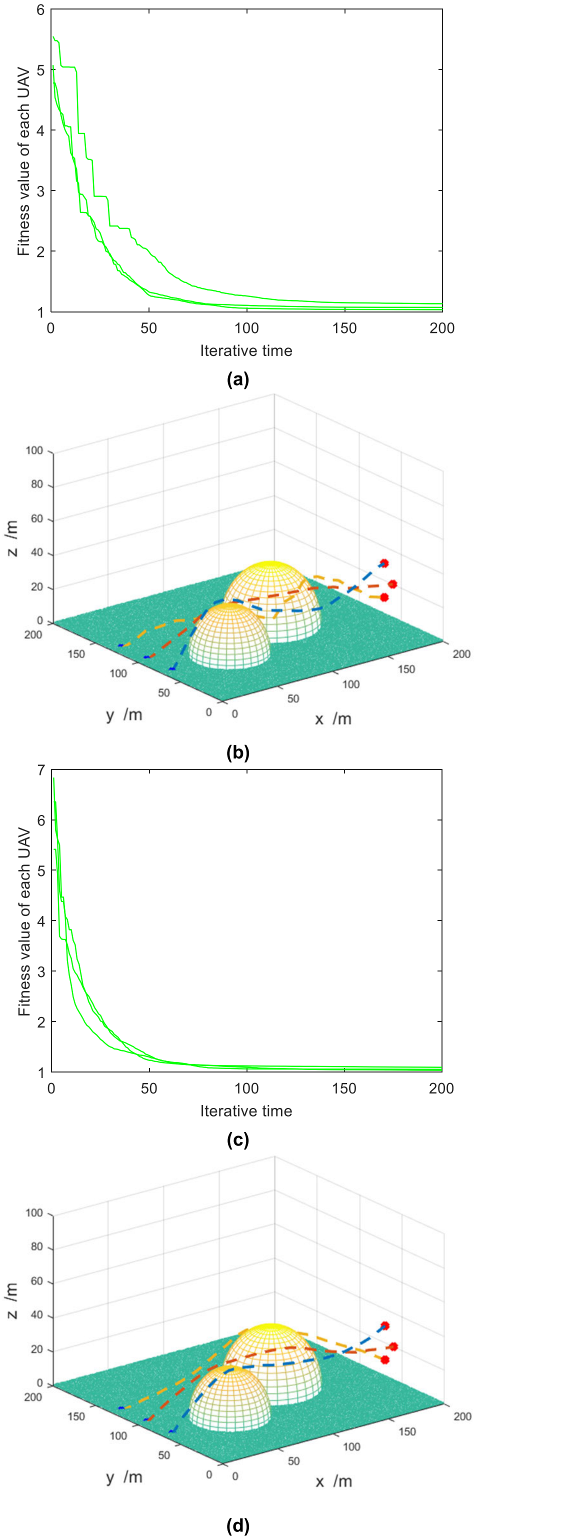
$$|U_1| \le 21N,\ |U_2| \le 0.67N \cdot m,\ |U_3| \le 0.67N \cdot m,\ |U_4| \le 0.11N \cdot m$$

The safe distance for each UAV is

$$d_x = d_y = d_z = 1m$$

A. PERFORMANCE OF IPSO

The path planning effectiveness of IPSO is firstly illustrated. The parameters in simulation are listed as follows: the swarm size is $S = 500$, the maximum iterations is $T = 200$, the inertia weight is $\omega_{\max} = 0.9$, $\omega_{\min} = 0.4$, the waypoint number is $D = 20$, the specific parameters for traditional PSO are $c_1 = c_2 = 2$, $V_{\max} = 2$ while those for IPSO are $c_{\max} = 3.5$, $c_{\min} = 0.5$, $V_1 = 3.5$, $V_2 = 0.5$. Path planning comparisons by the PSO and IPSO are shown in Figs. 4∼5.



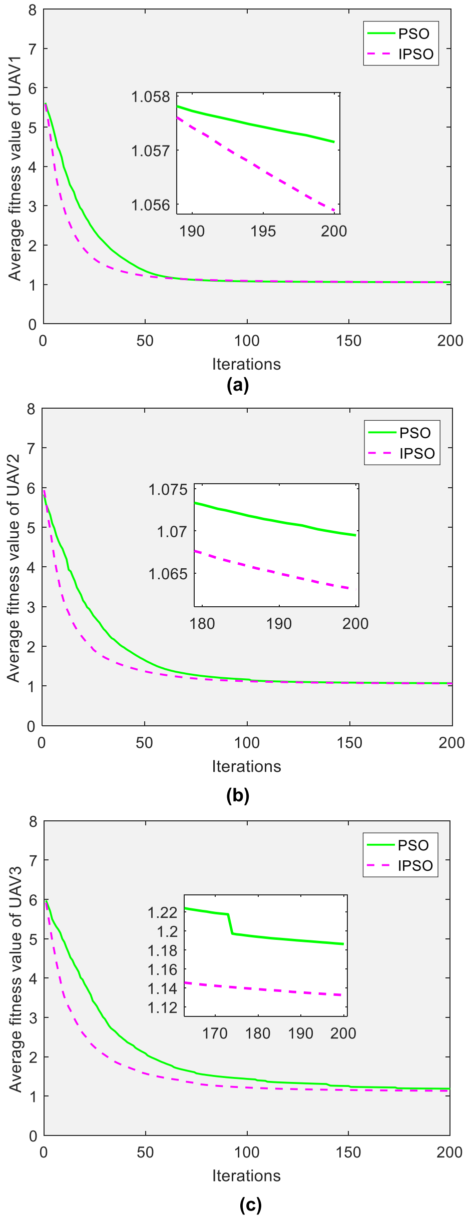
(a)

(b)

(c)

(d)

FIGURE 4. **Path planning comparisons between PSO and IPSO. (a) Fitness value of PSO. (b) Generated path by PSO. (c) Fitness value of IPSO. (d) Generated path by IPSO.**



(a)

(b)

(c)

FIGURE 5. **The 100-run average fitness value for each UAV. (a) UAV1. (b) UAV 2. (c) UAV 3.**

Fig. 4 shows the fitness value evaluations and three-dimension paths generated by PSO and IPSO. The fitness value is measured by the ratio of path distance to the line distance from starting position to final position and a unit penalty term is added when UAV conflicts with obstacles or other members. It is easy to see that both PSO and IPSO can guarantee feasible paths, but the fitness value of IPSO is smoother and converges faster than that of PSO. To further make fair comparison, we carry out the 100-run Monte Carlo simulation for PSO and IPSO respectively. Fig. 5 is the average fitness value comparison between PSO and IPSO, where the horizontal axis denotes the iterations and the vertical axis denotes average fitness value of each UAV. One can see that the IPSO guarantees faster convergence than PSO, the final fitness value of IPSO is also lower than that of PSO. It should be noted that the environment constraints in this scenario are relatively simple, thus the IPSO does not show obvious advantage in the final fitness values. When more complex environment constraints are taken into

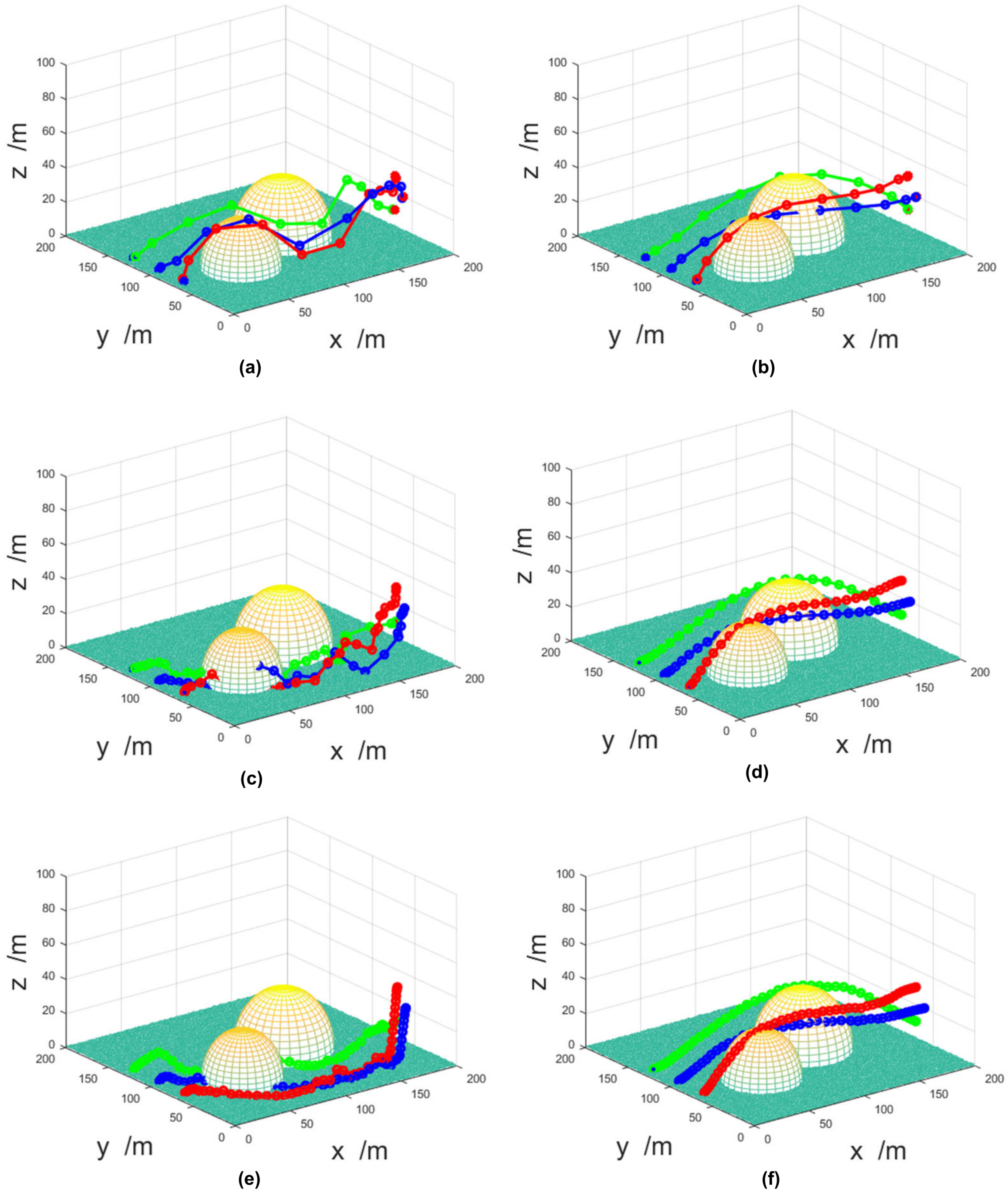


FIGURE 6. Trajectory comparisons between GPM and IPSO-GPM. (a) 10 points by GPM. (b) 10 points by IPSO-GPM. (c) 30 points by GPM. (d) 30 points by IPSO-GPM. (e) 50 points by GPM. (f) 50 points by IPSO-GPM.

consideration, the IPSO would show more obvious advantage than PSO. All the same, one can conclude from Figs. 4∼5 that the IPSO guarantees faster and better solution than PSO.

B. PERFORMANCE OF IPSO-GPM

The IPSO-GPM is implemented for simulation in this section. The initial values for GPM is firstly constructed via adopting curve fitting on the path waypoints generated by IPSO. UAV flying time used in the fitted curve is supposed to be 40 seconds with the consideration of path distance and maximum velocity. Other initial values for GPM, such as initial velocity and control thrust are respectively set as zero and 5N to obtain a favorable trajectory. In the following, different simulations with 10, 20, 30, 40, 50 and 60 collocation points are respectively carried out to make fair and well-founded comparisons between GPM and IPSO-GPM and the results are presented in Table 3 and Figs. 6∼7.

Table 3 gives the comparison summary between GPM and IPSO-GPM, in which "Time" means the procedure running time by using SNOPT solver. It can be found that although the objective time for different simulations are all 39.0362 seconds, the procedure running time for IPSO-GPM is much less than that by GPM. Besides, for simulations with 20,30,40,50 and 60 collocation points, the returning result by IPSO-GPM is "Optimal conditions satisfied" while that by GPM is "Iteration limit reached", which means that the proposed IPSO-GPM can successfully solve the optimization problem while traditional GPM does not have the capability.

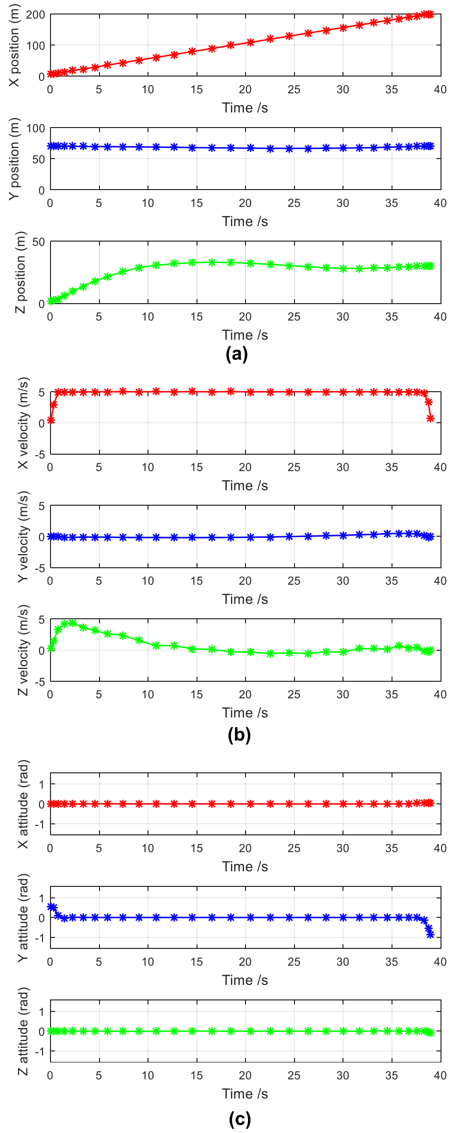


FIGURE 7. Intermediate variables for UAV 1 by IPSO-GPM. (a) Position. (b) Velocity. (c) Attitude angle.

TABLE 3. Comparison between GPM and IPSO-GPM.

Point Number	10	20	30	40	50	60
GPM Result	NIM	ILR	ILR	ILR	ILR	ILR
GPM Time(s)	3.2744	9.6301	28.9294	54.8527	132.9086	386.9146
GPM Objective(s)	39.0362	39.0362	39.0362	39.0362	39.0362	39.0362
IPSO-GPM Result	NIM	OCS	OCS	OCS	OCS	OCS
IPSO-GPM Time(s)	0.5832	1.0684	2.2449	5.7224	17.5921	25.8487
IPSO-GPM Objective(s)	39.0362	39.0362	39.0362	39.0362	39.0362	39.0362

where NIM means "Nonlinear infeasibilities minimized", ILR means "Iteration limit reached", OCS means "Optimality conditions satisfied".

Apart from Table 3, detailed comparisons are also carried out. The generated trajectories by the two methods are presented in Fig. 6, in which only the results of 10, 30 and 50 collocation points are given for space limit. One can see that the trajectories generated by IPSO-GPM are all flyable and can avoid conflicting with obstacles, while those by GPM with 30 and 50 collocation points conflict with the obstacles.

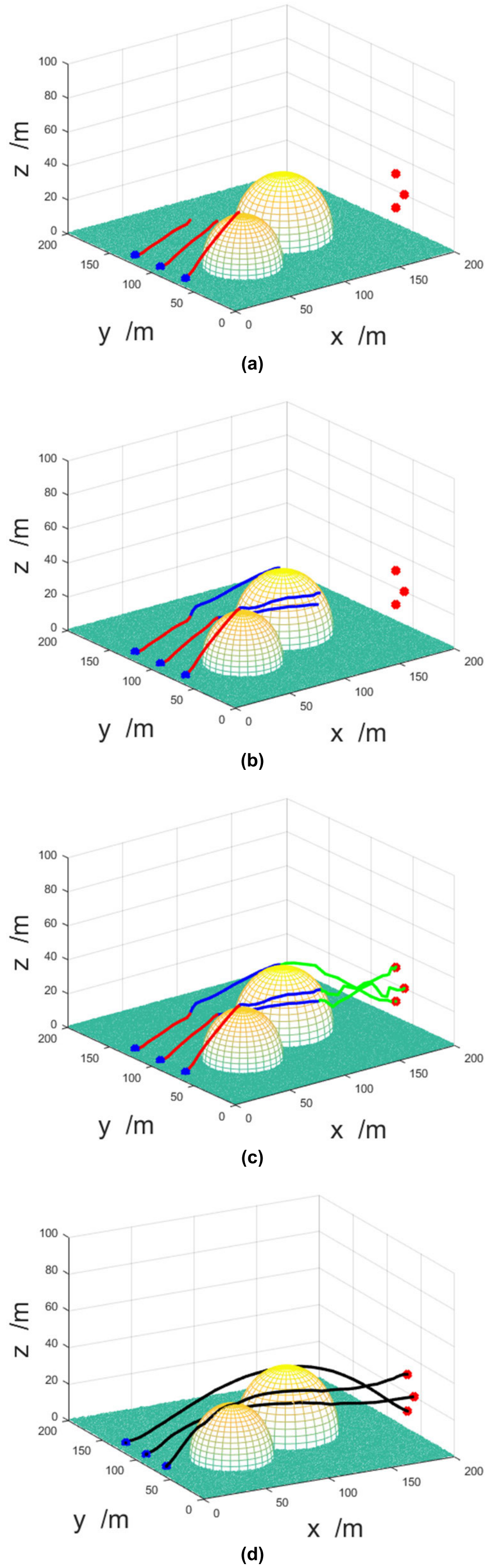


FIGURE 8. Segmented trajectory generation and comparison. (a) The first segmented trajectory. (b) The first two segmented trajectories. (c) The three segmented trajectories. (d) The one-segment trajectory.

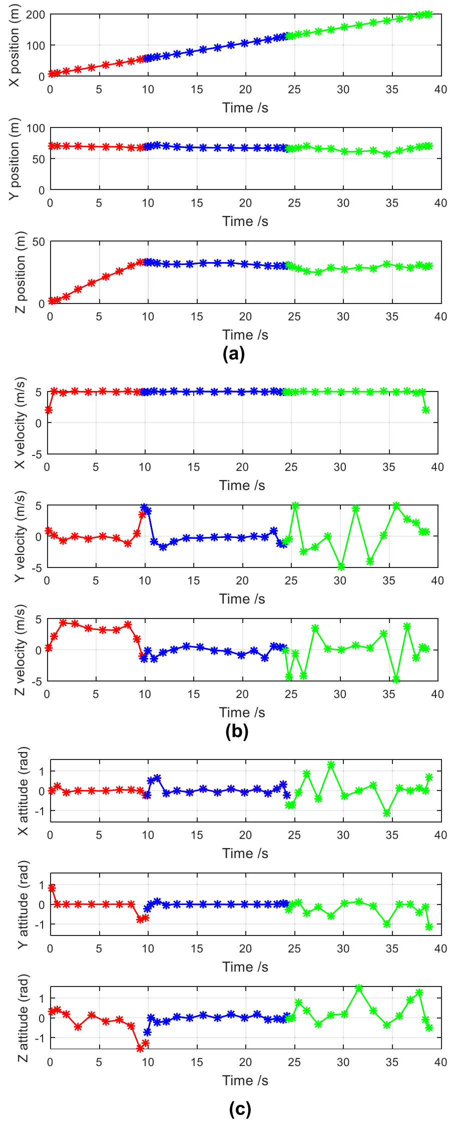


FIGURE 9. **Intermediate variables for UAV 1 using multi-segment strategy. (a) Position. (b) Velocity. (c) Attitude angle.**

As for the 10-collocation-point simulation, both the methods can generate feasible trajectories, but the returning results in Table 3 display ''Nonlinear infeasibilities minimized'', indicating more collocation points should be added to achieve better trajectory optimality.

Besides the 3-D trajectory, other intermediate variables including UAV states and control vector are also provided and shown in Fig. 7. Since the variables of different UAVs evaluate in a similar fashion, only the results of UAV 1 is given in the limited space. One can see from Fig. 7 that all variables are within the constraint bounds. In order to guarantee minimum flying time and maintain consistent with constraints, the velocity in X-axis achieves almost the maximum allowable velocity during the process. From Fig. 7, we can also easily deduce that the generated trajectory and intermediate variables can offer good guidance for the following control system.

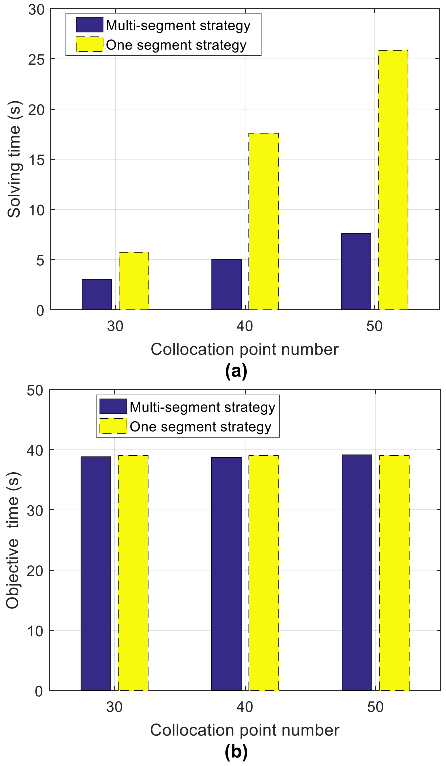


FIGURE 10. **Time comparisons for one-segment and multi-segment strategies. (a) Solving time. (b) Objective time.**

C. PERFORMANCE OF MULTI-SEGMENT STRATEGY

The multi-segment IPSO-GPM is carried out in this section. To simplify the dividing process, the fitted curve is divided into three segments according to the flying distance. In simulations, the position, velocity and attitude of the terminal point of last segment is set the same as those of the start point of next segment. The following Figs. 8∼10 and Table 4 give the results of multi-segment strategy.

In the results, the three segments are respectively marked by red, blue and green color. To make a fair comparison, the total number of collocation points for the three segments is set the same as that for the one-segment trajectory. In Fig. 8, the one-segment trajectory has 40 collocation points and the three segments respectively have 10, 15 and 15 collocation points. One can easily find that the multi-segment strategy can generate smooth and flyable trajectory similar with the one-segment trajectory. Fig. 9 shows the detailed intermediate variables for UAV 1. One can see that all variables are within the constraint bound, and some rapid changes only appear near the dividing points, which is also reasonable since the variables need to meet the boundary constraints in dividing points.

To further evaluate the performance of multi-segment strategy, simulation comparisons with 40, 50 and 60 collocation points are summarized in Table 4. One can see that all the three segments with different number of collocation points can achieve optimal solutions within a short solving time. To better show the rapidity of segmented planning, the sum of solving time and objective time by three-segment trajectory

TABLE 4. Results of multi-segment IPSO-GPM.

Total Points	40			50			60		
Segment	1st	2nd	3rd	1st	2nd	3rd	1st	2nd	3rd
Segment Point	10	15	15	15	20	15	15	25	20
Result	OCS	OCS	OCS	OCS	OCS	OCS	OCS	OCS	OCS
Solving time(s)	0.6940	1.1634	1.1792	1.6867	2.0995	1.2452	1.8516	4.3438	1.3860
Objective(s)	9.8529	14.4019	14.5852	9.7221	14.4012	14.5852	9.7221	14.4	15.0336

where OCS means "Optimality conditions satisfied".

and one-segment trajectory are compared in Fig. 10. One can see that the objective time by the two methods maintains almost the same, while the total solving time varies a lot. The total solving time of 40, 50 and 60 collocation points by segmented strategy is about a half, one-third and a quarter of that by one-segment strategy. So it is easy to conclude that the multi-segment strategy can greatly speed up the solving process and reduce the total solving time. With above comprehensive consideration, the multi-segment strategy improves the trajectory planning efficiency and is more time saving than the one-segment planning strategy.

Thus, taking all above analysis and comparisons into consideration, the designed IPSO-GPM and multi-segment strategy can significantly improve the planning efficiency.

VI. CONCLUSION

This paper investigates the hierarchical trajectory planning for quadrotor UAVs under obstacle environment. The IPSO, IPSO-GPM and multi-segment strategy are respectively designed and a satisfactory trajectory with higher efficiency can be obtained. The detailed findings and contributions are as follows:

1) Comparing with traditional PSO, the IPSO improves convergence speed and enhances the optimality of obtained solution via introducing adaptive varying parameters and position updating strategy.

2) By proposing IPSO-GPM and multi-segment strategy, the high-quality trajectory under 6-DOF model can be generated with much less procedure running time.

In the future, we will continue to research on the issue. How to quickly generate trajectory for UAV with dynamic obstacles would be interesting. Besides, considering the specific environment, how to adaptive choose collocation points number and position is meaningful for practical applications.

REFERENCES

[1] 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.

[2] 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.

[3] K. Harikumar, J. Senthilnath, and S. Sundaram, "Multi-UAV oxyrrhis 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] Á. Madridano, A. Al-Kaff, D. Martín, and A. de la Escalera, "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. Urtasun, "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, doi: 10.1016/j.isatra.2020.11.017.

[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.

[14] M. Zhang, J. Song, L. Huang, and C. Zhang, "Distributed cooperative search with collision avoidance for a team of unmanned aerial vehicles using gradient optimization," *J. Aerosp. Eng.*, vol. 30, no. 1, Jan. 2017, Art. no. 04016064.

[15] A. Shirazi, J. Ceberio, and J. A. Lozano, "Spacecraft trajectory optimization: A review of models, objectives, approaches and solutions," *Prog. Aerosp. Sci.*, vol. 102, pp. 76–98, Oct. 2018.

[16] X. Tang, Y. Shi, and L.-L. Wang, "A new framework for solving fractional optimal control problems using fractional pseudospectral methods," *Automatica*, vol. 78, pp. 333–340, Apr. 2017.

[17] I. M. Ross and M. Karpenko, "A review of pseudospectral optimal control: From theory to flight," *Annu. Rev. Control*, vol. 36, no. 2, pp. 182–197, Dec. 2012.

[18] R. Chai, A. Savvaris, A. Tsourdos, S. Chai, and Y. Xia, "A review of optimization techniques in spacecraft flight trajectory design," *Prog. Aerosp. Sci.*, vol. 109, Aug. 2019, Art. no. 100543.

[19] C. Dong and Y. Cai, "Reentry trajectory optimization for hypersonic glide vehicle with flexible initial conditions," *J. Aerosp. Eng.*, vol. 30, no. 5, Sep. 2017, Art. no. 04017033.

[20] B. Li, X. Qi, B. Yu, and L. Liu, "Trajectory planning for UAV based on improved ACO algorithm," *IEEE Access*, vol. 8, pp. 2995–3006, 2020.

[21] X. Xia, Y. Xing, B. Wei, Y. Zhang, X. Li, X. Deng, and L. Gui, "A fitness-based multi-role particle swarm optimization," *Swarm Evol. Comput.*, vol. 44, pp. 349–364, Feb. 2019.

[22] Y. Zhou, Y. Su, A, Xie A, and L. Kong, "A newly bio-inspired path planning algorithm for autonomous obstacle avoidance of UAV," *Chin. J. Aeronaut.*, to be published, doi: 10.1016/j.cja.2020.12.018.

[23] Y. Zhuang, H. Huang, S. Sharma, D. Xu, and Q. Zhang, "Cooperative path planning of multiple autonomous underwater vehicles operating in dynamic ocean environment," *ISA Trans.*, vol. 94, pp. 174–186, Nov. 2019.

[24] P. Yang, K. Tang, J. A. Lozano, and X. Cao, "Path planning for single unmanned aerial vehicle by separately evolving waypoints," *IEEE Trans. Robot.*, vol. 31, no. 5, pp. 1130–1146, Oct. 2015.

[25] C. Huang and J. Fei, "UAV path planning based on particle swarm optimization with global best path competition," *Int. J. Pattern Recognit. Artif. Intell.*, vol. 32, no. 06, Jun. 2018, Art. no. 1859008.

[26] Y. Liu, X. Zhang, X. Guan, and D. Delahaye, "Adaptive sensitivity decision based path planning algorithm for unmanned aerial vehicle with improved particle swarm optimization," *Aerosp. Sci. Technol.*, vol. 58, pp. 92–102, Nov. 2016.

[27] D. Tian and Z. Shi, "MPSO: Modified particle swarm optimization and its applications," *Swarm Evol. Comput.*, vol. 41, pp. 49–68, Aug. 2018.

[28] D. Tian, "Particle swarm optimization with chaos-based initialization for numerical optimization," *Intell. Autom. Soft Comput.*, vol. 24, no. 2, pp. 331–342, Jun. 2018, doi: 10.1080/10798587.2017.1293881.

[29] H. Modares and M. B. N. Sistani, "Solving nonlinear optimal control problems using a hybrid IPSO–SQP algorithm," *Eng. Appl. Artif. Intell.*, vol. 24, no. 3, pp. 476–484, 2011.

[30] Y. Zhuang and H. Huang, "Time-optimal trajectory planning for underactuated spacecraft using a hybrid particle swarm optimization algorithm," *Acta Astronautica*, vol. 94, no. 2, pp. 690–698, Feb. 2014.

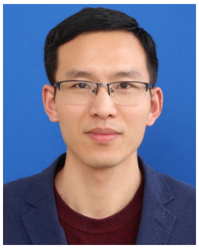
[31] Y. Zhuang, S. Sharma, B. Subudhi, H. Huang, and J. Wan, "Efficient collision-free path planning for autonomous underwater vehicles in dynamic environments with a hybrid optimization algorithm," *Ocean Eng.*, vol. 127, pp. 190–199, Nov. 2016.

[32] W. Wang, Q. Tao, Y. Cao, X. Wang, and X. Zhang, "Robot time-optimal trajectory planning based on improved cuckoo search algorithm," *IEEE Access*, vol. 8, pp. 86923–86933, May 2020.

[33] J. Li, M. Ran, and L. Xie, "Efficient trajectory planning for multiple non-holonomic mobile robots via prioritized trajectory optimization," *IEEE Robot. Autom. Lett.*, vol. 6, no. 2, pp. 405–412, Apr. 2021.

[34] C. Hua, J. Chen, F. Wang, and X. Guan, "Distributed adaptive containment control of uncertain QUAV multiagents with time-varying payloads and multiple variable constraints," *ISA Trans.*, vol. 90, pp. 107–115, Jul. 2019.

[35] O. Mofid and S. Mobayen, "Adaptive sliding mode control for finite-time stability of quad-rotor UAVs with parametric uncertainties," *ISA Trans.*, vol. 72, pp. 1–14, Jan. 2018.



SHIKAI SHAO received the Ph.D. degree from the School of Electrical Automation and Information Engineering, Tianjin University, Tianjin, China, in 2017. He is currently a Lecture with the School of Electrical Engineering, Hebei University of Science and Technology, Shijiazhuang, China. He is also a part-time Engineer with the State Key Laboratory of Satellite Navigation System and Equipment, Shijiazhuang. His research interests include formation optimization and coordinated control.

CHENGLONG HE received the Ph.D. degree from the Institute of Communication Measurement and Control Technology, Shijiazhuang, China, in 2017. He is currently a Senior Engineer with the State Key Laboratory of Satellite Navigation System and Equipment, Shijiazhuang. His research interests include satellite navigation and networks optimization.

YUANJIE ZHAO received the M.S. degree from the School of Computer and Communication Engineering, Tianjin University of Technology, Tianjin, China, in 2013. She is currently an Engineer with the School of Electrical Engineering, Hebei University of Science and Technology, Shijiazhuang. Her research interests include networks optimization and path planning.

XIAOJING WU received the M.S. and Ph.D. degrees from Yanshan University, in 2009 and 2012, respectively. She is currently an Associate Professor with the Hebei University of Science and Technology, China. Her research interests include nonlinear control, UAV control, and path planning.