

# UAV Trajectory Generation Based on Integration of RRT and Minimum Snap Algorithms

Bohui Shi<sup>1</sup>, Youmin Zhang<sup>2\*</sup>, Lingxia Mu<sup>1</sup>, Jing Huang<sup>1</sup>, Jing Xin<sup>1</sup>, Yingmin Yi<sup>1</sup>, Shangbin Jiao<sup>1</sup>, Guo Xie<sup>1</sup>, Han Liu<sup>1</sup>

<sup>1</sup>Shaanxi Key Laboratory of Complex System Control and Intelligent Information Processing  
Xi'an University of Technology, Xi'an, Shaanxi 710048, China

<sup>2</sup>Department of Mechanical, Industrial and Aerospace Engineering, Concordia University, Montreal, Quebec H3G 1M8, Canada

\*Email: youmin.zhang@concordia.ca

**Abstract**—Aiming at the problems that carrying out forest fire monitoring and fighting missions by using the Rapidly-exploring Randomized Tree (RRT) algorithm to plan paths cannot adapt to the autonomous movement of an Unmanned Aerial Vehicle (UAV) and that high-order dynamic characteristics may mutate during the mission, this paper investigates a trajectory generation algorithm based on the integration of the RRT algorithm and the minimum snap algorithm. First, the RRT algorithm is used to generate the initial path, then the minimum snap algorithm is used to smooth the initial path and obtain a trajectory suitable for the actual flight of the UAV. However, because the UAV is considered as a particle in the simulation, during the actual flight, this trajectory may not guarantee the safe flight of the UAV and may cause the UAV to collide with an obstacle or other nearby UAVs in the cases of formation flight. To solve this problem, flight corridor concept is used to limit the UAV's flight trajectory for ensuring the safe flight of the UAV. Simulation results show that the algorithm can effectively ensure the safety, smoothness, feasibility, and trajectory of unmanned aerial vehicles.

**Keywords**—RRT, Minimum snap, Trajectory optimization, Flight corridor, UAV

## I. INTRODUCTION

In recent years, with global warming effects, forest fires have occurred even more frequently worldwide and caused significant losses to the ecological environment, the safety of property and life. Forest fires are sudden and destructive. For forest fire prevention and fire extinguishing work, the most commonly used monitoring and detecting methods mainly use watchtowers and human observers to monitor the surrounding environment, which usually requires a large amount of labor, is limited by time and space, vulnerable to vision limitation of scope, and is likely to threaten personnel safety [1][2]. With the past development of technologies, humans can also monitor forest fires using aircraft and satellites. In forest fire fighting, the current main method is manual fighting, which is primitive, inefficient, and has great potential on safety hazards. In recent years, helicopters have also been used for forest fire fighting due to their fast and efficient response features, but there are still some safety risks for onboard pilots. Because of its advantages of relatively low cost, together with high maneuverability, wide field of view, simple operation, and safety and reliability, UAVs have attracted significant attention of many researchers in recent years. As reported at [3], Li's team at Northwestern Polytechnical University successfully developed the "Tian Lei No. 1" fire extinguishment UAV against forest fire and completed a forest fire fighting test in Yunnan at the end of 2019. It is very important to find a trajectory that is safe, smooth, feasible, and consistent with the dynamic characteristics of the UAV for performing forest fire monitoring and suppression missions.

The trajectory planning performed by the UAV during the execution of the forest fire fighting task means that the UAV finds a safe flight trajectory from the starting point to the terminal point and avoids obstacles under the unknown environments. It is mainly divided into front-end path planning and back-end trajectory optimization. The front-end path planning algorithm uses a related algorithm to generate an initial path from the starting point to the terminal point, but the path may not meet the actual flight of the UAV. Back-end trajectory optimization is based on the initial path planned by the front-end path planning algorithm, and is optimized to produce a safe, smooth, applicable, and collision-free flight trajectory.

Currently, the commonly used front-end path planning algorithms are mainly divided into two categories. 1) Search-based algorithms, such as: Breadth First Search (BFS), Depth First Search [4], Dijkstra [5], A\* [6][7], Jump Point Search [8]. The search-based algorithms describe the environmental information using a graph structure, start from the starting point, and search according to the specified rules step by step. When an obstacle or boundary is encountered, the search ends, otherwise the search continues until the target node is found and the path search is completed. 2) Sampling-based algorithms, such as: Probabilistic Road Map (PRM) [9], Rapidly-exploring Random Tree (RRT) [10][11]. This sampling-algorithm is to: firstly, sample in the map randomly, and find a random sampling point; secondly, based on this point, continue sampling, and generate a search tree until the leaf node of the tree reaches the end point or near the end point that meets the threshold constraint; finally, according to the search tree reverse search, find a path from the beginning to the end. The significant advantage of this algorithm is that it does not rely on the graph structure.

Back-end trajectory optimization algorithms mainly include: B-spline curve [12], Bezier curve [13][17], K-trajectory algorithm [14], and minimum snap [15], etc. Wu [12] proposed to use B-spline curves to optimize the initial path planned by the Artificial Potential Field (APF) algorithm, make the initial path smoother, and easily control the robot to move along the curve. Ding [13] proposed a method combining Bezier curve and A\* algorithm to realize the path planning of robot manipulator. This algorithm solves the problem that the rotation angle of the robot arm is large, and makes the planned path smoother. Li [14] used Levenberg-Marquardt (LM) and K-trajectory algorithms to solve the problem of inherent oscillation when the mobile robot uses APF method to find the path when there are obstacles and narrow channels in the environment processing to reduce the

Fig. 1. Functional diagram of UAV trajectory generation

turbulence in the search process, and the K-trajectory algorithm was used to further smooth the trajectory. Mellinger [15] used the minimum snap algorithm to generate the trajectory of a quadrotor UAV in a strictly restricted indoor environment, ensuring that the UAV meets the constraints on speed, acceleration and input conditions during the flight.

## II. DESIGN OF UAV TRAJECTORY GENERATION ALGORITHM

### 2.1 Trajectory generation algorithm structure

As shown in Fig. 1 for the functional diagram of UAV trajectory generation, the UAV trajectory generation function mainly includes the acquisition of spatial environment information, map construction, front-end path planning, and back-end trajectory optimization.

As shown in Fig. 2 for a particular application to forest fire detection mission, the UAV-based forest fire surveillance system consists of a single UAV and a ground station, to obtain the image information through the UAV's onboard camera(s), to process the image information for constructing an environmental map, and to determine the location of the suspected fire. At the same time, the location information of the UAV is obtained through the GPS system. At the ground station, path planning is carried out based on the space environment information, the information of the suspected fire spot and the location information of the UAV, plan a safe and reliable flight trajectory for the UAV, according to the planned trajectory, control information is sent to the UAV, so as to realize the UAV forest fire surveillance task.

### 2.2 RRT algorithm for front-end path planning

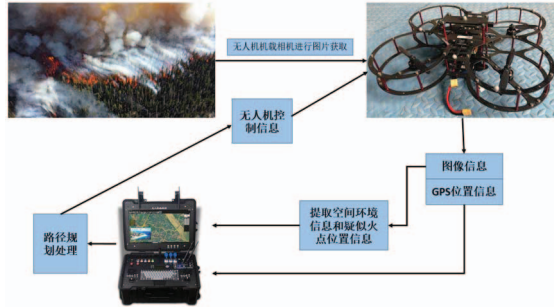


Fig. 2. The UAV-based forest fire inspection system

The advantage of sampling-based path planning algorithm over search-based path planning algorithm is that it does not need to traverse the entire search space. The sampling-based path planning algorithm is to randomly sample points on the map and use line segments that do not pass through obstacles to sample points. They are connected to form a tree; secondly, an obstacle-free path from the starting point to the terminal point is searched backward on the tree structure, thereby improving the efficiency of the algorithm.

As a typical sampling-based path planning algorithm, RRT algorithm is more suitable for path planning in more complex environments and high-dimensional environments. The RRT algorithm is used for front-end path planning in this paper. The RRT algorithm firstly performs random sampling

in the global space to find random points; secondly, it uses line segments that do not pass through obstacles to connect the sampling points for forming a branch of the tree; after that, other branches are generated through iterations, when the target point is added to the tree structure, the iteration process is completed; finally, a traversable path from the starting point to the terminal point is found backward along the tree structure.

The RRT algorithm finds the nearest neighbors by comparing the distance between the random sampling point  $X_{rand}$  and all nodes in the tree to find the node  $X_{near}$  closest to  $X_{rand}$ . The distance  $L$  is calculated using Euclidean distance, that is:

$$L(X_{rand}, X_{near}) = \sqrt{(x_{rand} - x_{near})^2 + (y_{rand} - y_{near})^2} \quad (1)$$

Among them,  $x_{rand}$  and  $y_{rand}$  represent the abscissa and ordinate of the random sampling point  $X_{rand}$ ;  $x_{near}$  and  $y_{near}$  represent the abscissa and ordinate of the nearest adjacent point  $X_{near}$ .

In the following pseudo-code of the RRT algorithm,  $M$  is used for the map,  $X_{init}$  and  $X_{goal}$  are the start and terminal points, and  $T$  is the generated RRT. Firstly, the path is initialized; Secondly, the random sampling point  $X_{rand}$  is generated by the random sampling function:  $Sample(\cdot)$ ; the nearest neighbor search function:  $X_{near}(\cdot)$ , it is used to obtain the nearest node  $X_{near}$ ; generate a new sampling point  $X_{new}$  through the function  $steer(\cdot)$ ; The branch  $E_i$  of the tree is generated by the function  $Edge(\cdot)$ ; for the generated branch  $E_i$ , determine whether the branch  $E_i$  passes an obstacle through the function  $CollisionFree(M, E_i)$ . If it fails, the branch is added to the tree structure. Finally, determine whether the new sampling point is the target point, if the new sampling point is

Input:  $M, X_{init}, X_{goal}$

Result: A path  $T$  from  $X_{init}$  to  $X_{goal}$

$T.init()$ ;

for  $i=1$  to  $n$  do

$X_{rand} \leftarrow Sample(M)$ ;

$X_{near} \leftarrow Near(X_{rand}, T)$ ;

$X_{new} \leftarrow Steer(X_{rand}, X_{near}, SteapSize)$ ;

$E_i \leftarrow Edge(X_{new}, X_{near})$ ;

If  $CollisionFree(M, E_i)$ , then

$T.addNode(X_{new})$ ;

$T.addEdge(E_i)$ .

End

If  $X_{new} = X_{goal}$ , then

Success();

End

End

the target point, complete the tree generation, otherwise, continue to generate new tree branches iteratively.

The advantages of RRT are: 1) It aims to find a passable path from the beginning to the end; 2) It is more targeted than the Probabilistic Road Map (PRM) algorithm. The main disadvantage is that the generated path does not necessarily conform to the dynamic characteristics of the UAV. Therefore, the back-end trajectory needs to be optimized to make the generated trajectory suitable for the actual flight of the UAV.

### 2.3 Minimum snap algorithm for back-end trajectory optimization

The significant disadvantages of path planning based on the RRT algorithm are: the generated path does not meet the dynamic characteristics of the UAV, and the generated trajectory cannot guarantee the smooth flight of the UAV.

In order to better control the movement of the UAV, the back-end trajectory optimization is performed on the basis of the initial trajectory generated by the RRT algorithm. The goal of trajectory optimization is to smooth the processing on the basis of the generated initial path to obtain a safe, feasible, and smooth trajectory, in line with the dynamic characteristics of the UAV [16].

The reasons for the trajectory optimization are: 1) in order to make the generated trajectory more suitable for the autonomous movement of the UAV; 2) the high-order dynamic characteristics of the UAV cannot be abruptly changed; 3) the UAV should not stop moving during a turn; 4) save energy.

#### 2.3.1 Minimum snap trajectory optimization

Polynomials are often used to describe the trajectories of UAVs. In engineering applications, generally, fifth-order polynomials are used to describe the trajectories when minimum jerk is performed, and seventh-order polynomials are used to describe the trajectories when the minimum snap is used, as shown in the following equation. The minimum jerk is to find the minimum value of the target function-jerk of the trajectory. In terms of visual tracking, this function is often used to find the minimum parallax change rate of the UAV during the flight. The smaller the parallax change rate, the better the visual tracking effect. The minimum snap is to find the minimum value of the objective function-snap of the trajectory. In terms of trajectory smoothing, this function is often used to find the minimum thrust change rate of the UAV during flight. The smaller the thrust change, the more conducive to energy saving.

Minimum jerk:

$$\min p(t) = \min(p_0 + p_1 t + p_2 t^2 + p_3 t^3 + p_4 t^4 + p_5 t^5) \quad (2)$$

Minimum snap:

$$\min p(t) = \min(p_0 + p_1 t + p_2 t^2 + p_3 t^3 + p_4 t^4 + p_5 t^5 + p_6 t^6 + p_7 t^7) \quad (3)$$

When performing forest firefighting tasks, the actual energy consumption of the UAV should be as low as possible to ensure that the minimum snap algorithm can be used for trajectory smoothing. Assume that there is an uneven path with a total of 5 nodes. As shown by the polyline in Fig. 3(a), the goal of trajectory optimization is to smooth the trajectory

using an algorithm. The problem formulation and its solution can be represented as follows:

#### 1) Polynomial description of segmented trajectory

In Fig. 3(a), the planned path has 5 nodes and 4 paths in total, that is, there are 4 polynomials, which are:

$$\begin{aligned} p_1(t) &= p_{10} + p_{11}t + p_{12}t^2 + p_{13}t^3 + p_{14}t^4 + p_{15}t^5 + p_{16}t^6 + p_{17}t^7 \\ p_2(t) &= p_{20} + p_{21}t + p_{22}t^2 + p_{23}t^3 + p_{24}t^4 + p_{25}t^5 + p_{26}t^6 + p_{27}t^7 \\ p_3(t) &= p_{30} + p_{31}t + p_{32}t^2 + p_{33}t^3 + p_{34}t^4 + p_{35}t^5 + p_{36}t^6 + p_{37}t^7 \\ p_4(t) &= p_{40} + p_{41}t + p_{42}t^2 + p_{43}t^3 + p_{44}t^4 + p_{45}t^5 + p_{46}t^6 + p_{47}t^7 \end{aligned} \quad (4)$$

#### 2) Time segmentation of segmented trajectories

Common methods for time division include: uniform speed time division, time division based on trapezoidal speed changes, and fixed time division.

#### 3) Solving the coefficient matrix $Q$

Minimum snap transforms the problem of trajectory smoothing into a Quadratic Optimization (QP) problem, and solves the multi-stage trajectory polynomial coefficients to determine the trajectory. The objective function is as follows:

$$\min f(p) = \min(p^{(4)}(t))^2 = \min p^T Q p \quad (5)$$

$$Q = \begin{bmatrix} Q_1 & & & \\ & Q_2 & & \\ & & Q_3 & \\ & & & Q_4 \end{bmatrix} \quad (6)$$

where  $p(t)$  is the  $n$ -th order polynomial of the piecewise trajectory.  $Q$  is a diagonal matrix. By solving the minimum snap of each trajectory,  $Q_i$  of each trajectory can be obtained.

#### 4) Constraints

UAV trajectory smoothing should satisfy two constraints, namely, equality constraints and inequality constraints.

Equality constraints are used to ensure that the UAV can reach the terminal point from the starting point and the position, speed, acceleration, and jerk are continuous, mainly including: position constraints, velocity constraints, acceleration constraints, and jerk constraints.

*Position constraints:*

$$P_i(T) = p_{i+1}(0), i = 1, 2, 3 \quad (7)$$

Among them,  $P_i(T) = p_{i+1}(0)$  ( $i = 1, 2, 3$ ) represents the position information of the  $i$ -th trajectory at time 0 and time  $T$ .

*Velocity constraints:*

$$\dot{p}_i(T) = \dot{p}_{i+1}(0), i = 1, 2, 3 \quad (8)$$

Among them,  $\dot{p}_i(T) = \dot{p}_{i+1}(0)$  ( $i = 1, 2, 3$ ) represents that the final velocity of the previous trajectory is equal to the initial velocity of the next trajectory.

*Acceleration constraints:*

$$\ddot{p}_i(T) = \ddot{p}_{i+1}(0), i = 1, 2, 3 \quad (9)$$

Among them,  $\ddot{p}_1(0) = 0, \ddot{p}_4(T) = 0$  represents that the final acceleration of the previous trajectory is equal to the initial acceleration of the next trajectory.

*Jerk constraints:*

$$\ddot{p}_i(T) = \ddot{p}_{i+1}(0), i = 1, 2, 3 \quad (10)$$

Among them,  $\ddot{p}_1(0) = 0, \ddot{p}_4(T) = 0$  represents that the last jerk of the previous trajectory is equal to the initial jerk of the next trajectory.

### 2.3.2 Minimum snap algorithm with flight corridor constraints

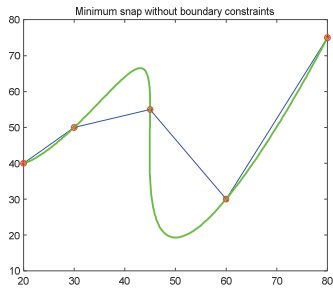
When there are obstacles in the environment, using the ordinary minimum snap algorithm may not guarantee that the UAV can fly safely along the optimized path. In response to this problem, a flight corridor is used to limit the trajectory to ensure the safe flight of the UAV.

For each node of the planned initial path, it can be restricted to a rectangular frame, which has no contact with the obstacle.

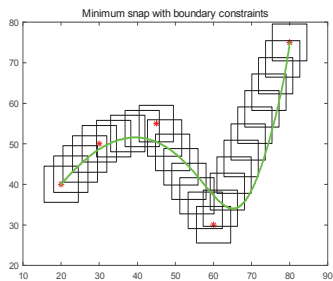
Which is:

$$\begin{cases} x_{\min} \leq p_{ix}(t_i) \leq x_{\max} \\ y_{\min} \leq p_{iy}(t_i) \leq y_{\max} \end{cases} \quad (11)$$

Among them,  $p_{ix}(t_i)$  represents the  $x$  coordinate of the  $i$ -th trajectory at time  $t_i$ , and  $p_{iy}(t_i)$  represents the  $y$ -coordinate of the  $i$ -th trajectory at time  $t_i$ .  $x_{\min}, x_{\max}$  represent the left and right borders of the border, respectively, and  $y_{\min}, y_{\max}$



(a)



(b)

Fig. 3. Graph of trajectory optimization results based on the minimum snap

represent the top and bottom borders of the border, respectively.

The trajectory smoothing result for a 5-segment initial path is shown in Fig. 3(a). When the trajectory is confined in a safe corridor, the optimization result is shown in Fig. 3(b).

### III. ALGORITHM IMPLEMENTATION

Fig. 4 shows the flow chart of the algorithm for trajectory generation of a UAV. The RRT algorithm is used to generate the initial path, and the minimum snap algorithm with equality constraints and inequality constraints is used to optimize the trajectory.

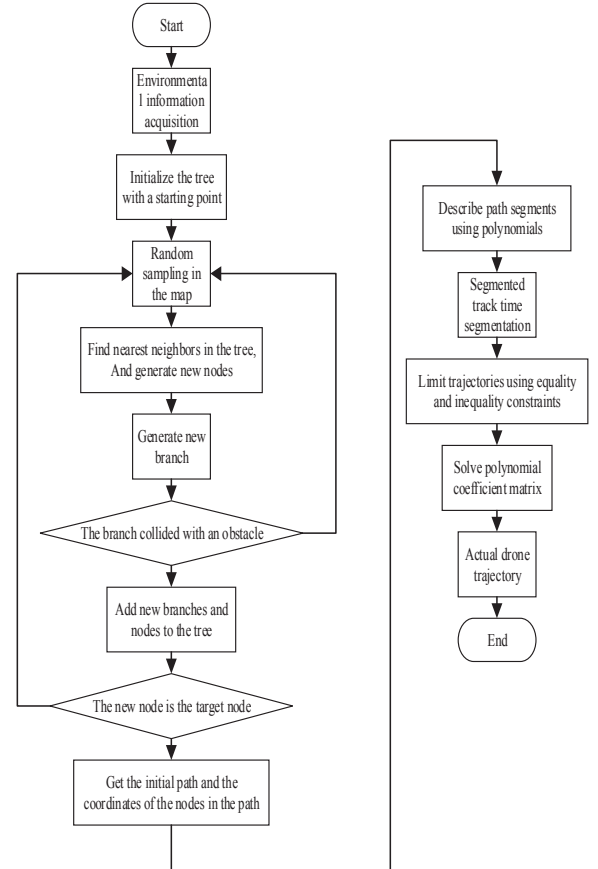


Fig. 4. Flow chart of the algorithm

1) Firstly, use the starting point to initialize the tree structure, and randomly sample the map to generate new sampling points; secondly, calculate the distance between all nodes in the tree and the sampling points by Euclidean distance, and find the nearest neighbors' nodes; then, new nodes and new branches are generated according to certain rules, and the nodes and branches are added to the tree structure;

2) Loop execution. If the new node is not the terminal point, continue sampling and generate a new node. Until the end is found, the complete tree structure is obtained;

3) According to the tree structure, search backward from the terminal point to the starting point to obtain the initial trajectory;

4) The initial trajectory is segmented, and each segmented trajectory is described using a polynomial, and the trajectory



is restricted using equality constraints and inequality constraints to obtain the actual flight trajectory of the UAV.

#### IV. SIMULATION AND EXPERIMENT

##### 4.1 Simulation environment and simulation tasks

Simulation environment: The processor model is: Intel Pentium (R) CPU G2030 @ 3.00GHZ; the operating system is Windows 10;

Simulation software: MATLAB R2018b;

Simulation task: Perform experimental verification on the proposed trajectory generation algorithm to plan a safe, smooth and energy-saving trajectory for the actual flight of the UAV.

##### 4.2 Analysis of simulation results

In order to facilitate the research, in the experiment, the map environment was simplified to a two-dimensional space environment, and the RRT algorithm was used to generate the initial path in different environments. The result is shown in Fig. 5.

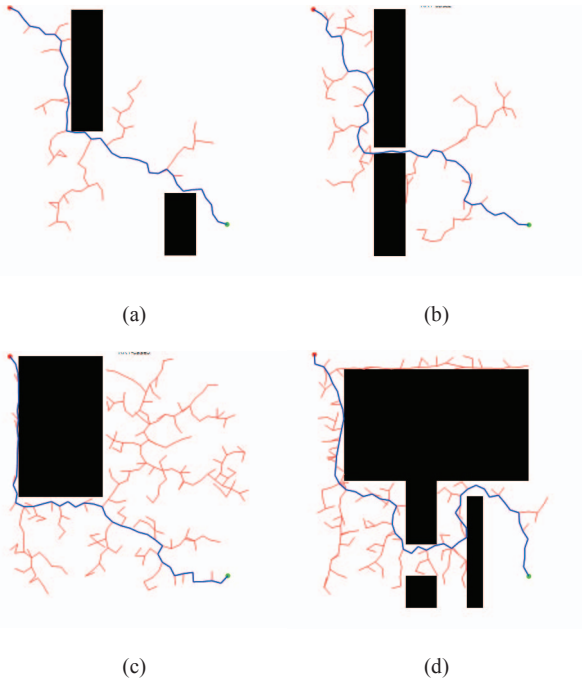


Fig. 5. Path planning based on the RRT algorithm

The complexity of the environment will affect the efficiency of the path generated by the RRT algorithm. The simulation experiment was performed in different environments using the RRT algorithm, and the map size was 800m\*800m. To test the impact of environmental complexity on the RRT algorithm, four types of environment maps are selected. As shown in Fig. 5, Fig. 5(a) is the ordinary environment, Fig. 5(b) is the environment of the narrow passage of the village head, and Fig. 5(c) is the environment where the obstacle is closer to the starting point, Fig. 5(d) is an environment with many obstacles. In the above four environments, using this algorithm can quickly find an initial path from the starting point to the terminal point. The probability of obstacles in the environment, the speed of path finding, and the number of nodes are shown in Table 1. It can be seen from Table 1 that as the complexity of the environment increases, the number of nodes in the initial path

Table 1. RRT path planning results in different environments

Case	Time	Number of nodes	Obstacle coverage (%)	Path length (m)
1	0.395	49	6	1440
2	0.605	45	12	1320
3	0.752	46	19	1350
4	0.699	56	40	1650

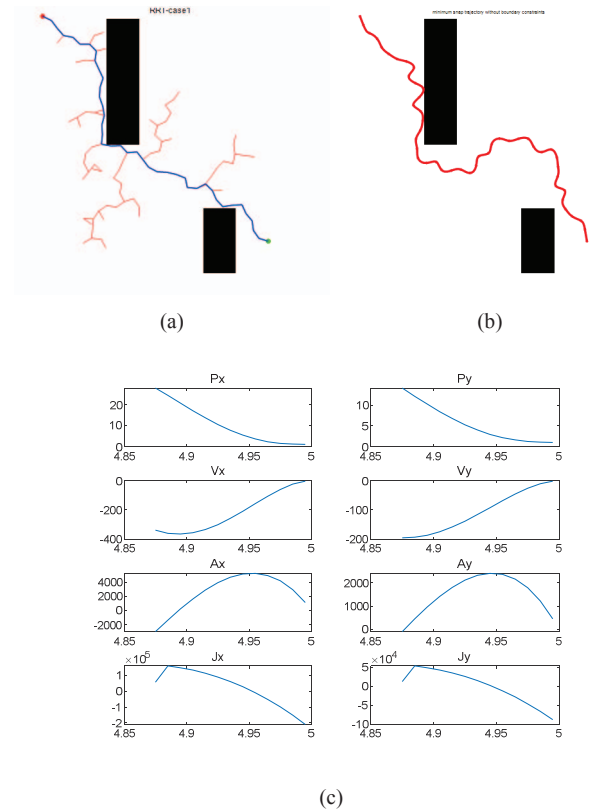


Fig. 6. Comparison of minimum snap results with flight corridor constraints

generated by the RRT algorithm increases, the path length also increases, and the smoothness of the path decreases.

The minimum snap algorithm was used to optimize the initial path planned by the RRT algorithm. The optimization results were compared with the initial path planned by the RRT algorithm. In the simulation, it is assumed that  $p$ ,  $v$ ,  $a$ , and the jerk at the initial time are all 0. It is assumed that the polynomial order after each trajectory optimization is 7. The optimized speed, acceleration, and jerk are suitable for the actual flight of the UAV.

The trajectory map optimized using the minimum snap algorithm is shown in Fig. 6. Fig. 6(a) shows the results of RRT trajectory optimization without boundary constraints, and Fig. 6(b) shows the results of RRT trajectory optimization planning with flight corridor constraints. Fig. 6(c) is a graph of the position, speed, acceleration, and jerk of the UAV. From this graph, it can be concluded that the UAV consumes less energy during flight. In actual flight, UAVs cannot be considered only as particles, and the size of the UAV itself needs to be considered.

By comparing the two paths in Fig. 7(a) and Fig. 7(b), it is concluded that both can ensure the smooth flight of the UAV. Fig. 7(a) is the flight trajectory planned using the minimum snap algorithm without the flight corridor constraints. It can be seen that the UAV will collide with obstacles during flight. Fig. 7(b) shows the flight trajectory planned using the minimum snap algorithm with flight corridor constraints. The planned trajectory meets the requirements for smooth and safe flight of the UAV.

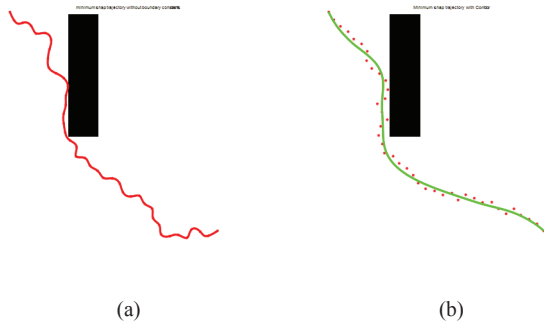


Fig. 7. Minimum snap results with flight corridor constraints

## V. CONCLUSION

Through simulation experiments on the trajectory generation algorithm proposed in this paper, the following conclusions can be obtained:

1) Using the RRT algorithm, one can quickly plan an initial path from the starting point to the terminal point for the UAV under different flight environments. The algorithm is highly adaptable. However, as the complexity of the environment increases and the initial path has more vertices, the smoothness will decrease.

2) To be suitable to the actual flight of the UAV, using the minimum snap algorithm with flight corridor constraints can generate a safe, feasible and smooth flight trajectory, which can also ensure that the UAV save energy during flight. The minimum snap algorithm uses simple polynomials for trajectory smoothing, and by taking into consideration to the position, velocity, acceleration and jerk constraints of the UAV, the resulting smooth trajectory is more suitable for the actual flight of the UAV; on this basis, by using the flight corridor to restrict the flight trajectory, collisions with obstacles can be avoided and the safe flight of the UAV can be ensured.

As a summary, it can be seen that the trajectory generated by the integrated algorithm of RRT and minimum snap, and by taking into account the UAV's dynamic characteristics, safety, smoothness, and other factors, the resulting trajectory is more suitable for the actual flight of the UAV for the practical mission of forest fires surveillance.

## ACKNOWLEDGEMENTS

This work is partially supported by the National Natural Science Foundation of China (Grants 61833013 and 61573282), Natural Science Foundation of Shaanxi Province (Grant 2019JQ-751), and Natural Sciences and Engineering Research Council of Canada.

## REFERENCES

[1] C. Yuan, Y. M. Zhang, and Z. X. Liu, "A survey on technologies for automatic forest fire monitoring, detection, and fighting using unmanned aerial vehicles and remote sensing techniques," Canadian

Journal of Forest Research, 2015, 45(7): 783-792.

[2] Z. T. Jiao. Research on Forest Fire UAV System. Xi'an University of Technology, Master Thesis, 2019.

[3] Z. K. Li. Substantial progress has been made in the practical use of forest fire extinguishing drones. <https://news.nwpu.edu.cn/info/1003/66062.htm>, 2019-10-23.

[4] E. Žunić, A. Djedović and B. Žunić, "Software solution for optimal planning of sales persons work based on Depth-First Search and Breadth-First Search algorithms," The 39th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO), Opatija, 2016, pp. 1248-1253.

[5] Z. Nie and H. Zhao, "Research on robot path planning based on Dijkstra and ant colony optimization," 2019 International Conference on Intelligent Informatics and Biomedical Sciences (ICIIBMS), Shanghai, China, 2019, pp. 222-226.

[6] T. Chen, G. Zhang, X. Hu and J. Xiao, "Unmanned aerial vehicle route planning method based on a star algorithm," The 13th IEEE Conference on Industrial Electronics and Applications (ICIEA), Wuhan, 2018, pp. 1510-1514.

[7] H. Su, H. Yun, J. He, F. Zhang and Y. Wang, "Multi-aircraft path planning method based on cooperative search A-star algorithm," 2019 IEEE International Conference on Unmanned Systems (ICUS), Beijing, China, 2019, pp. 267-272.

[8] L. Ma, X. Gao, Y. Fu and D. Ma, "An improved jump point search algorithm for home service robot path planning," 2019 Chinese Control and Decision Conference (CCDC), Nanchang, China, 2019, pp. 2477-2482.

[9] R. Al-Hmouz, T. Gulrez and A. Al-Jumaily, "Probabilistic road maps with obstacle avoidance in cluttered dynamic environment," Proceedings of the 2004 Intelligent Sensors, Sensor Networks and Information Processing Conference, 2004, Melbourne, Australia, 2004, pp. 241-245.

[10] Y. Wang and Y. Huang, "Mobile robot path planning algorithm based on rapidly-exploring random tree," 2019 IEEE International Conferences on Ubiquitous Computing & Communications (IUCC) and Data Science and Computational Intelligence (DSCI) and Smart Computing, Networking and Services (SmartCNS), Shenyang, China, 2019, pp. 555-560.

[11] Y. Q. Dong, Y. M. Zhang, J. L. Ai, "Experimental test of unmanned ground vehicle delivering goods using RRT path planning algorithm," Unmanned Systems, 2017, 5(1): 45-57.

[12] Z. Wu, W. Su and J. Li, "Multi-robot path planning based on improved artificial potential field and B-spline curve optimization," 2019 Chinese Control Conference (CCC), Guangzhou, China, 2019, pp. 4691-4696.

[13] H. Ding, Y. Li, Y. Chai and Q. Jian, "Path planning for 2-DOF manipulator based on Bezier curve and A\* algorithm," 2018 Chinese Automation Congress (CAC), Xi'an, China, 2018, pp. 670-674.

[14] G. S. Li, W. S. Chou. An improved potential field method for mobile robot navigation. High Technology Letters, 2016, 22(1):16-23.

[15] D. Mellinger and V. Kumar, "Minimum snap trajectory generation and control for quadrotors," 2011 IEEE International Conference on Robotics and Automation, Shanghai, 2011, pp. 2520-2525.

[16] R. Chai, A. Savvaris, A. Tsourdos, Y. Xia, S. Chai. "Solving multiobjective constrained trajectory optimization problem by an extended evolutionary algorithm," IEEE Transactions on Cybernetics, 2020, 50(4): 1630-1643.

[17] W. Zhou, X. P. Wang, H. S. Sun, Y. Chen. Trajectory optimization of quadrotor UAV based on Bezier curve. Journal of Electronic Measurement and Instrument, 2019, 33(10): 53-58.

[18] P.-C. Sara, S. Jürgen, R. Bernhard, A.L.-O. José, B.-P. Eva. UAV trajectory optimization for minimum time search with communication constraints and collision avoidance. Engineering Applications of Artificial Intelligence, 2019, 85.

[19] Y. C. Yao, H. Zhou, W. L. Gu, S. Tian. A summary of UAV path planning methods based on hybrid algorithms. South Agricultural Machinery, 2018, 49(12): 2-5.

[20] D. Liu. Research and Application of Multi-UAV Route Planning Based on Improved Ant Colony Algorithm. Yantai University, 2018.