Neurocomputing 171 (2016) 878–888

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Modified central force optimization (MCFO) algorithm for 3D UAV path planning

Yongbo Chen [a], Jianqiao Yu [a], Yuesong Mei [a,*], Yafei Wang [b], Xiaolong Su [c]

[a] School of Aerospace Engineering, Beijing Institute of Technology, Beijing 100081, China
[b] Jiangsu Automation Research Institute, Lianyungang, Jiangsu 222061, China
[c] The 41st Institute of Fourth Academy of Aerospace Science and Technology Corporation, Xian, Shaanxi 710025, China

ARTICLE INFO

Article history:
Received 16 March 2015
Received in revised form
5 July 2015
Accepted 11 July 2015
Available online 26 July 2015

Keywords:
Unmanned aerial vehicle (UAV) path planning
Modified central force optimization (MCFO) method
Convergence analysis
Linear difference equation method

ABSTRACT

Path planning for the three-dimensional (3D) unmanned aerial vehicles (UAV) is a very important element of the whole UAV autonomous control system. In this paper, a modified central force optimization (MCFO) method is introduced to solve this complicated path-optimization problem for the rotary wing vertical take-off and landing (VTOL) aircraft. In the path planning process, the idea from the particle swarm optimization (PSO) algorithm and the mutation operator of the genetic algorithm (GA) are applied to improve the original CFO method. Furthermore, the convergence analysis of the whole MCFO method is established by the linear difference equation method. Then, in order to verify the effectiveness and practicality of this new path planning method, the path following process is put forward based on the six-degree-of-freedom quadrotor helicopter control system. At last, the comparison simulations among the six algorithms show that the trajectories produced by the whole MCFO method are more superior than the original CFO algorithm, the GA, the Firefly algorithm (FA), the PSO algorithm, the random search (RS) way and the other MCFO algorithm under the same conditions. What is more, the path following process results show that the path planning results are practical for the real dynamic model of the quadrotor helicopter.

1. Introduction

Unmanned aerial vehicles (UAVs) have demonstrated their usefulness in both military and civilian applications [1]. The path planning problem is a key element of the unmanned aerial vehicle (UAV) autonomous control system. It is a complicated global optimization problem which allows the UAV to autonomously compute the optimal or near-optimal path from the start point to the target based on the requirements and restraints of the task. With the extensive use of various types of UAVs, the topic has attained high popularity in recent years.

A series of methods have been raised to deal with this complicated optimization problem, such as artificial potential field (APF) method [2], A* algorithm [3], genetic algorithm (GA) [4], artificial intelligence algorithms [5–6] and so on. Among all the methods, computational intelligence methods, which have been developing rapidly in recent years, are widely used.

As an example, the authors of [4] used the GA to solve the multi-constraints 3D UAV path planning problem. The authors of [6] compared the parallel GA with the PSO algorithm in the real-time 3D path planning problem. The authors in [7] developed an any-time algorithm using particle swarm optimization (PSO) to solve the multiple UAVs path planning problem. The authors of [8] presented two types of path planners based on evolutionary algorithm in the on-line and off-line situation. Besides these common algorithms, some new computational intelligence methods were presented. An improved Gravitational Search Algorithm (GSA) was proposed to solve the path planning problem by the authors of [9]. Although most computational intelligence methods have a powerful global optimization ability, the efficiencies of different methods are uneven for the same problem under the similar parameters. So the efficiency is a very important index to measure whether an algorithm is good or bad.

Central Force Optimization (CFO) was originally presented as a new optimization algorithm by Formato in 2007, which is based on the metaphor of gravitational kinematics [10]. It can solve some test functions and real-world problems effectively. A series of researches on this algorithm are presented in some Refs. [11–13], but the study and application of this algorithm is still in an original state.

* Corresponding author.
E-mail addresses: bit_chenyongbo@163.com (Y. Chen), jianqiao@bit.edu.cn (J. Yu), mys001@bit.edu.cn (Y. Mei), wangyafei@bit.edu.cn (Y. Wang), suxiaolong@bit.edu.cn (X. Su).

http://dx.doi.org/10.1016/j.neucom.2015.07.044
0925-2312/

However, in the field of path planning for UAV, there has been no application of CFO algorithm up to now. At the same time, the convergence analysis of the CFO algorithm is scarce in actual researches. In this paper, a modified CFO algorithm is used to solve the UAV path planning problem. Here, the idea of the PSO and the variation in information of GA are introduced into the CFO to get a superior performance of CFO, and then an improved CFO algorithm is used to search the optimal or suboptimal path with complicated constraints. After the presentation of our MCFO algorithm, some discussions about the convergence analysis of the MCFO algorithm is finished. In order to verify the effectiveness of our proposed approach, it is compared with the CFO and other methods, such as GA, FA, PSO and random search method (RS) under the same environments. Then, the path following process is introduced to prove the practicality of the planning result.

The general framework of this paper is as follows: first, the obstacles' representations are statically defined. Second, the cost function and penalty function are introduced. Then, the optimal path is computed off-line and the flight speed scheme is predefined. Finally, the UAV can move in the space according to the computed path.

The structure of this paper is as follows: in Section 2, the environment description models of the UAV path planning problem are presented. The cost function of the path is presented. At the same time, the obstacle constraint condition of the path is expressed as the penalty function. In Section 3, the original CFO algorithm is introduced. In Section 4, the improved CFO algorithm with the mutation operations and PSO operations for UAV is presented. In Section 5, the convergence analysis of the modified CFO algorithm is presented based on the linear difference equation method. In Section 6, the path smoothing process is finished by the space circular arcs. In Section 7, the path following process is completed by the six degrees of freedom simulation model of the quadrotor helicopters. The simulation model can verify the usability of the path planning result. In Section 8, the simulations are accomplished which include the results of the improved CFO algorithm and some comparative experiments with the original CFO, the other MCFO algorithm, the basic GA, the FA, the PSO algorithm and the random search (RS) way. Finally, further discussions and conclusions are accomplished in Section 9.

2. Problem formulation

In the complex combat terrain, the UAV flies from the start point to the target point according to the mission planning result. During this flying process, the UAV may face the orographic barriers, the fire threats, the radar scan areas and some other obstacles. The primary mission of the path planning algorithm is to obtain a network diagram connecting the starting point with the target point by meshing the mission area. So the essence of the UAV path planning problem is a path-optimization problem. We assume that the UAV maintains a given flight speed scheme which is designed in advance. In this way, the path planning problem is simplified into a static broken line planning problem. In our work, we use the mathematical model in the UAV path planning problem, which is described as follows.

2.1. Environmental modeling

Modeling of the obstacles is a key part of the UAV path planning. The way used to describe the obstacles has an effect on the representation of the trajectory path and search algorithm. The reasonable modeling way can simplify the problem. In the small-scale environment, especially in the urban environment, the planning spaces are always not flat. It is unreasonable to simplify the planning space into the 2-dimensional (2D) scene. In the planning space, we define the starting point and the target point respectively as $\boldsymbol{S}$ and $\boldsymbol{T}$. In the process of the UAV moving from the starting point to the target point, it may meet many typical obstacles, such as: buildings, mountains, other vehicles, radar area and so on. They can be presented in the form of cubes, floating balls and other simple models. The slenderness ratio of the obstacle is the main element to decide the species of the simple model. If the slenderness ratio of the obstacle is big, the cuboid will be used to describe it. Otherwise, the sphere will be used. The parameterization of the obstacle set is established by the geometric centers of the real obstacles. In the real environment, the accurate coordinates of the geometric centers are very difficult to obtain usually. So in order to remove the influence of the measuring error, the simple models need to be magnified extra to ensure that the models can completely encase the real obstacles. The path planning algorithm is to generate an optimal or sub-optimal path between $\boldsymbol{S}$ and $\boldsymbol{T}$ considering all these obstacles (Fig. 1).

In the planning space, as long as the coordinates of $\boldsymbol{S}$ and $\boldsymbol{T}$ are limited to non-negative, a right-handed Cartesian coordinate is established optionally. According to the x-axis coordinates of $\boldsymbol{S}$ and $\boldsymbol{T}$, we can design the x-axis range of the path points. If the coordinates of $\boldsymbol{S}$ and $\boldsymbol{T}$ are (x_1, y_1, z_1) and (x_n, y_n, z_n), we divide x-axis range (x_1, x_n) into $n-1$ equal portions and defined as $x_1, x_2, x_3,\ldots, x_n$. The perpendicular planes $(P_1, P_2, P_3,\ldots, P_n)$ of the x-axis are established according to the corresponding split points. Put a discrete point in each vertical plane P_i $(i_1=2,\ldots, n-1)$, engendering a collection of discrete points $C=\{\boldsymbol{S}, (x_2, y_2, z_2), (x_3, y_3, z_3),\ldots, (x_{n-1}, y_{n-1}, z_{n-1}), \boldsymbol{T}\}$. These waypoints are connected by the straight lines. Then we can get a whole flight path which is a stripe structure. In this way, the path planning problem is turned into optimizing the coordinates series to obtain a better cost function value of the objective function. In order to simplify the solution to the optimization problem, the optimization variable $\boldsymbol{L}$ is defined as: $\boldsymbol{L}=[y_2, y_3,\ldots, y_{n-1}, z_2, z_3,\ldots, z_{n-1}]$. Then the path planning problem turns into a $2n-4$ dimensional optimization problem. It is known that higher dimension will lead to higher complexity and higher precision. So the value of n is very important to the solution to this problem. Because the number n of the portions determines the dimensions of the mission environment, in order to control it and improve the optimization efficiency, n can be limited in [5,15].

Remark: because the model of the path is a series of broken lines, this method can only be applied for the rotary wing vertical take-off and landing (VTOL) aircraft.

2.2. Cost function

It is known that searching for the best path is often associated with searching for the shortest path. But in the real battlefield environment the UAV needs to adapt to the constantly changing conditions. So in the case of the UAV path planning, the optimal path

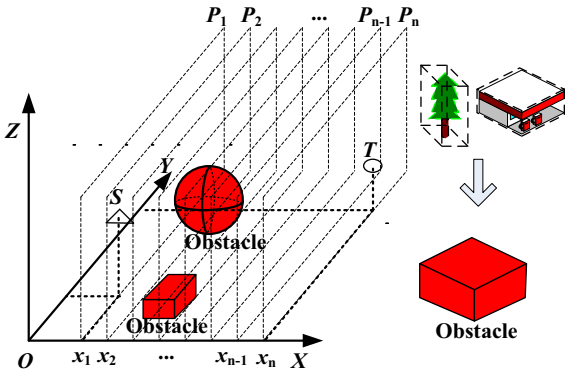

Fig. 1. Modeling of the planning space

is more complex and includes many different characteristics. If these characteristics are taken into consideration, the more complicated cost function of the UAV path planning is used in this paper. The multi-objective cost function in our paper is as follows [14]:

$$J_{\text{cost}} = w_1 \cdot J_{\text{path}} + w_2 \cdot J_{\text{height}} + w_3 \cdot J_{\text{smooth}}, \tag{1}$$

where J_{cost} denotes the total cost function; J_{path} means the cost function of the length of the path; J_{height} is the cost function of the standard deviation of the height; J_{smooth} is the cost function of the smoothness of the planning path; w_i, i_2=1,2,3 denote the weights of each function and they satisfy the conditions:

$$\begin{cases} w_{i_1} \geq 0 \\ \sum_{i_1=1}^{3} w_{i_1} = 1 \end{cases}. \tag{2}$$

In the UAV path planning process, the length of the planning path is also very important for most path planning missions. Obviously, the shorter path can save more fuel, save more time and give more security. Then, in general, J_{path} is defined as follows:

$$J_{\text{path}} = \sum_{k=1}^{n-1} \left\| (x_{k+1}, y_{k+1}, z_{k+1}) - (x_k, y_k, z_k) \right\|_2, \tag{3}$$

where (x_k, y_k, z_k) means the k-th waypoint in a whole planning path.

$$path_{i_1,\lambda}(\boldsymbol{L}) = \begin{cases} \lambda \boldsymbol{S} + (1-\lambda)[x_{i_1+1}, \boldsymbol{L}(i_1), \boldsymbol{L}(i_1+n-2)], (i_1 = 1) \\ \lambda[x_{i1}, \boldsymbol{L}(i_1-1), \boldsymbol{L}(i_1+n-3)] + (1-\lambda)[x_{i_1+1}, \boldsymbol{L}(i), \boldsymbol{L}(i_1+n-2)], (i_1 = 2,3,\ldots,n-2) \\ \lambda[x_{i-1}, \boldsymbol{L}(i_1-1), \boldsymbol{L}(i_1+n-3)] + (1-\lambda)\boldsymbol{T}, (i_1 = n-1) \end{cases} \quad \lambda \in [0,1], \tag{9}$$

At the same time, the stable flying height of the UAV is also important for the UAV path planning process. For most air vehicles, the flying height should not change too drastically. The stable flying height can help to reduce the burden of the control system and save more fuel. Then, J_{height} is defined as follows:

$$J_{\text{height}} = \sqrt{\frac{1}{n} \sum_{k=1}^{n} (z_k - \overline{z})^2}, \tag{4}$$

where

$$\overline{z} = \frac{1}{n} \sum_{k=1}^{n} z_k, \tag{5}$$

In general, the fuel cost of the UAV flying process is smallest when the UAV moves in uniform rectilinear motion. Considering the flight efficiency of the UAV, the work state of the UAV is inefficient when the UAV does the turning maneuvers. Then, J_{smooth} is defined as follows:

$$J_{\text{smooth}} = (x_n - x_1) \sum_{k=1}^{n-2} \arccos\left(\frac{\varphi_{k+1} \varphi_k}{|\varphi_{k+1}||\varphi_k|} \right), \tag{6}$$

where φ_k means $(x_{k+1}-x_k, y_{k+1}-y_k, z_{k+1}-z_k)$.

The cost function J_{cost} represents a specific scenario when the optimal path minimizes the distance traveled, the most stable flying height and the smoothest flying process, while respecting the UAV performance characteristic. The multi-objective cost function is comprehensive and complex for most UAV path planning process.

2.3. Optimization model formulation

In the area of the optimization, many problems are constrained by some constraint conditions [1]. Generally speaking, this kind of optimization problems can be defined as follows:

$$\min_{\boldsymbol{x}} f(\boldsymbol{x}'), \boldsymbol{x}' = [x'_1, x'_2, \ldots, x'_m]$$
$$s.t. \quad \begin{array}{l} h_{j_1}(\boldsymbol{x}') \leq 0, j_1 = 1,2,\ldots,p, \\ g_{j_2}(\boldsymbol{x}') = 0, j_2 = 1,2,\ldots,q \end{array} \tag{7}$$

where $\boldsymbol{x}'$ means the independent variable, $x'_{i'} \in [x_{i'\min}, x_{i'\max}]$, i'=1, 2,…, m; $f(\bullet)$ means the cost function of the optimization problem; $h_{j1}(\bullet)$ are the inequality constraint functions, p is the number of the inequality constraints; $g_{j2}(\bullet)$ are the equality constraint functions, q is the number of the equality constraints.

According to section II.A, II.B and Eq. 7, the optimization model of the path planning problem can be defined as follows:

$$\min_{\boldsymbol{L}} J_{\text{cost}}(L)$$
$$s.t. \quad path_{i_1,\lambda}(L) \notin Obstacle \cup Ground, \tag{8}$$

where *Obstacle* and *Ground* represent the obstacle set and the ground set in the planning space, respectively; $path_{i1,\lambda}(\bullet)$ means the path points in the broken line path between the i_1-th and i_1+1-th planes. We can obtain the coordinates of any path points based on the values of λ and i

After we get the standard form of the path planning optimization problem, a small problem about how to implement the constraints appears. In order to solve this problem, several path points are chosen uniformly in the path. The specific selection way is to change the value of λ at a fix interval. Of course, the smaller interval can judge whether the path through the obstacle and the ground or not more accurately. But it will cost more time. In this paper, our interval is 0.01.

It is known that the constrained optimization problem is more difficult to solve than the unconstrained optimization problem. In order to simplify the problem, the penalty function method is applied to simplify the constraints [15]. In this way, the constrained optimization problem turns into the unconstrained optimization problem.

If L satisfies:

$$s.t. \quad path(\boldsymbol{L}) \notin Obstacle \cup Ground, \tag{10}$$

In this case, $\boldsymbol{L}$ is called feasible solution.

For the unfeasible solution, the cost function needs to be redefined as:

$$\min_{\boldsymbol{L}} J'_{\text{cost}}(\boldsymbol{L}) = c_J J_{\text{cost}}(\boldsymbol{L}), \tag{11}$$

where c_J is the constant which is much bigger than 1.

Finally, the UAV path planning problem can be established as the unconstrained optimization model whose variable is the coordinates of the waypoints in $n-2$ corresponding planes. So the model can be rewritten as:

$$\min_{\boldsymbol{L}} J'_{\text{cost}}(\boldsymbol{L}) = \begin{cases} J_{\text{cost}}(\boldsymbol{L}) & path(\boldsymbol{L}) \notin Obstacle \cup Ground \\ c_J J_{\text{cost}}(\boldsymbol{L}) & else \end{cases}. \tag{12}$$

3. Central force optimization (CFO)

Because the dimensions of the path planning optimization problems are high, the traditional methods are hard to apply. The central force optimization (CFO) method is used to solve this problem. The CFO algorithm is a new intelligence particle optimization method, in which the search algorithm is inspired by the law of gravity. Every solution means a particle. It models the probes that fly though the search space under the influence of gravity [16]. Each particle attracts every other particle with the virtual gravity force, as is shown in Fig. 2. The virtual gravity force is caused by the mass of each particle which is dependent on the cost function of the solution.

In a decision space defined by $p^{\min}(k) \le p(k) \le p^{\max}(k)$, $k=1,\ldots, N$, the $p(k)$ being decision variables $\boldsymbol{p}=[p(1), p(2),\ldots, p(N)]$, determine the locations and values of the objective function $f(\boldsymbol{p})$, where N is the dimension of objective function; $p^{\min}(k)$, $p^{\max}(k)$are the bound of the k-th dimension variables. The basic procedures of the original CFO algorithm are as follows: (a) initialization, (b) acceleration calculation, (c) Motion [17].

In the initialization process, a pollution N_p of particles is created randomly in N-dimension space:

$$\boldsymbol{p}_i^j = [p_i^j(1), \ldots, p_i^j(d), \ldots, p_i^j(N)], i = 1, 2, \ldots, N_p, \tag{13}$$

where $p_i^j(d)$ is the d-th dimension position of the i-th particle in the j-th iteration.

Then, the acceleration of the i-th particle in the j-th iteration is defined as follows:

$$\boldsymbol{a}_i^j = G \sum_{\substack{k=1 \\ k \ne j}}^{N_p} U(f(\boldsymbol{p}_k^j) - f(\boldsymbol{p}_i^j)) \cdot (f(\boldsymbol{p}_k^j) - f(\boldsymbol{p}_i^j))^{\alpha} \cdot \frac{\boldsymbol{p}_k^j - \boldsymbol{p}_i^j}{\|\boldsymbol{p}_k^j - \boldsymbol{p}_i^j\|^{\beta}}, i = 1, 2, \ldots, N_p, \tag{14}$$

where G is gravitation constant; $U(\bullet)$ is the judgement function, which meets $U(\bullet) = \begin{cases} 1 & \bullet \ge 0 \\ 0 & \bullet < 0 \end{cases}$; α and β are the tuning parameters and they satisfy $\alpha > 0$, $\beta > 0$; $f(\boldsymbol{p}_i^j)$ means the cost function of the i-th particle in the j-th iteration.

At last, the particles move according to the Newton's law of motion:

$$\boldsymbol{p}_i^{j+1} = \boldsymbol{p}_i^j + \boldsymbol{v}_i^j \Delta t + \frac{1}{2}\boldsymbol{a}_i^j \Delta t^2, i = 1, 2, \ldots, N_p, \tag{15}$$

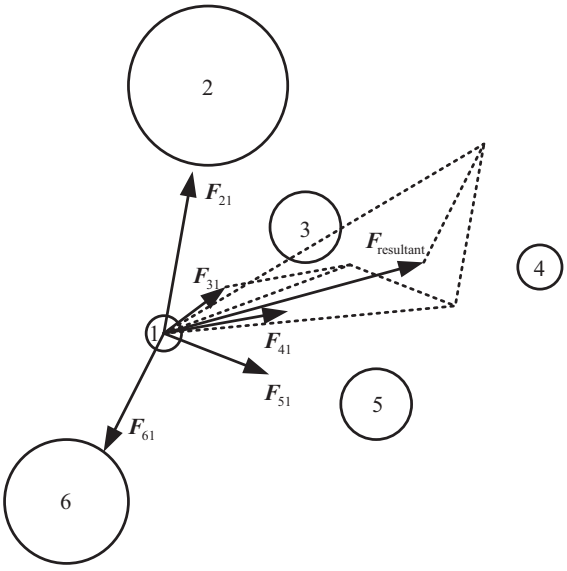

Fig. 2. The resultant force of the gravity.

where $\boldsymbol{p}_i^{j+1}$ is the position of the i-th particle in the $j+1$-th iteration; $\boldsymbol{v}_i^j$ is the velocity vector of the i-th particle in the j-th iteration. In the original CFO method, $\boldsymbol{v}_i^j$ is ignored ($\boldsymbol{v}_i^j=[0, 0, \ldots,0]_{1\times N}$); and Δt means the time constant.

If $p_i^j(d)$ is bounded, the supplementary information needs to be added, as follows:

$$p_i^j(d) = \begin{cases} p_i^{\min}(d) + h(p_i^{j-1}(d) - p_i^{\min}(d)) & \text{if} \quad p_i^j(d) < p_i^{\min}(d) \\ p_i^{\max}(d) - h(p_i^{\max}(d) - p_i^{j-1}(d)) & \text{else if } p_i^j(d) > p_i^{\max}(d) \\ p_i^j(d) & \text{else} \end{cases}, \tag{16}$$

where h means the accommodation coefficient.

The whole pseudo code of the original CFO algorithm is shown by Fig. 3.

4. Modified central force optimization (MCFO)

4.1. Defects of the original CFO algorithm

The original CFO algorithm has many merits, such as the high rate of convergence, the simple forms of mathematics, the ease implementation, the certainty of the parameter and so on. Its effect has been tested by many test functions in Refs. [10–12]. But the rate of convergence of the CFO which is based on the parameters G, α and β is both its greatest strength and its central weakness. In the beginning stages, the convergence speed of the original CFO algorithm is high enough for most problems. However, because the convergence speed largely depends upon the differences between the particles, when the differences between the particles decrease (i. e. $|(f(\boldsymbol{p}_i^j) - f(\boldsymbol{p}_k^j))|^{\alpha}$ converges to zero) the convergence speed will greatly slow down. So we need to introduce some new idea from the other algorithms to improve the convergence speed of the end of the optimization process. In

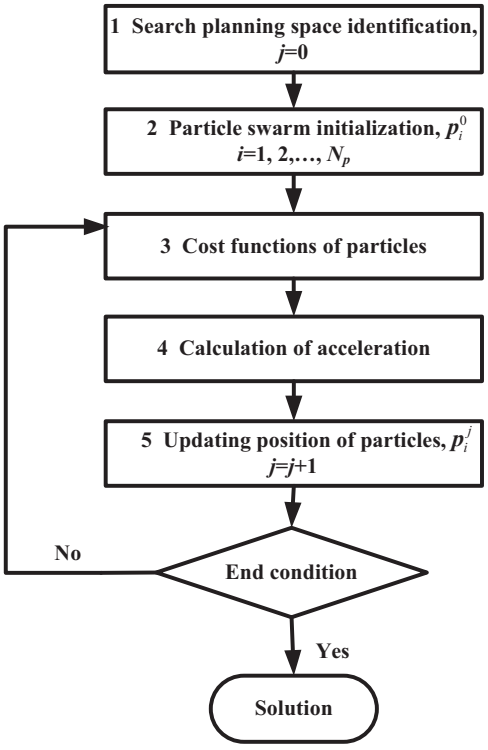

Fig. 3. The flow chart of the original CFO algorithm.

order to solve this main problem, the ideas from the PSO and the mutation are introduced.

4.2. Idea from PSO

The particle swarm optimization (PSO) algorithm, proposed by Kennedy and Eberhart [18], is an efficient evolutionary computation method. It is motivated from the simulation of the social behavior (flock of birds) [16]. Similar to the CFO method, the PSO method is also to update the pollution of particles by using an operator according to the cost function values obtained from the planning space. However the movement strategies are different. Advantages of the PSO method are easy implementation, simple structure, speed, and robustness.

In the PSO algorithm, the searching strategy of the particles is calculated as follows [16,18]:

$$p_i^{j+1}(d) = p_i^j(d) + v_i^{j+1}(d)$$
$$v_i^{j+1}(d) = w^j v_i^j(d) + c_1 r_{1i}^j (pbest_i(d) - p_i^j(d)) + c_2 r_{2i}^j (gbest(d) - p_i^j(d)), \quad (17)$$

where r_{1i}^j and r_{2i}^j are two random variables in the range [0, 1]; c_1 and c_2 are positive constants; w^j is the inertia weight; $pbest_i = [pbest_i(1),\ldots, pbest_i(d),\ldots, pbest_i(N)]$ and $gbest = [gbest(1),\ldots, gbest(d),\ldots, gbest(N)]$ are the best position of the i-th particle and the best particle among all existing particles, respectively.

In the searching procedure of the CFO method, the CFO is memory-less and only the current positions of the particles play roles in the updating procedure [9]. However, the PSO algorithm uses a kind of memory for updating the velocity (due to $pbest_i$ and $gbest$). The original CFO method is improved by the operator in the PSO algorithm. In this way, the modified CFO method has the memory ability. The updating equation can be defined as:

$$\tilde{p}_i^j(d) = p_i^j(d) + c_1 r_{i1}^j (pbest_i(d) - p_i^j(d)) + c_2 r_{i2}^j (gbest(d) - p_i^j(d)), \quad (18)$$

where $\tilde{p}_i^j(d)$ means the new position of the d-th dimension position of the i-th particle in the j-th iteration (after the PSO operator).

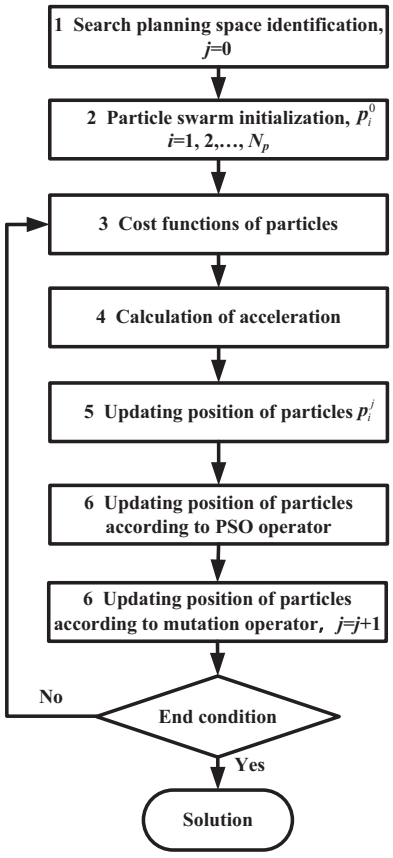

Fig. 4. General principle of MCFO.

4.3. Idea of mutation

Genetic algorithm that imitates the survival of the fittest is a stochastic and heuristic search method based on the mechanics of natural selection and genetics [19]. According to the biological principle of evolution, one of the reasons of many living thing species extermination is the lack of the fine variation. So in order to avoid the local minimum problem, the mutation operator is an important basic operator in the GA.

The original CFO method makes full use of the individual powers, especially for the elites. The elites whose cost functions are better than the average level will cause larger gravity to lead other particles. But the changes of the elites are less. This phenomenon will result in the local minimum problem. So the mutation operator of the GA needs to be introduced into the CFO method to improve the exploration of the new search space.

The main operational approach is as follows: first, a judge variable ζ is introduced to judge whether the mutation is effective or not for p_i^j. Second, the range of the mutation value is limited by a decreasing expression. Then, a random variable ξ impacts on the decreasing expression. At last, the whole mutation operator is applied to update the position of the particle. The mutation operator is defined as:

$$m_i^j(d) = \begin{cases} \xi e^{-cj} & \zeta \le c_3 \\ 0 & \zeta > c_3 \end{cases}$$
$$\widehat{p}_i^j(d) = \begin{cases} \tilde{p}_i^j(d) + m_i^j(d) & if \quad f(\widehat{p}_i^j(d)) > f(\tilde{p}_i^j(d)) \\ \tilde{p}_i^j(d) & else \end{cases}, \quad (19)$$

where c and c_3 are the given constant coefficients, and c meets $c > 0$; $\widehat{p}_i^j(d)$ means the new position of the d-th dimension position of the i-th particle in the j-th iteration (after the mutation operator); ξ is in the range $[-0.5, 0.5]$.

In this way, we combined mutation operation with CFO method which modifies the solutions with poor fitness in order to add diversity of the population to improve the search efficiency.

4.4. Whole MCFO algorithm for UAV path planning

The MCFO method combines the original CFO algorithm, the PSO operator and the mutation operator. The whole pseudo code of the modified CFO algorithm is shown by Fig. 4.

The detail frame of the modified central force optimization algorithm for the UAV path planning problem is shown as below

Step 1: The modeling of the planning space is established according to Section 2.1. Then the 3D UAV path planning problem turns into the optimization problem by the help of the optimization variable $\boldsymbol{L}$ and the cost function.

Step 2: Initialization. Set the generation counter $j=1$; Initialize the population of initial particles population randomly; Define all parameters, such as: the time constant Δt, the given constant coefficients c and c_1 and so on.

Step 3: Calculate the multi-objective cost function of each path formed by relative parameters based on Section 2.2. Because the MCFO method is more suitable for the optimization problem to find the maximum function, the cost function J_{cost} needs to be redefined as $M - J'_{cost}$, where M is bigger than all

cost functions that might appear. Then, choose the best path in those paths, and define it as *gbest*. At the same time, define the initial particles population as $pbest_i(i=1, 2,\ldots, N_p)$.

Step 4: While the end condition is not satisfied or $j<$ MaxGeneration, do the following things: (a) Calculate the acceleration of the particles in different directions based on Eq. (14), then, update the position of the particles according to Eq. (15); (b) with the help of *gbest* and $pbest_i(i=1, 2,\ldots, N_p)$, update the position of the particles according to Eq. (18). The idea from the PSO algorithm is introduced into the MCFO. Then, create a random number ζ. Based on the judge variable and cost functions, the mutation operator is introduced to improve the diversity of the population. The positions of the particles move depending on Eq. (19). Choose the best path in those paths, and define it as *gbest*. At the same time, define the best previous position of the i-th particle as $pbest_i(i=1, 2,\ldots, N_p)$. Order j as $j+1$, and repeat this step.

Step 5: Post-process, output and visualize the results.

5. Convergence analysis

The convergence analysis of MCFO is developed here. Some convergence conditions are put forward in this Section. All particles using the CFO method converge to the optimum solutions, however, nothing will be said about whether these solutions represent local or global optimums. Without loss of generality, the MCFO method is limited in one-dimensional situation ($N=1$) and the selection of the particle i is arbitrary. During the updated process, assume that, for all iterations, p_i^j meets:

$$f(\widetilde{p}_i^j)>f(\tilde{p}_i^j)\text{and}\zeta^j\le c_3, \tag{20}$$

Definition 5.1. (Linear Difference Equations) Let $\boldsymbol{J}^+$ is a set of the nonnegative integer, square nonsingular matrix $A(\bullet)$: $\boldsymbol{J}^+\to R^{n\times n}$, $f^*(\bullet)$: $\boldsymbol{J}^+\to R^n$, $\boldsymbol{x}^*(j)\in R^n$, for each $j\in\boldsymbol{J}^+$, the linear difference equation and corresponding homogeneous linear equations with variable coefficients are as follows:

$$\boldsymbol{x}^*(j+1)=A(j)\boldsymbol{x}^*(j)+\boldsymbol{f}^*(j), \tag{21}$$

$$\boldsymbol{x}^*(j+1)=A(j)\boldsymbol{x}^*(j). \tag{22}$$

Theorem 5.1. *The movement equation of the particle p_i^j is a first-order linear difference equation with variable coefficients.*

Proof. Define $\boldsymbol{M}_i=\{k|\ f(p_k^j)-f(p_i^j)>0,\ k=1, 2,\ldots, N_p\}$. The whole update of the MCFO method is rewritten as:

$$\begin{aligned}p_i^{j+1}&=p_i^j+a_i^j,\\ a_i^j&=\frac{1}{2}G\sum_{k\in\boldsymbol{M}_i}\frac{(f(p_k^j)-f(p_i^j))^\alpha}{\left|p_k^j-p_i^j\right|^\beta}(p_k^j-p_i^j)\Delta t^2+c_1r_{1i}^j(pbest_i-p_i^j)\\ &\quad+c_2r_{2i}^j(gbest-p_i^j)+\xi^j e^{-cj}.\end{aligned} \tag{23}$$

Define

$$\begin{aligned}\phi_i^j&=G\sum_{k\in\boldsymbol{M}_i}\frac{(f(p_k^j)-f(p_i^j))^\alpha}{\left|p_k^j-p_i^j\right|^\beta}p_k^j,\\ \theta_i^j&=G\sum_{k\in\boldsymbol{M}_i}\frac{(f(p_k^j)-f(p_i^j))^\alpha}{\left|p_k^j-p_i^j\right|^\beta}.\end{aligned} \tag{24}$$

Eq. (23) is rewritten as

$$\begin{aligned}p_i^{j+1}&=p_i^j+\frac{1}{2}\Delta t^2\phi_i^j-\frac{1}{2}\Delta t^2\theta_i^jp_i^j+c_1r_{1i}^j(pbest_i-p_i^j)\\ &\quad+c_2r_{2i}^j(gbest-p_i^j)+\xi^j e^{-cj}\\ &=\left(1-\frac{1}{2}\Delta t^2\theta_i^j-c_1r_{1i}^j-c_2r_{2i}^j\right)p_i^j\\ &\quad+\left(\frac{1}{2}\Delta t^2\phi_i^j+c_1r_{1i}^jpbest_i+c_2r_{2i}^jgbest+\xi^j e^{-cj}\right).\end{aligned} \tag{25}$$

Define

$$\begin{aligned}A_i(j)&=1-\frac{1}{2}\Delta t^2\theta_i^j-c_1r_{1i}^j-c_2r_{2i}^j\\ f_i^*(j)&=\frac{1}{2}\Delta t^2\phi_i^j+c_1r_{1i}^jpbest_i+c_2r_{2i}^jgbest+\xi^j e^{-cj}.\end{aligned} \tag{26}$$

Go on with a variable substitution, Eq. (25) is formed into

$$p_i^{j+1}=A_i(j)p_i^j+f_i^*(j), \tag{27}$$

where $A_i(j)\in R$, $f_i^*(j)$ is bounded.

So theorem 5.1 is proved.

After obtaining the differential form of the intelligent algorithm (like: Eq. (27)), there are many references applying the linear system theory which is based on the characteristic equations of the differential equations to solve the convergence analysis problems [20–23]. Based on the linear system theory, the restriction condition to guarantee the system stable is that the spectral radius of the system matrix $A(j)$ stays in the unit circle. It is the necessary and sufficient condition of the asymptotic stability for the discrete linear time-invariant system. But this conclusion is not suitable for the discrete linear time-variant system [24]. The authors in Ref. [25] put forward two counter-examples respectively which show two situations. The first one shows that the movement is also unstable when the spectral radius of the system matrix $A(j)$ stays in the unit circle for the discrete linear time-variant system. Then the second counter-example shows that the movement is still stable when the spectral radius of the system matrix $A(j)$ stays outside the unit circle for the discrete linear time-variant system. Obviously, the differential forms of the intelligent algorithms are all time-variant. So the conclusions of the above references are circumscribed by the assumption of the fixity coefficient. The new conclusion needs to be put forward.

Theorem 5.2. *$\forall f^*(j)\in[\boldsymbol{J}^+, R^n]$, for the linear difference equations, the stability degree of the linear difference equation is equivalent to the stability degree of the general solution of the corresponding homogeneous linear equations [26].*

Theorem 5.3. *If there is a kind of matrix norm that satisfy*

$$\|A(j)\|_x\le\frac{j+1}{j+2}, j=j_0, j_0+1,\ldots, \tag{28}$$

where $\|\bullet\|_x$ means a certain norm, $\forall j_0\in R$. Then the trivial solution of Eq. (22) has unanimous asymptotic stability [26]. It is only a sufficient condition for the uniformly asymptotic stability of the solution.

According to Theorems 5.2 and 5.3, a sufficient condition which can ensure that the movement equation of the particles p_i^j based on the CFO method is stable (in other words, each particle p_i^j will converge to its own stable P_i) is shown as follows:

$$|\lambda|=\left|1-\frac{1}{2}\Delta t^2\theta_i^j-c_1r_{1i}^j-c_2r_{2i}^j\right|\le\frac{j+1}{j+2}, j=j_0, j_0+1,\ldots, \tag{29}$$

where λ is the eigenvalue of $A(j)$.

Based on the above sufficient condition, the particles $\{p_i^j\}$ converge to the stable values $\{P_i\}$, $\lim_{j\to\infty}p_i^j=P_i$. Then, we need to prove that all particles using the CFO method converge to a stable P. However, nothing will be said about whether these solutions represent the local or global optimums.

Theorem 5.4. *Under condition of Eq. (29), $\{P_i,\ i=1, 2.\ ,\ N_p\}$ will converge to the optimum solutions (local or global).*

Proof. Based on Eq. (15) and $\lim_{j\to\infty} p_i^j = P_i$, we have

$$\lim_{j\to\infty} \tilde{a}_i^j = \lim_{j\to\infty}(p_i^{j+1} - p_i^j) = P_i - P_i = 0. \tag{30}$$

Further, substitute Eq. (23) into Eq. (30), one has:

$$\lim_{j\to\infty} \tilde{a}_i^j = \frac{1}{2}G\sum_{k\in \boldsymbol{M}_i} \frac{(f(P_k)-f(P_i))^\alpha}{|P_k-P_i|^\beta}(P_k-P_i)\Delta t^2 + c_1 r_{1i}^j(best_{P_i}-P_i) + c_2 r_{2i}^j(best-P_i) = 0, \tag{31}$$

where $best_{P_i}$ means the best position of the i-th particle in the whole update processing; $best$ means the best position of all particles in the whole update processing.

We assume that the particles meet: $best_{P_i} \neq P_i \cup best \neq P_i$. Because of the randomness of r_{1i}^j and r_{2i}^j, Eq. 31 cannot meet in the update processing (Reductio ad absurdum). So you can get that:

$$\sum_{k\in \boldsymbol{M}_i} \frac{(f(P_k)-f(P_i))^\alpha}{|P_k-P_i|^\beta}(P_k-P_i) = 0 \cap best_{P_i} = P_i \cap \text{best} = P_i, \tag{32}$$

For $\forall i$ if it exists: $\text{best} = P_i$, you can get that all particles using the MCFO method converge to a stable optimums solution P

$$P_1 = P_2 = \cdots = P_{N_p} = best = P. \tag{33}$$

So theorem 5.4 is proved.

In conclusion, although there are some random variables (ξ, r_{1i}^j, r_{2i}^j) in the updating process, the convergence of the MCFO method can be guaranteed as long as these random variables satisfy Eq. (29). The range of the eigenvalue in the complex plane is shown in Fig. 5.

6. Path smoothing process

Consistent with solutions proposed in the other similar reference, our solution generates a path composed of line segments [6,7,9]. Although these results can be sufficient for the quad-rotor helicopter, the UAV also needs to stop at every waypoint to guarantee that the dynamic constraints of the UAVs not violated. This forces the UAV to tilt forward and backward repeatedly to accelerate and decelerate which is highly inefficient and contributes to the fast drainage of the batteries. Furthermore, the fact that the UAV needs to stop so frequently increases the flight time unnecessarily, contributing again to the drainage of the batteries.

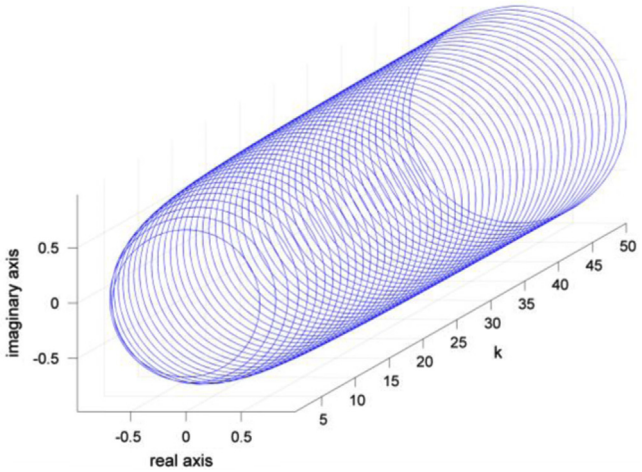

Fig. 5. The range of the eigenvalue in the complex plane.

To remove the most discontinuities in the velocity, we smooth the final path by connecting the line segments with simple space circular arcs. The radius of the circular arcs is computed by the rated speed and the rated acceleration of the UAV. The relationship among the acceleration, curvature radius and velocity is

$$a = \frac{v^2}{\rho} \tag{34}$$

where a is the rated acceleration of the UAV, v is the rated velocity of the UAV, ρ is the curvature radius of the circular arcs. Then, the center of the space circle arcs is obtained by the points of tangency and the broken point. Finally, the coordinates of all points on the circular arcs are obtained by the center and the radius of the space circle arcs.

Of course, there are some special situations making that the smoothed path is not valid. For example, because the line segments stay too close to the obstacles, although these line segments do not bump into the obstacles, the smoothed path passes through the obstacle. The situations of the common and invalid smoothing process are shown in Fig. 6. When these special situations happen at some broken points, the path smoothing of these broken points will be ignored. The quad-rotor UAV also needs to stop at these special broken points, but the occurrence probability of these situations is greatly reduced compared with before. So the flying states of the UAV improve greatly in most situations.

7. Path following process

After the path planning and smoothing process, the reasonableness of the computed path also needs to be verified by the path following process. In the path following process, the real dynamic model of UAVs can be applied to simulate the real UAV flying state. The quadrotor helicopter is used to follow the planning results which include the position information of the UAV [27]. Obviously, the position information is time-varying. But in this paper, the planning results are all the static broken lines which are not time-varying. In the beginning of Section 2, we assume that the UAV maintains the given flight speed scheme which is designed in advance. In this way, the path planning result can turn into time-varying. By this way, the path following process is effective to verify the practicability of the path planning method. The structure of the whole system is shown in Fig. 7.

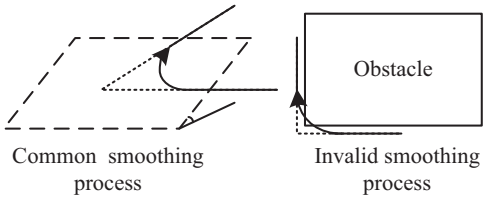

Fig. 6. The sketch map of the common and invalid smoothing process.

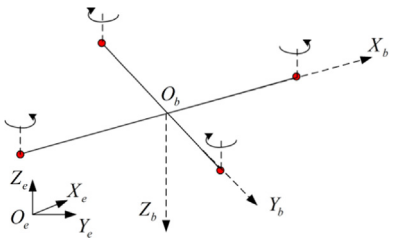

Fig. 7. Quadrotor helicopter structural diagram [27–28].

$O_eX_eY_eZ_e$ and $O_bX_bY_bZ_b$ respectively are the ground coordinate system and the body coordinate system.

The speed scheme is designed as follows: of course, the direction of the speed of the UAV is limited by the path smoothing result. It is obtained by the coordinates of the waypoints. During the moving process, the velocity of the UAV keeps at the rated state in most of the time, including the space circular arcs segment. The velocity of the UAV is 0 m/s when it reaches the special broken points of the path. So we can define the velocity scheme on each smoothed path independently. Shown in Fig. 8 is the velocity scheme of the shortest time.

where θ satisfies $\tan\theta = a_{\text{rated}}$, and a_{rated} is the rated acceleration of the UAV, v_{rated} is the rated velocity of the UAV.

According to this local speed scheme, a whole speed scheme of the path is obtained by the combination. Then, the static broken lines turn into the time-varying position results. Those time-varying position results of the path planning method are the input of the position tracker of the UAV. The position tracker is made up by an inner-loop attitude controller and an outer-loop position controller. The dynamics model of the quadrotor helicopter is built by Ref. [28]. The attitude controller and the position controller are both realized with classical Proportion-Integration-Differentiation (PID) method [29–31]. The PID controllers have many structures, but the most classical one is in the following form:

$$u_{\text{follow}}(t) = K_P e(t) + K_I \int_0^t e(\tau)d\tau + K_D \frac{de(t)}{dt} \tag{35}$$

where $u_{\text{follow}}(t)$ is the control variable, the tracking error $e(t)$ is defined as $e = W_r - W$ where W_r is the scheduled output and W is the real output. K_P, K_I and K_D are controller gains associated with proportional (P), integral (I), and derivative (D) actions, respectively. The structure of the whole system is shown in Fig. 9.

The inner-loop attitude controller is used to control the attitude close-loop of the UAV. Its inputs include the expected time-varying attitude angle obtained by the inverse solution of the outer-loop position controller and the real attitude angle obtained by the real dynamics of the UAV. The outer-loop position controller is applied to track the expected trajectory under the help of the stable attitude loop. The inputs of the outer-loop position controller are the time-varying coordinates of the UAV obtained by the path planning process and the real position information obtained by the real dynamics of the UAV.

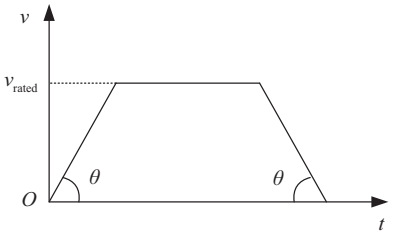

Fig. 8. The sketch map of the velocity scheme.

8. Simulation experiments

The simulation constraint conditions in the examples are as follows:

The length, the width and the height of the planning space are 600 m × 600 m × 100 m. The starting point coordinates value of the UAV is (0 m, 250 m and 0 m). The target point coordinates value of the UAV is (600 m, 600 m, and 40 m). The colorful 3-D geometries mean the obstacles. The abstract scene is shown in Fig. 10.

After the planning space is established, the CFO method, the MCFO 1 method, the MCFO 2 method, the GA, the PSO algorithm, the FA [32] and the random search (RS) way are introduced to solve the UAV path planning problem. The MCFO 1 method means the CFO improved by the mutation operator only. The MCFO 2 method is the whole MCFO method. All these methods are applied by the same initial parameters.

The initialization of the parameters of all search algorithms: the population size is set to 100 ($N_p = 100$); The x-axis is divided into 12 equal portions ($n = 12$); The maximum iterations are 200. The weights of each function are 0.6, 0.2 and 0.2 ($w_1 = 0.6$, $w_2 = 0.2$, $w_3 = 0.2$). The mutation probabilities of the GA, the MCFO 1 method and the MCFO 2 method are all 10% ($c_1 = 0.1$). Because of the randomness of the parameters, the simulation results are usually

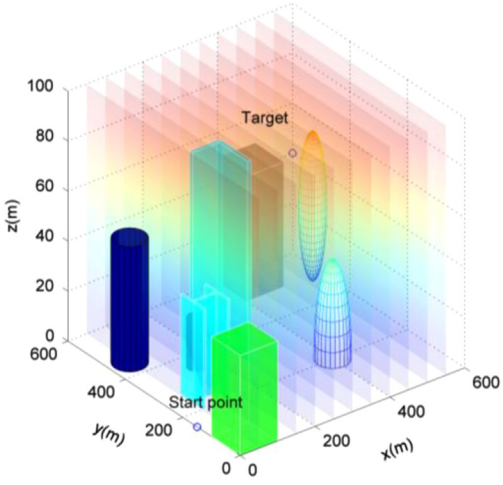

Fig. 10. The abstract scene of the planning space. (For interpretation of the references to color in this figure, the reader is referred to the web version of this article.)

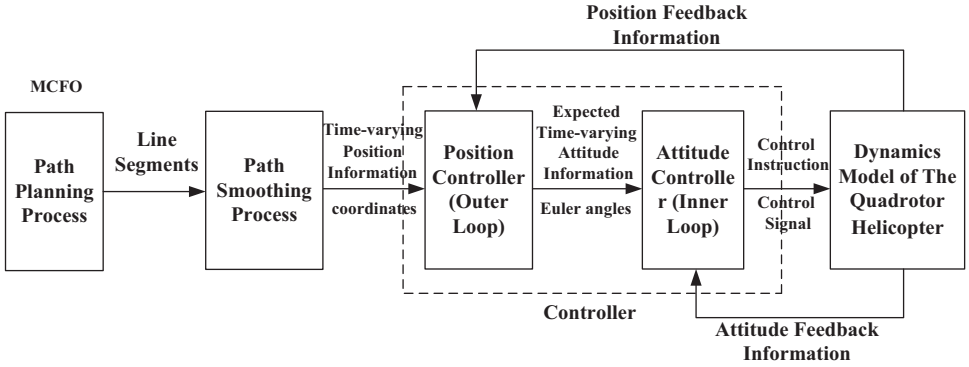

Fig. 9. The whole system chart [27–28].

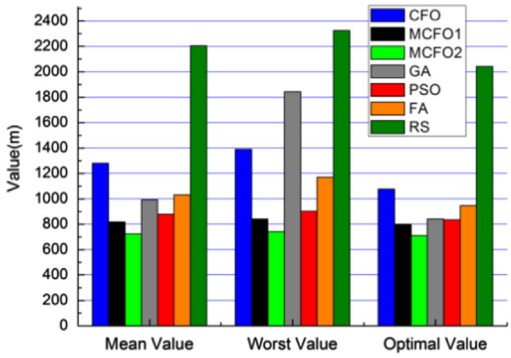

Fig. 11. The statistical results of these algorithms. (For interpretation of the references to color in this figure, the reader is referred to the web version of this article.)

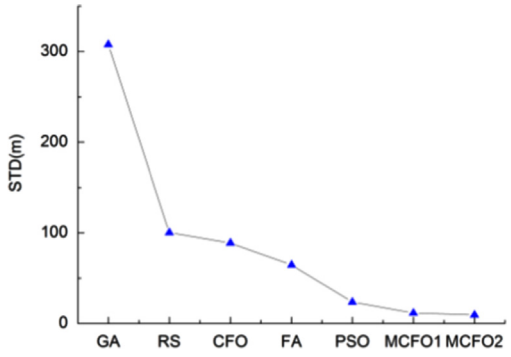

Fig. 12. The STD result values of these algorithms.

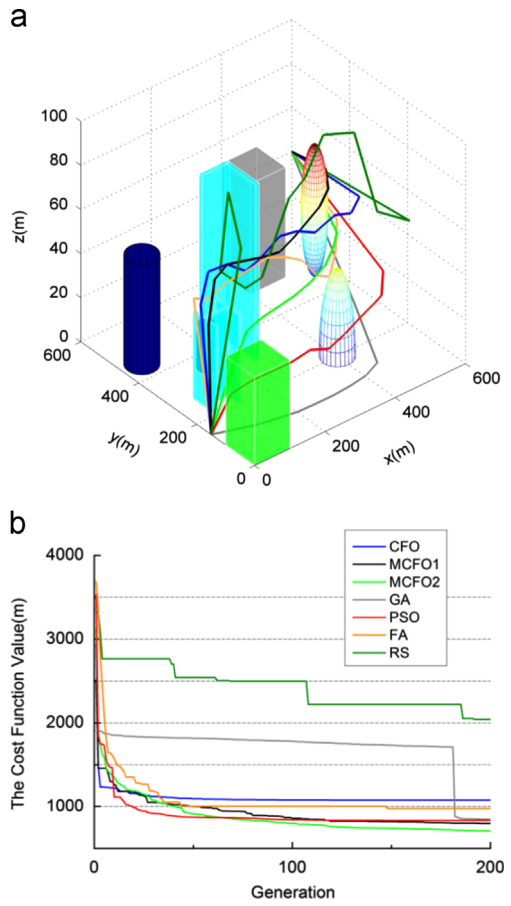

Fig. 13. The best results of each algorithm. (For interpretation of the references to color in this figure, the reader is referred to the web version of this article.)

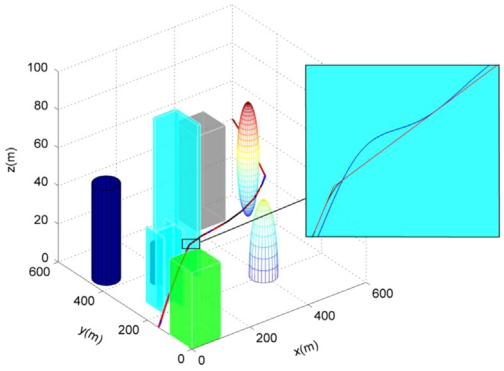

Fig. 14. The path following result. (For interpretation of the references to color in this figure, the reader is referred to the web version of this article.)

different for the same method. If the above factor is taken into consideration, the simulations need to be repeated for many times. The comparative experiments run repeatedly (ten times) for every method. The simple statistical results are shown in Fig. 11.

From these results, we can see that the optimal value, the worst value and the mean value of the cost functions of the MCFO algorithm are better than the ones of the other algorithms obviously. The sequence of the standard deviation (STD) is the GA, the random search way (RS), the original CFO algorithm, the FA, the PSO algorithm, the MCFO1 algorithm and the MCFO2 algorithm. The specific STD value is shown in Fig. 12.

It can thus be seen that the result of the MCFO2 method is much more stable than the others. Above all, the whole new proposed modified CFO algorithm has good global search ability for the UAV path planning problem.

In order to show the difference of these methods intuitively, the best results (including path planning result and cost function curves) in 10 experiments for each method are shown in Fig. 13.

Obviously, we can see that the path planning result of the MCFO2 method is much shorter and smoother than the ones of other algorithms. In order to further verify the feasibility of the path planning result of the MCFO2 method, the path following process is finished by the frame of Section 6. The parameters of the UAV are shown as follows: the rated velocity constraint of the VTOL UAV is $v_{\text{rated}}=5$ m/s; the rated acceleration constraint of the VTOL UAV is $a_{\text{rated}}=0.5$ m/s^2.

The path following result is shown in Fig. 14, where the red line means the real path planning result, the black line means the path smoothing result and the blue line means the UAV path following result. Figs. 14 and 15 indicate that the real UAV dynamic system can track the path smoothing result successfully. The following error can be controlled in 1.2 m which is acceptable for this problem. During the whole path following process, the velocity of the UAV is shown in Fig. 16, where the red imaginary line means the reference velocity scheme; the blue line means the real velocity data. We can see that the real velocity data of the UAV are in good agreement with the reference velocity scheme.

Therefore, the feasibility of the planning result is verified. The control system can help the UAV to follow the UAV path planning result successfully. The results are reasonable and practical.

9. Conclusion

At first, the 3D UAV path planning problem turns into the path-optimization problem with the help of the environment modeling process. The object of the optimization problem is to minimize the multi-objective cost function which includes the minimization of the path length, the minimization of the fuel cost and the best

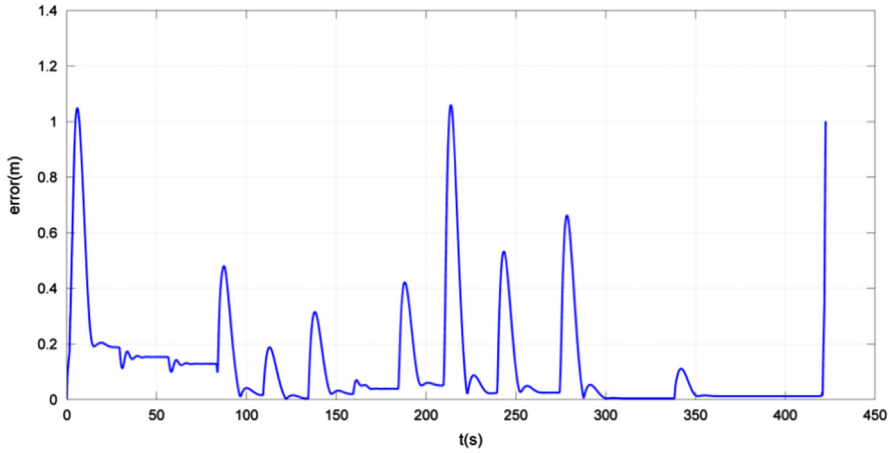

Fig. 15. The following error.

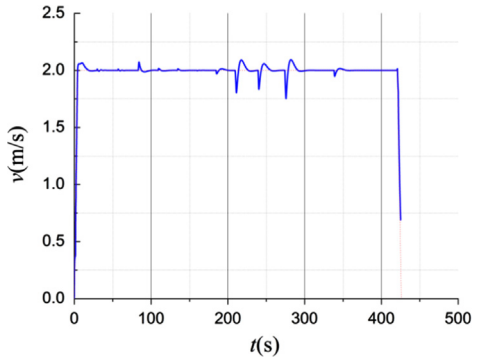

Fig. 16. The flying data of the UAV. (For interpretation of the references to color in this figure, the reader is referred to the web version of this article.)

smoothness of the planning path. Second, a modified central force optimization (MCFO) algorithm is presented for UAV path planning in complicated 3D environments. The ideas of the PSO operator and mutation operator are applied into the traditional CFO method. The whole detailed frame of the MCFO algorithm for the UAV path planning problem is also presented. Third, the convergence analysis of MCFO is developed based on the method in the linear difference equations. At the same time, some convergence conditions are put forward. Then, the path following process based on the quadrotor helicopter control system is put forward to verify the effectiveness and practicality. At last, the simulation results show that the MCFO algorithm can solve the 3D UAV path planning problem successfully. Its performance in these problems is more outstanding than the original CFO algorithm, the GA, the FA, the PSO and the random search way obviously. Most importantly, the path planning result is effective and practical for the real dynamic model of the quadrotor helicopter.

References

[1] D.R. Nelson, D.B. Barber, T.W. McLain, R.W. Beard, Vector field path following for miniature air vehicles, IEEE Trans. Robot. 23 (3) (2007) 519–529.

[2] O. Khatib, Real-time obstacle avoidance for manipulators and mobile robots, Int. J. Robot. Res. 5 (1) (1986) 90–98.

[3] M. Garcia, A. Viguria, A. Ollero, Dynamic graph-search algorithm for global path planning in presence of hazardous weather, J. Intell. Robot. Syst. 69 (2013) 285–295.

[4] J.L. Peng, X.X. Sun, F. Zhu, J. Zhang, 3-D path planning with multi-constrains based on genetic algorithm, in: Proceedings of the 27th Chinese Control Conference, CCC 2008, Kunming, China, 2008, pp. 94–97.

[5] Z.H. Peng, L. Sun, J. Chen, J.P., Wu, Path planning of multiple UAVs low-altitude penetration based on improved multi-agent coevolutionary algorithm, in: Proceedings of the 30th Chinese Control Conference 2011, Yantai, China, 2011, pp. 4056–4061.

[6] V. Roberge, T. Mohammed, L. Gilles, Comparison of parallel genetic algorithm and particle swarm optimization for real-time UAV path planning, IEEE Trans. Ind. Inform. 9 (1) (2013) 132–141.

[7] P.B. Sujit, R. Beard, Multiple UAV path planning using anytime algorithms, in: Proceedings of the 2009 American Control Conference, American Institute of Aeronautics and Astronautics (AIAA), St. Louis, USA, 2009, pp. 2978–2983.

[8] K.P. Valavanis, Advances in Unmanned Aerial Vehicles: State of the Art and Road to Autonomy, Springer, Florida, 2007.

[9] P. Li, H.B. Duan, Path planning of unmanned aerial vehicle based on improved gravitational search algorithm, Sci. China Technol. Sci. 55 (10) (2012) 2712–2719.

[10] R.A. Formato, Central force optimization: a new metaheuristic with applications in applied electromagnetics, Prog. Electromagn. Res. 77 (2007) 425–491.

[11] R.A. Formato, Central force optimization: a new nature inspired computational framework for multidimensional search and optimization, Stud. Comput. Intell. 129 (2008) 221–238.

[12] R.C. Green II, L.-F. Wang, M. Alam, Training neural networks using central force optimization and particle swarm optimization: insights and comparisons, Expert Syst. Appl. 39 (2012) 555–563.

[13] K.R. Mahmoud, Cenlral force optimization: nelder-mead hybrid algorithm for rectangular microstrip antenna design, Electromagnetics 31 (8) (2011) 8866–8872.

[14] Y.Y. Wang, T.T. Wei, X.J. Qu, Study of multi-objective fuzzy optimization for path planning, Chin. J. Aeronaut. 25 (1) (2012) 51–56.

[15] J.J. Liang, H. Song, B.Y. Qu, Performance evaluation of dynamic multi-swarm particle swarm optimizer with different constraint handling methods on path planning problems, in: Proceedings of the 2013 IEEE Workshop on Memetic Computing (MC), Singapore, 2013, pp. 65–71.

[16] R. Esmat, N. Hossein, S. Saeid, GSA: a gravitational search algorithm, Inf. Sci. 179 (2009) 2232–2248.

[17] D.-S. Ding, D.L. Qi, X.P. Luo, J.F. Chen, X.J. Wang, P.Y. Du, Convergence analysis and performance of an extend central force optimization algorithm, Appl. Math. Comput. 219 (2012) 2246–2259.

[18] J. Kennedy, R.C. Eberhart, Particle swarm optimization, in: Proceedings of the IEEE International Conference on Neural Networks, Perth, WA, 1995, pp. 1942–1948.

[19] Q.B. Deng, J.Q. Yu, N.F. Wang, Cooperative task assignment of multiple heterogeneous unmanned aerial vehicles using a modified genetic algorithm with multi-type genes, Chin. J. Aeronaut. 26 (5) (2013) 1238–1250.

[20] F. van den Bergh, A.P. Engelbrecht, A study of particle swarm optimization particle trajectories, Inf. Sci. 176 (8) (2006) 937–971.

[21] D.S. Ding, D.L. Qi, X.P. Luo, J.F. Chen, X.J. Wang, P.Y. Du, Convergence analysis and performance of an extended central force optimization algorithm, Appl. Math. Comput. 219 (4) (2012) 2246–2259.

[22] H.M. Emara, H.A. Al Fattah, Continuous swarm optimization technique with stability analysis, in: Proceeding of the 2004 American Control Conference, Boston, USA, 2005, pp. 2811–2816.

[23] E. Ozcan, C.K. Mohan, Particle swarm optimization: surfing the waves, in: Proceedings of the 1999 Congress on Evolutionary Computation, 1999. CEC 99. Washington, DC, 1999, pp. 1939–1944.

[24] X.L. Jin, L.H. Ma, T.J. Wu, J.X. QIAN, Convergence analysis of the particle swarm optimization based on stochastic processes, Acta Autom. Sin. 33 (12) (2007) 1263–1268.

[25] Y. Xiao, X.Y. Du, Theorem and algorithm of asymptotic stability test for time variant discrete system, J. North. Jiaotong Univ. 22 (6) (1998) 1–7.

[26] W.D. Liao, Z.X. Liu, Stability conditions based on Cauchy-matrix for some classes of time-varying linear difference equations, J. Huazhong Normal Univ. (Nat. Sci.) 35 (4) (2001) 386–389.

[27] Y.B. Chen, J.Q. Yu, X.L. Su, G.C. Luo, Path planning for multi-UAV formation, J. Intell. Robot. Syst. 77 (1) (2015) 229–246.
[28] Y.Q. Bai, H. Liu, Z.Y. Shi, Y.S. Zhong, Robust flight control of quadrotor unmanned air vehicles, Robot 34 (5) (2012) 519–524.
[29] M.Ö. Efe, Neural network assisted computationally simple PIλDμ control of a quadrotor UAV, IEEE Tran. Ind. Inform. 7 (2) (2011) 354–361.
[30] A.P. Schoellig, F.L. Mueller, R.D. Andrea, Optimization-based iterative learning for precise quadrocopter trajectory tracking, Auton. Robots 33 (2012) 103–127.
[31] G.V. Raffo, M.G. Ortega, F.R. Rubio, An integral predictive/nonlinear H1 control structure for a quadrotor helicopter, Automatica 46 (2010) 29–39.
[32] X.S. Yang, Firefly algorithm, stochastic test functions and design optimisation, Int. J. Bio-inspired Comput. 2 (2) (2010) 78–84.

Yongbo Chen received his B.S. degree in Beijing Institute of Technology in 2012. He is currently working toward a Ph.D. degree at the School of Aerospace Engineering, Beijing Institute of Technology, Beijing, China. His research interests include UAV path planning, multi-UAV path planning and UAV mission planning.

Jianqiao Yu received B.S., M.S. and Ph.D. degrees from Beijing Institute of Technology in 1994, 1997 and 2007 respectively. He is presently a Professor with the school of Aerospace Engineering, Beijing Institute of Technology. His main research interests include flight dynamics and control, cooperative control, flight vehicle system design and robust control.

Yuesong Mei was born in 1978. He received his Ph.D. degree in flight vehicle design from Beijing Institute of Technology in 2007. He is now a lecture of School of Aerospace Engineering, Beijing Institute of Technology. His research interests include flight vehicle system design, flight dynamics and control.

Yafei Wang, born in 1987, received B.S. and Ph.D. degrees from Beijing Institute of Technology in 2009 and 2015 respectively. Now he is an engineer of Jiangsu Automation Research Institute, Lianyungang, Jiangsu. His research interests include flight dynamics and control, flight vehicle system design.

Xiaolong Su received B.S. and M.S. degrees from Beijing Institute of Technology in 2012 and 2015 respectively. He is an Assistant Engineer in The 41st Institute of Fourth Academy of Aerospace Science and Technology Corporation, Xian, Shaanxi, China. His research interests include flight dynamics and control and flight vehicle system design.