Journal of Physics: Conference Series

PAPER • **OPEN ACCESS**

Optimal Formation Control for Quadrotors with Collision Avoidance Based on Dynamic Constraints

IOP Conf. Series: Journal of Physics: Conf. Series **1215** (2019) 012018 doi:10.1088/1742-6596/1215/1/012018

Optimal Formation Control for Quadrotors with Collision Avoidance Based on Dynamic Constraints

P F Jin[1], J Q Yu[1], Z Y Jia[1] and D Yang[1]

[1]School of Aerospace Engineering, Beijing Institute of Technology, Beijing 100081, China

Abstract. This paper studied the optimal formation control of quadrotor UAV based on the dynamic model, and the collision avoidance between quadrotors is considered. By constructing the problem into a standard convex quadratic programming problem, we hope to improve the solving efficiency of the formation control problem. Firstly, the nonlinear dynamic model of quadrotor is linearized and the prediction model is established. Then, the safety zone constraints are transformed from a circular zone to a half-plane zone, making the optimization problem be a standard convex quadratic programming problem. Finally, the quadratic programming problem is solved using distributed receding horizon optimization. Numerical simulations in three-dimensional space show that this method can obtain the optimal formation trajectory with collision avoidance, and can improve the solving efficiency.

1 Introduction

In the field of optimal formation control for multiple unmanned aerial vehicles (UAVs), model predictive control (MPC) is often used to solve optimization problems, usually based on kinematic models. Researchers studied the formation trajectory optimization of multiple UAVs based on the particle kinematics model [[1]]. The dynamic characteristic of quadrotor UAV is quite different from fixed-wing aircraft. It is necessary to study the formation control of quadrotors according to its 6-DOF dynamic model. Quadrotor's dynamics has strong nonlinearity, so the direct applying of nonlinear dynamic model will increase the difficulty of solving optimization. Linearizing the dynamic model can help the optimization problem be solved quickly under the linear MPC framework.

Collision avoidance should be considered during formation control. Researchers adopt some intelligent optimization algorithms, such as A* algorithm[[2]], Ant Colony Algorithm (ACA) [[3]], Particle Swarm Optimization (PSO) [[4]], the Artificial Potential Field (APF) [[5]], searching for cooperative formation control of multiple UAVs. However, the trajectories solved by these intelligent optimization algorithms are generally not optimal, and they usually consume large computing resources. Another feasible solution for the formation control problem is the Mixed Integer Programming (MILP) [[6]], which transforms the problem into a mixed integer linear programming model by introducing auxiliary decision variables, and then uses CPLEX, a relatively mature optimization tool, to solve the problem centrally. In [[7]], the collision avoidance constraints are linearized into the range of abscissa and ordinate coordinates, and the receding horizon control is used to reduce the calculation time, which makes the real-time trajectory planning with collision avoidance be realized.

When the performance index function is a quadratic function with linear constraints, the quadratic programming algorithm can be used to solve the optimization, which is more efficient than intelligent algorithm. However, the circular safety zone constraints of quadrotors are nonlinear constraints and cannot be accepted by quadratic programming algorithm. In this paper, the safety zone constraints are transformed from a circular zone into a half-plane zone, which can be expressed by a linear inequality, and the optimal variables belong to a nonempty closed convex set. Then the optimal problem is transformed into a standard convex quadratic programming problem, which can be solved by mature quadratic programming algorithms.

2 Modelling

2.1 Linear dynamic model of a quadrotor UAV

For a single quadrotor UAV, the following continuous time particle kinematics model is used to describe it [[8]]:

$$\begin{cases} \ddot{x} = (\sin\phi\sin\psi + \cos\phi\sin\theta\cos\psi)u_z \\ \ddot{y} = (\cos\phi\sin\theta\sin\psi - \sin\phi\cos\psi)u_z \\ \ddot{z} = (\cos\phi\cos\theta)u_z - g \\ \ddot{\phi} = u_\phi \\ \ddot{\theta} = u_\theta \\ \ddot{\psi} = u_\psi \end{cases} \tag{1}$$

Based on the following operations: 1) small angle assumption; 2) omitting high-order minor terms; 3) applying $u_z = g$ to horizontal motion, we can obtain

$$\dot{s}_i = \bar{A}s_i + \bar{B}u_i + \bar{C} \tag{2}$$

In the formula (2),

$$\bar{A} = \begin{bmatrix} 0&1&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&9.81&0&0&0 \\ 0&0&0&1&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&-9.81&0&0&0&0&0 \\ 0&0&0&0&0&1&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&1&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&1&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&1 \\ 0&0&0&0&0&0&0&0&0&0&0&0 \end{bmatrix}, \bar{B} = \begin{bmatrix} 0&0&0&0 \\ 0&0&0&0 \\ 0&0&0&0 \\ 0&0&0&0 \\ 0&0&0&0 \\ 1&0&0&0 \\ 0&0&0&0 \\ 0&1&0&0 \\ 0&0&0&0 \\ 0&0&1&0 \\ 0&0&0&0 \\ 0&0&0&1 \end{bmatrix}, \bar{C} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ -9.81 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{3}$$

s_i is the state vector of the quadrotor i, and $s_i = [x_i; v_{xi}; y_i; v_{yi}; z_i; v_{zi}; \phi_i; p_i; \theta_i; q_i; \psi_i; r_i]$, including the current position, speed, attitude angle and angular velocity of the quadrotor. u_i is the control input, and $u_i = [u_{zi}; u_{\phi i}; u_{\theta i}; u_{\psi i}]$. Discretize (2) and the following discrete model is obtained:

$$s_i(k+1) = As_i(k) + Bu_i(k) + C \tag{4}$$

In (4), there is

$$A = I + \bar{A}\Delta t, \quad B = \bar{B}\Delta t, \quad C = \bar{C}\Delta t \tag{5}$$

2.2 Prediction model and performance index function

The core idea of MPC is to use a known model, the current state of the system and future control sequence to predict the future output of the system. The output length is an integral multiple of the control period [[9]]. Since the future control is unknown, it is necessary to optimize the problem according to certain optimization conditions to obtain the future control sequence.

In the prediction time domain, at time k, the state sequence and control sequence of the quadrotor i domain are defined as

$$\tilde{s}_i(k) = [s_i(k+1\mid k); s_i(k+2\mid k); \cdots ; s_i(k+N\mid k)] \tag{6}$$

$$\tilde{u}_i(k) = [u_i(k\mid k); u_i(k+1\mid k); \cdots ; u_i(k+N-1\mid k)] \tag{7}$$

N is the prediction time domain length. The prediction model during flight can be expressed as

$$\tilde{s}_i(k) = Es_i(k) + F\tilde{u}_i(k) + G \tag{8}$$

In (8), there is

$$E = \begin{bmatrix} A \\ A^2 \\ \vdots \\ A^N \end{bmatrix}, F = \begin{bmatrix} B & 0 & \cdots & 0 \\ AB & B & \cdots & 0 \\ \vdots & \vdots & & 0 \\ A^{N-1}B & A^{N-2}B & \cdots & B \end{bmatrix}, G = \begin{bmatrix} I \\ A+I \\ \vdots \\ A^{N-1}+A^{N-2}+\cdots+1 \end{bmatrix} C \tag{9}$$

The control objective of the problem is to minimize the control energy under various constraints, so in the prediction time domain, the performance index function of the quadrotors system with the control $\tilde{u}_i(k)$ can be

$$J(\tilde{u}_i(k)) = \sum_{i=1}^{N_V} \tilde{u}_i'(k)\tilde{Q}\tilde{u}_i(k) \tag{10}$$

In (10), $\tilde{Q} = I_N \otimes Q, Q = (diag(1,0,0,0))$. N_V is the number of quadrotors. Equation (10) is a standard quadratic function, and constructing the standard quadratic programming problem will facilitate the quick solving of the optimization.

2.3 Constraints

There are many constraints that must be satisfied in the formation configuration of quadrotors. This paper takes three types of constraints into consideration, including terminal state constraints, platform performance constraints, and inter-aircraft collision avoidance constraints.

(1) Terminal state constraints require that each quadrotor must reach the reference target point at the terminal time to form and maintain a specific formation, which is expressed as

$$s_i(k+N\mid k) = \theta + \Delta\theta_i \tag{11}$$

In (11), θ is reference state vector for quadrotors formation in target point, and $\Delta\theta_i$ represents the formation structure, including the relative position and velocity vector.

(2) The platform performance constraints require that the values of velocity, attitude angle and control input during flight do not exceed their limits. In order to simplify the processing, only the components in the horizontal are considered. The velocity vector is projected to different directions to obtain $\forall m \in [1,\cdots,M]$, $\forall i \in [1,\cdots,N_V]$

$$v_{xi}\cos\left(\frac{2m\pi}{M}\right) + v_{yi}\sin\left(\frac{2m\pi}{M}\right) \le v_{\max} \tag{12}$$

M is a positive integer. As for the attitude angle limits and control input limits, there are

$$-\phi_{\max} \le \phi_i \le \phi_{\max}, -\theta_{\max} \le \theta_i \le \theta_{\max}, -\psi_{\max} \le \psi_i \le \psi_{\max} \tag{13}$$

$$-u_{\phi\max} \le u_{\phi i} \le u_{\phi\max}, -u_{\theta\max} \le u_{\theta i} \le u_{\theta\max}, -u_{\psi\max} \le u_{\psi i} \le u_{\psi\max}, -u_{z\max} \le u_{zi} \le u_{z\max} \tag{14}$$

(3) Collision avoidance constraints: In the 2D case, define $pos_i(k)$ and $pos_j(k)$ as the position vector of the quadrotor i and quadrotor j at time k, and $pos_i(k) = [x_i(k); y_i(k)]$. $pos_i(k+1)$ represents the position vector of the quadrotor i at the next time. The safety zone is generally a circle whose center is on the quadrotor, with a safety distance R. The constraint of a circle is a quadratic inequality, and the quadratic programming problem with quadratic inequality constraints and non-convex optimization variables is not easy to find the optimal solution. Consider changing the safety zone from a circle constraint to a half-plane constraint, as shown in figure 1.

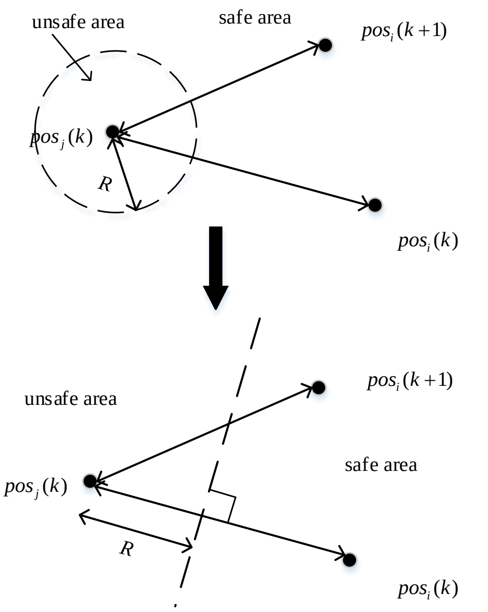

Figure 1. The collision avoidance constraints in 2D case

For the half-plane constraints in figure 1, the position of the quadrotor must satisfy

$$(pos_i'(k+1) - pos_j'(k))(pos_i(k) - pos_j(k)) \ge \| pos_i(k) - pos_j(k) \|_2 \ R, \ \ i, j = 1, \cdots N_V, j \ne i \tag{15}$$

Based on the above analysis, the performance index function of trajectory optimization for the quadrotors formation with collision avoidance is expressed as

$$J(\tilde{u}_i(k)) = \sum_{i=1}^{N_V} \sum_{n=0}^{N-1} u_i'(k+n \mid k) Q u_i(k+n \mid k) \tag{16}$$

Therefore, the optimization model is

$$\begin{aligned} &\min_{u_1, u_2, \cdots, u_{N_V}} \ J(\tilde{u}_i(k)) \\ &\text{s.t.} \\ &\begin{cases} h(\tilde{s}_i(k), \tilde{u}_i(k)) = 0, & \forall i = 1, 2, \cdots, N_V \\ g(\tilde{s}_i(k), \tilde{u}_i(k)) \le 0, & \forall i = 1, 2, \cdots, N_V \\ f(\tilde{s}_i(k), \tilde{s}_j(k), \tilde{u}_i(k)) \le 0, \forall i, j = 1, 2, \cdots, N_V, i \ne j \end{cases} \end{aligned} \tag{17}$$

In (17), $h(\tilde{s}_i(k), \tilde{u}_i(k))$ is the terminal state constraints, $g(\tilde{s}_i(k), \tilde{u}_i(k))$ is the platform performance constraints, and $f(\tilde{s}_i(k), \tilde{s}_j(k), \tilde{u}_i(k))$ is the collision avoidance constraints between quadrotors. The above performance index function is a quadratic function, constrained by some linear equalities and linear inequalities, and can be optimized by mature quadratic programming algorithm to obtain the optimal solution.

3 Optimal formation control algorithm

The optimization model (17) is a centralized model. As the number of quadrotors increases, the number of optimization decision variables and constraints increases, resulting in a sharp increase in computational complexity and unsatisfactory on-line computational efficiency. Moreover, for quadrotors system, there is no centralized optimization node in many cases. In order to avoid the system failure caused by centralized node's problem and reduce the scale of the optimization problem, a distributed algorithm is proposed for quadrotors.

Due to the possible model mismatch and disturbance, the actual state of the quadrotor in flight will deviate from the predicted state. In order to further improve the computational efficiency, the distributed trajectory optimization and receding horizon control are combined to carry out distributed receding horizon optimization, that is: at any time, based on the current state, plan the future trajectory over a period of time, solve the optimal control sequence. Only the first item of the control sequence is applied to the platform, and this solution process will be repeated the next time.

3.1 Model of the distributed receding horizon optimization

A communication-based distributed receding horizon optimization model is used to decompose the centralized optimization problem into a local trajectory optimization problem for each quadrotor. Specifically, for the quadrotor i, the performance index function of local optimization is

$$J_i(\tilde{u}_i(k)) = \sum_{n=0}^{N-1} u_i'(k+n\mid k) Q u_i(k+n\mid k) \tag{18}$$

The distributed optimization model is

$$\begin{aligned} &\min_{u_i} J_i(\tilde{u}_i(k)) \\ &\text{s.t.} \\ &\begin{cases} h(\tilde{s}_i(k), \tilde{u}_i(k)) = 0 \\ g(\tilde{s}_i(k), \tilde{u}_i(k)) \le 0 \\ f(\tilde{s}_i(k), \tilde{s}_j(k), \tilde{u}_i(k)) \le 0, \forall j \in N_i \end{cases} \end{aligned} \tag{19}$$

In (19), N_i is a set of the quadrotor nodes within the communication distance of the quadrotor i. From the above model, it can be seen that the optimization problem of the quadrotor i is only related to the local decision variable $\tilde{u}_i(k)$, which means the scale of the optimization problem is greatly reduced. In the process of the distributed solution, each quadrotor need to communicate with other quadrotors and exchange their solution information. The information transmitted by each quadrotor includes the state sequence and control sequence in prediction time domain.

3.2 Optimal formation control based on MPC

The cooperative collision avoidance of a quadrotors system is essentially a cooperative game problem of multi-objective and multi-player [[10]]. The Nash optimal trajectories for a quadrotors system with collision avoidance is defined as: for an arbitrary quadrotor i in the system, there is a control sequence $\tilde{u}_i^* = [u_i^*(k\mid k);\cdots;u_i^*(k+N-1\mid k)]$, can satisfy the following inequality for any control sequence

$$J_i(\tilde{u}_i^*, \tilde{u}_{j\neq i}^*) \le J_i(\tilde{u}_i, \tilde{u}_{j\neq i}^*) \tag{20}$$

Then $\tilde{u}_i^*$ is the Nash optimal control sequence of quadrotor i for the quadrotors system.

The Nash optimal control sequence can be obtained by the following iterative process: First, at time k, each quadrotor estimates the initial control sequence and state sequence and sends them to other quadrotors through communication network. On the basis of obtaining the control sequence and state sequence of other quadrotors, each quadrotor can obtain the needed control sequence at present iteration by solving the standard quadratic programming problem. If the control sequence does not satisfy (20), it indicates that the whole system does not reach Nash equilibrium. Further iteration can make the local and global performance better until the system reaches Nash equilibrium. After that, the control sequence of each quadrotor is the Nash optimal control sequence.

4 Simulations

In order to verify the rationality of this algorithm, three unmanned aerial vehicles are taken as an example to conduct in-plane formation simulation experiments. Solve the optimal trajectories in the process of formation configuration without/with collision avoidance respectively. The simulation hardware performance is: Intel Core i5-2400 CPU, 3.10GHz, 4GB RAM. The simulation software environment is MATLAB.

The initial state vectors of three UAVs are

$$s_1(0) = [50\text{m}, 20\text{m/s}, 50\text{m}, 0\text{m/s}, 0\text{m}, 0\text{m/s}, 0\text{rad}, 0\text{rad/s}, 0\text{rad}, 0\text{rad/s}, 0\text{rad}, 0\text{rad/s}]^{\mathrm{T}}$$
$$s_2(0) = [450\text{m}, -20\text{m/s}, 450\text{m}, 0\text{m/s}, 0\text{m}, 0\text{m/s}, 0\text{rad}, 0\text{rad/s}, 0\text{rad}, 0\text{rad/s}, 0\text{rad}, 0\text{rad/s}]^{\mathrm{T}}$$
$$s_3(0) = [50\text{m}, 0\text{m/s}, 450\text{m}, -20\text{m/s}, 0\text{m}, 0\text{m/s}, 0\text{rad}, 0\text{rad/s}, 0\text{rad}, 0\text{rad/s}, 0\text{rad}, 0\text{rad/s}]^{\mathrm{T}}$$

The formation reference target point is (322 m, 250 m, 50 m), and the positions of three quadrotors relative to the reference point are (-20 m, 20 m, 0 m), (-20 m, -20 m, 0 m), (0 m, 0 m, 0 m) limited by the formation. All three quadrotors are required to have terminal speed vector (20 m/s,0 m/s, 0 m/s), with maximum speed not exceeding 25 m/s and maximum accelerations not exceeding 3.5 m/(s^2) in flight. The safe distance between aircraft to avoid collision is set to 25 m. Sampling interval is 0.5 sec. The decision-making interval for trajectory optimization is 1 sec. The prediction time domain length is 20 sec, that is, every predictive trajectory consist of 40 sampling points. Figure 2 show the optimal formation trajectories with considering collision avoidance.

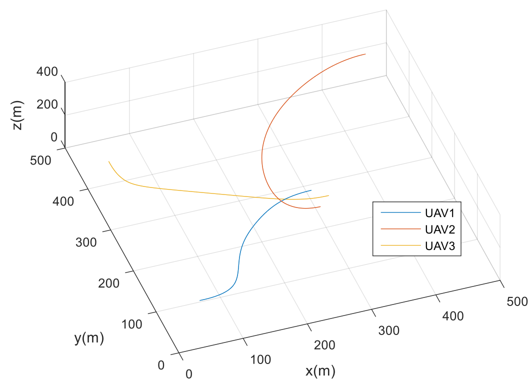

Figure 2. The optimal formation trajectories with considering collision avoidance

It can be seen that the optimal formation control can be obtained by solving the quadratic programming problem after changing the safety zone from a circle to a half-plane. Three quadrotors reach the reference target point at the designated time and constitute the required formation successfully.

Figure 3 and figure 4 show the distance between each two quadrotors in the two simulation cases. The minimum distances in flight are 11.57 m, 15.53 m and 17.05 m without considering collision avoidance. After adding the half-plane collision avoidance constraint, the distance between any two quadrotors is not less than 25 m, which means the collision avoidance is realized successfully.

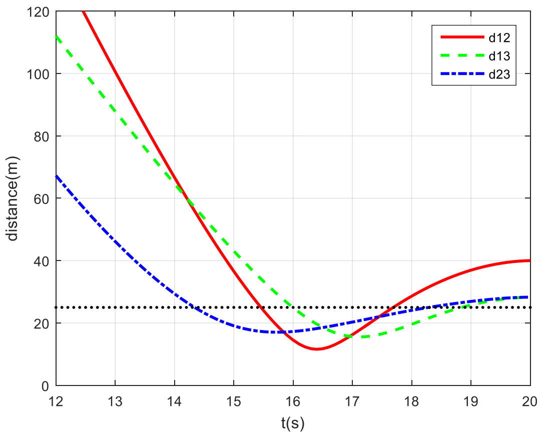

Figure 3. The distances between the quadrotors during flight without collision avoidance

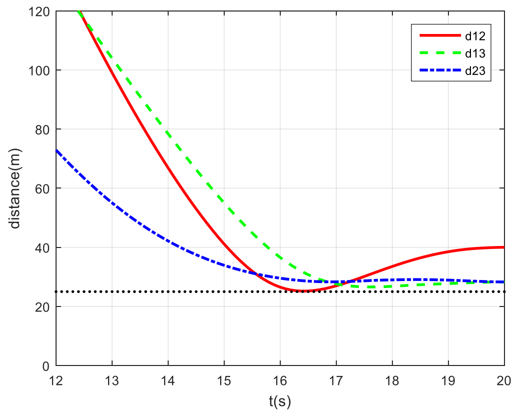

Figure 4. The distances between the quadrotors during flight with collision avoidance

In the iteration process, the performance index function value using distributed algorithm changes as shown in figure 5. The Nash optimal state is soon reached, with the performance index value being 11773.5. The first performance index value is lower because the iteration process takes the optimal control sequence of not considering collision avoidance constraints as the initial control sequence, which saves control energy needed for collision avoidance.

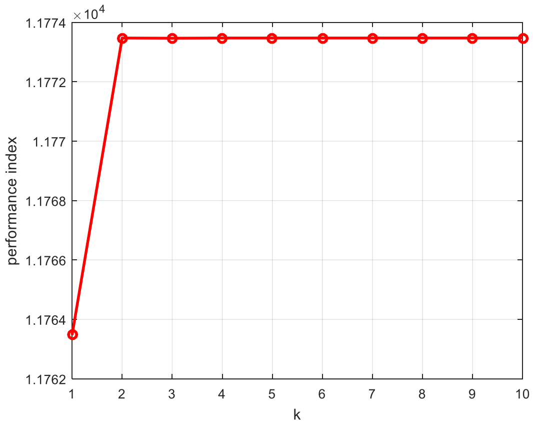

Figure 5. The performance index function value using distributed algorithm

Table 1 is the running time of centralized and distributed algorithm for three quadrotors, and table 2 is the running time of convex quadratic programming and Nonlinear programming respectively used in the centralized algorithm.

Table 1. Running time comparison

Solving Algorithm		Running Time (sec)
Centralized		6.532
Distributed	UAV1	2.676
	UAV2	2.357
	UAV3	2.793

Table 2. Running time comparison

Distributed Algorithm	Average Running Time (sec)
Convex quadratic programming	2.608
Nonlinear programming	4.174

It's obvious that when the formation scale is three, the distributed solution time is less than half of the centralized solution time. As the formation scale increases with the number of quadrotors, it is foreseeable that the computational burden of distributed algorithm will be greatly reduced compared with centralized algorithm. And from table 2, if the circular safety zone is adopted, the average solving time of nonlinear programming is 4.174s, which is much longer than that of convex quadratic programming. It shows us that using the half-plane safety zone to solve the collision avoidance is more efficient

5 Conclusions

This paper studied the optimal formation control with collision avoidance between quadrotors. Firstly, the nonlinear dynamic model of quadrotor is linearized. MPC is used to establish the prediction model of the optimization problem based on the linear dynamic model. Then, three kinds of constraints are analyzed, and among them, the safety zone constraints are transformed from the quadratic inequality of the circular zone into the linear inequality of the half-plane zone. A distributed optimization model of quadrotors formation control is presented. The model is decomposed into several local optimization models of single quadrotor, and we use the distributed receding horizon optimization to solve them. Finally, a formation configuration flight is simulated to validate that the algorithm can obtain Nash optimal formation trajectories under the condition of cooperative collision avoidance.

6. References

[1] Shen L, Niu Y and Zhu H 2013 Theories and methods of autonomous cooperative control for multiple UAVs. (Beijing: National Defense Industry Press)

[2] Liang X, Meng G and Xu Y 2018 A geometrical path planning method for unmanned aerial vehicle in 2D/3D complex environment. *Intelligent Service Robotics*, 11.3, pp 301-312

[3] Rathbun D, Kragelund S, Pongunwattana A and Capozzi B 2002 An evolution based path planning algorithm for autonomous motion of a UAV through uncertain environments. *AIAA 21st Digital Avionics System conference*, pp 551-607

[4] Pongpunwattana A 2004 Real-time planning for teams of autonomous vehicles in dynamic uncertain environments. (Washington: University of Washington)

[5] Zhu X, Yan M, Zhang C, Lin H and Qu Y 2017 UAV formation collision avoidance control method based on improved artificial potential field. *Journal of Harbin Engineering University*, 38.6, pp 961-968

[6] Richards A and How J P 2002 Aircraft trajectory planning with collision avoidance using mixed integer linear programming. *American Control Conference*, 3.3, pp 1936-1941

[7] Kamal W A, Gu D W and Postlethwaite I 2005 Real time trajectory planning for UAVs using MILP. *IEEE Conference on Decision and Control, and the European Control Conference*, pp 3381-3386

[8] Cowling I 2008 Towards autonomy of a quadrotor UAV. (Cranfield: Cranfield University)

[9] Gong J, Jiang Y and Xu W 2014 Model predictive control for self-driving vehicles. (Beijing: Beijing Institude of Technology Press)

[10] Nash 1950 The bargaining problem. Econometrica, 18.2, pp 155-162

7. Acknowledgment

This work was supported by the National Natural Science Foundation of China under Grant No.11532002.