

Received November 18, 2021, accepted December 8, 2021, date of publication December 20, 2021, date of current version January 6, 2022.

Digital Object Identifier 10.1109/ACCESS.2021.3136874

Fast Cooperative Trajectory Generation of Unmanned Aerial Vehicles Using a Bezier Curve-Based Shaping Method

ZE YU^{ID}, NAIMING QI^{ID}, MINGYING HUO^{ID}, ZICHEN FAN^{ID},
AND WEIRAN YAO^{ID}, (Member, IEEE)

Department of Aerospace Engineering, Harbin Institute of Technology, Harbin 150001, China

Corresponding author: Naiming Qi (qinmok@163.com)

This work was supported in part by the National Natural Science Foundation of China under Grant 11772104 and Grant 11702072, in part by the Fundamental Research Funds for Central Universities under Grant HIT.NSRIF.2017018, and in part by the China Postdoctoral Science Foundation under Grant 2019T120275.

ABSTRACT This study uses a Bezier curve-based shaping (BCBS) approach based on the dynamic model to quickly generate 3D cooperative trajectories for unmanned aerial vehicles (UAVs). It can facilitate the solution of the flight trajectory problem that occurs after the reallocation of mission points. Trajectory coordinates are expanded based on the Bezier curves that naturally satisfy boundary conditions (BCs); trajectories with the minimum flight time and those satisfying the dynamic constraints and collision-free conditions can be obtained by adjusting the Bezier coefficients. A three-UAV flight simulation is carried out to verify the effectiveness of the proposed method. Further, the performance of the proposed method is compared with that of the Gauss pseudospectral method (GPM). The simulation results of the BCBS method are used as the initial values for GPM optimization. It is demonstrated that the BCBS method requires a lower computation time than the direct solver, which is only 0.07% of the latter, and obtains similar optimization results (3.34% difference). This is considerably important for the rapid generation of optimized trajectories with the limited computing power of onboard computers. Furthermore, this method is expected to achieve online collaborative trajectory generation for multiple UAVs in view of its high computational efficiency.

INDEX TERMS Bezier-shaped functions, collision avoidance, trajectory optimization, multi-UAV cooperation.

I. INTRODUCTION

Unmanned aerial vehicles (UAVs) have been developed rapidly and used widely in recent decades [1]. Compared to manned combat units, UAVs provide the advantages of low cost and zero casualties. Therefore, UAVs are widely used in the military and civilian fields [2] such as for reconnaissance, surveillance, interference, relay communications, forest fire detection, and meteorological observation. With the increasing complexities of application scenarios, it is difficult for a single UAV to perform missions that can satisfy demands. Multiple UAVs have greater flexibility and adaptability and can perform better in some mission scenarios. Multi-UAV collaboration has become an important direction for UAV

technology, and in the future, it is expected that UAVs will be developed with a focus on clustering, autonomy, and high intelligence.

However, complex and variable multi-vehicle missions pose challenges for UAV technology. In an unexpected situation, the mission of UAVs may be disturbed or even changed. UAVs need to quickly generate flyable cooperative trajectories to the new mission points in emergencies. Therefore the completion of a complex multi-UAV cooperative mission requires a feasible UAV cooperative trajectory planning algorithm, which can generate safe and effective multi-UAV cooperative trajectories and provide UAVs a certain degree of autonomy. Further, because of the limited computing power of the onboard computer, the calculation amount of the UAV cooperative trajectory planning algorithm needs to be as small as possible. And the difficulty with multi-UAV cooperation is the need to improve the coordination of the UAV group

The associate editor coordinating the review of this manuscript and approving it for publication was Abderrahmane Lakas^{ID}.

such that it can avoid collisions between individuals under complex circumstances and ensure cooperative operation.

Trajectory planning algorithms generally use an optimal control theory. Popular numerical solutions of the optimal control theory include the direct (“discretise, then optimise”) and indirect methods (“optimise, then discretise”) [1], both of which need to be based on dynamics and involve a large amount of calculation (integral calculation). The pseudospectral optimal control theory is a popular strategy in the direct method, and it has been used to solve many aviation and aerospace problems [3]–[5]. In more recent developments, many improvements have been suggested for the pseudospectral method. Zhang *et al.* [6] developed an algorithm that integrates the Legendre pseudospectral method and the artificial potential field method. Their algorithm effectively solved the trajectory planning problem of the UAV in an obstacle-rich environment. Rogowski and Maroński [7] used the Chebyshev pseudospectral method to generate a trajectory for a glider in the vertical plane. Yang *et al.* [8] used Gauss pseudospectral method (GPM) in the trajectory optimization of a ramjet-powered vehicle. Sun *et al.* [9] optimized the 6-degree freedom trajectory of the parafoil delivery system with GPM. Other direct methods include the convex optimization method [10]. Pepy and Hérissé [11] proposed an algorithm based on an indirect shooting method.

The geometric method corresponds to the method using the optimal control theory. Babel [12] used shortest path algorithms for network optimization. They assumed that there are many line elements in space; connecting the directed line elements can act as the trajectory of the UAV. Other studies used Dubins paths, considering constraints such as obstacle avoidance, collision avoidance, and curvature limitation [13]–[16]. However, all UAVs move at a constant speed, and therefore, the optimization target is the path length. The Pythagorean hodograph (PH) method is another effective geometric method used for cooperative path planning [17]–[19]; dynamical constraints are described using geometric differential characteristics such as curvature and torsion. Although these purely geometric methods have high computational efficiency, they do not consider the dynamics of the vehicles. In the task planning for designing a large number of UAV trajectories, the most effective task allocation strategy can be quickly filtered using a purely geometric method; however, it may cause some trajectories to be less flyable.

The Bezier curve-based shaping (BCBS) approach presented in this paper employs a combination of geometry and dynamics, and it can, therefore, be called a geometric-dynamics method [20]. Inspired by the introduction of the concept of the shape function in Petropoulos and Longuski’s article [21], the Bezier approach was proposed in our previous paper [22]. Several works of literature [23], [24] employ Bezier curves for the path planning of robots. Bezier curves have been used to connect straight lines or circles to make the curvature of the UAV trajectories smoother and more continuous [25]–[27]; other similar methods include

B-spline [28]. Yu *et al.* [29] used the temporal–spatial Bezier curve to optimize the trajectory of multiple UAVs directly, and they obtained the suboptimal time for multiple vehicles to arrive simultaneously. Under the same conditions, the arrival time calculated using the temporal–spatial Bezier curve is shorter, and the curvature of the trajectory is more continuous compared with [30]. Although the temporal–spatial Bezier curve considers some dynamic constraints and collision avoidance, this method is a cubic Bezier curve. Under the condition that the boundary conditions (BCs) are determined, a cubic Bezier function can determine a unique curve without optimizing the Bezier coefficients.

The contributions of this study are as follows. First, this paper considers the dynamics model and constraints of the UAV with the collision avoidance of cooperative trajectories, instead of using the kinematics model in other papers [13]–[16] where the constraints on the turning radius and path length cannot consider the dynamic characteristics of the UAV. The method proposed in this paper takes thrust, angle of attack, and roll angle as control variables, which can be expressed by the curve coordinates and their derivatives obtained by the Bezier method. Second, the BCBS approach is adopted to design a multi-UAV cooperative trajectory quickly and efficiently. Assuming that the trajectory of the UAV is in the form of the Bezier function, the optimized trajectory can be obtained by optimizing the Bezier coefficients. The characteristics of the Bezier curve can satisfy the dynamic constraints and BCs of the UAV. A simulation scenario, wherein three UAVs depart and arrive simultaneously and may collide with each other, is designed to verify the feasibility of the proposed method. In the simulation, the Bezier method can quickly obtain the trajectory of the minimum flight time.

The remainder of this paper is organized as follows. Section 2 describes the dynamic model of the UAV, dynamic constraints, and collision-free conditions. Section 3 introduces a 3D cooperative trajectories generation framework using the Bezier approach, which solves the trajectory optimization problem using nonlinear programming problem (NLP) for Bezier coefficients. Section 4 describes a simulation that is performed in the set scenario to verify the feasibility of the method; the results are compared with those of GPM. Section 5 presents more simulated cases to test and verify the applicability and effectiveness of the method. Finally, some concluding remarks are described.

II. PROBLEM DESCRIPTION

A. MISSION FRAMEWORK OF MULTI-VEHICLE

The multi-UAV mission framework described in this paper is a centralized architecture. For a general multi-UAV mission framework, the ground station generates all the trajectories and sends them to each vehicle before and during the mission (Fig. 1). However, the mission environment is complex and prone to changes. Once an emergency occurs, the ground station may not have time to change trajectories as soon as possible due to the long distance between the ground

station and the UAV. UAVs need to have the ability to quickly generate trajectories autonomously. In the framework without a ground station(Fig. 2), the vehicles exchange and transmit information such as their current position and speed in real time. In the UAV group, any UAV can be used as the central vehicle to command and dispatch other UAVs. The central vehicle can be selected among the UAVs in the middle of the UAV group, which can facilitate the connection between the central unit and other vehicles. When there is an emergency that require the trajectories of the UAVs to be temporarily changed, the central vehicle quickly calculates a set of flyable flight trajectories based on the flight status of all the UAVs and transmits them to the extensions.

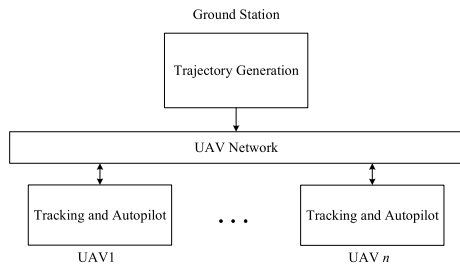


FIGURE 1. Multi-UAV mission framework with ground station.

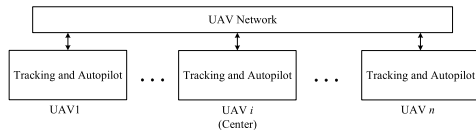


FIGURE 2. Multi-UAV mission framework without ground station.

B. DYNAMIC MODEL OF UNMANNED AERIAL VEHICLE (UAV)

A three-dimensional coordinate system needs to be determined to describe the position and motion of the vehicles conveniently. During missions, the rotation and curvature of the earth can be ignored because the flight distance and altitude are within a small range.

First, a three-dimensional inertial coordinate system is established in space, where the OX axis and OY axis are on the horizontal plane, and the OZ axis is perpendicular to the horizontal plane and points in the opposite direction of gravity. Then a speed coordinate system is established; the UAV speed vector coincides with the OX_v axis. The transformation relationship between the two coordinate systems can be expressed by the flight path angle γ and the heading angle ψ , which are the vertical and horizontal projection angles of OX and OX_v , respectively.

It is assumed that the mass of the vehicles does not change during flight, there is no sideslip, and the angle of attack is relatively small. The 6-DoF dynamic equations are

$$\begin{aligned} \dot{x} &= v \cos \gamma \cos \psi \\ \dot{y} &= v \cos \gamma \sin \psi \\ \dot{z} &= v \sin \gamma \end{aligned}$$

$$\begin{aligned} \dot{v} &= \frac{1}{M}(-D + F_T \cos \alpha - Mg \sin \gamma) \\ \dot{\gamma} &= \frac{1}{Mv}(L \cos \phi + F_T \sin \alpha \cos \phi - Mg \cos \gamma) \\ \dot{\psi} &= \frac{1}{Mv \cos \gamma}(L \sin \phi + F_T \sin \alpha \sin \phi) \end{aligned} \quad (1)$$

where $x, y,$ and z are the position coordinates of the vehicle in the inertial frame, and v is the speed. L and D are the lift and drag forces of the UAV, respectively.

$$\begin{aligned} L &= \frac{1}{2} \rho v^2 S C_L \\ C_L &= C_{L\alpha} \alpha \\ D &= \frac{1}{2} \rho v^2 S C_D \\ C_D &= C_{D0} + K C_L^2 \end{aligned} \quad (2)$$

S is the effective area, and C_L is the lift coefficient. In the range where the angle of attack is small, C_L can be regarded as proportional to the angle of attack α . Drag coefficient C_D includes the zero-lift drag coefficient and the part related to lift. K is a constant. In addition, $F_T, \alpha,$ and ϕ can be passed to the autopilot of the vehicle as control variables. $(x, y, z, v, \gamma, \psi)$ are regarded as state variables.

The flight path and heading angles can be expressed as

$$\begin{aligned} \gamma &= \arcsin\left(\frac{\dot{z}}{v}\right) \\ \psi &= \arctan\left(\frac{\dot{y}}{\dot{x}}\right) \end{aligned} \quad (3)$$

Further, the derivatives of γ and ψ are

$$\begin{aligned} \dot{\gamma} &= \frac{\ddot{z}v - \dot{z}\dot{v}}{v^2 \sqrt{1 - \dot{z}^2/v^2}} \\ \dot{\psi} &= \frac{\ddot{y}\dot{x} - \dot{y}\ddot{x}}{\dot{x}^2 + \dot{y}^2} \end{aligned} \quad (4)$$

where $\ddot{x}, \ddot{y},$ and \ddot{z} can be obtained using the variables of the Bezier method. The higher-order part of α can be ignored when the angle of attack is small. So it can be assumed that $\sin \alpha \approx \alpha$ and $\cos \alpha \approx 1$. Then Eq.(1) can be solved by the coordinates of the trajectories and their derivative.

$$\begin{aligned} F_T &= M\dot{v} + \frac{1}{2}v^2 S C_{D0} + Mg \sin \gamma \\ \phi &= \arctan\left(\frac{v \cos \gamma \dot{\psi}}{v \dot{\gamma} \cos \gamma}\right) \\ \alpha &= \frac{Mv\dot{\gamma} + Mg \cos \gamma}{\frac{1}{2} \cos \phi \rho v^2 S C_\alpha + F_T} \end{aligned} \quad (5)$$

where

$$\dot{v} = \frac{\dot{x}\ddot{x} + \dot{y}\ddot{y} + \dot{z}\ddot{z}}{\sqrt{\dot{x}^2 + \dot{y}^2 + \dot{z}^2}}$$

After obtaining the flight trajectories of the UAVs using the Bezier method, the control variables of the trajectories can be solved. The flyable trajectories must satisfy the dynamic constraints of the vehicles, such as speed, acceleration, and flight angles, moreover, the control input of the UAV also

has a certain range of restrictions. These constraints can be described as

$$\begin{aligned} v_{\min} \leq v \leq v_{\max}, \gamma_{\min} \leq \gamma \leq \gamma_{\max}, \\ a \leq a_{\max}, |\alpha| \leq \alpha_{\max}, |\phi| \leq \phi_{\max}, \\ F_{T \min} \leq F_T \leq F_{T \max} \end{aligned} \quad (6)$$

where $a = \sqrt{\ddot{x}^2 + \ddot{y}^2 + \ddot{z}^2}$. For multiple UAVs, each vehicle has corresponding constraints based on its capability.

C. COOPERATION CONSTRAINTS

As the number of vehicles in the mission increases, the cooperation among the vehicles need to be maintained to complete the mission, which includes time and space cooperation [31]. Some complex tasks require vehicles to reach the target points simultaneously or sequentially [32], [33]. This paper focuses on the study of multiple vehicles reaching the target points at the same time. Therefore, there are the following constraints on the flight time T_i of N UAVs.

$$T_i = T_j \quad i, j = 1, \dots, N, \quad i \neq j \quad (7)$$

Space cooperation requires to have a safe distance between the UAVs. If the vehicles cannot maintain a safe distance, it can affect flight control and cause collisions. The distance constraint can be satisfied by maintaining the distance between the vehicles to be greater than the minimum safe distance. The interval is calculated for the time where the mission times of two UAVs overlap. The vehicle position vector is \mathbf{p} , and the safe distance constraint between the i -th and j -th vehicles can be written as

$$\min_{\substack{i,j=1,\dots,N \\ i \neq j}} \|\mathbf{p}_i(t) - \mathbf{p}_j(t)\|_2^2 \geq E_d^2, \forall t \in [t_i, t_f] \quad (8)$$

where

$$t_i = \max\{t_{i,i}, t_{i,j}\}, t_f = \min\{t_{f,i}, t_{f,j}\},$$

$t_{i,i}$, and $t_{i,j}$ indicate the mission start time of the i -th and j -th vehicles, and correspondingly, $t_{f,i}$ and $t_{f,j}$ indicate the mission end times. Further, E_d denotes the minimum safety distance, which is far greater than the required minimum safety distance. It can avoid the insufficient safety distance caused by errors resulting from control, discrete curve, air flow, etc.

III. BEZIER CURVE-BASED SHAPING APPROACH

A. STATES APPROXIMATION

A vehicle describing the flight trajectory generated by the BCBS method is considered as an example. First, the dimensionless parameter $\tau = t/T \in [0, 1]$ of the flight time t is introduced, where T is the total flight time of the vehicle. Therefore, the 12 BCs of the trajectory generation problem can be written as

$$\begin{aligned} x(\tau = 0) = x_i, \quad x(\tau = 1) = x_f, \\ x'(\tau = 0) = T\dot{x}_i, \quad x'(\tau = 1) = T\dot{x}_f \\ y(\tau = 0) = y_i, \quad y(\tau = 1) = y_f, \end{aligned}$$

$$\begin{aligned} y'(\tau = 0) = T\dot{y}_i, \quad y'(\tau = 1) = T\dot{y}_f \\ z(\tau = 0) = z_i, \quad z(\tau = 1) = z_f, \\ z'(\tau = 0) = T\dot{z}_i, \quad z'(\tau = 1) = T\dot{z}_f \end{aligned} \quad (9)$$

where the normal subscripts “ i ” and “ f ” represent the initial and final states; the superscripts ‘ and ’ represent the derivative of the dimensionless time parameter τ and the flight time t , respectively.

The three-dimensional position coordinates of the vehicle flight trajectory can be expanded into the form of the Bezier curve function using the BCBS method [22].

$$\begin{aligned} x(\tau) &= \sum_{j=0}^{n_x} B_{x,j}(\tau)P_{x,j} \\ y(\tau) &= \sum_{j=0}^{n_y} B_{y,j}(\tau)P_{y,j} \\ z(\tau) &= \sum_{j=0}^{n_z} B_{z,j}(\tau)P_{z,j} \end{aligned} \quad (10)$$

where n_x , n_y , and n_z are the degrees of the Bezier function; $P_{x,j}$, $P_{y,j}$, and $P_{z,j}$ are the control coefficients of the Bezier curve; and $B_{x,j}$, $B_{y,j}$, and $B_{z,j}$ are Bezier basis functions [34], which can be written as follows (Examples are given in the x dimension to avoid repetition):

$$B_{x,j}(\tau) = \frac{n_x!}{j!(n_x - j)!} \tau^j (1 - \tau)^{n_x - j} \quad j \in [0, n_x] \quad (11)$$

From Eq.(10), the first derivative of x with respect to the time of flight t can be obtained as

$$x'(\tau) = \sum_{j=0}^{n_x} B'_{x,j}(\tau)P_{x,j} \quad (12)$$

where $B'_{x,j}(\tau)$ can be obtained from Eq.(11).

$$B'_{x,j}(\tau) = \begin{cases} -n_x(1 - \tau)^{n_x - 1} & j = 0 \\ \frac{n_x!}{(j - 1)!(n_x - j)!} \tau^{j-1} (1 - \tau)^{n_x - j} & j \in [1, n_x - 1] \\ -\frac{n_x!}{j!(n_x - j - 1)!} \tau^j (1 - \tau)^{n_x - j - 1} & j = n_x \\ n_x \tau^{n_x - 1} & j = n_x \end{cases} \quad (13)$$

Substituting $\tau = 0$ and $\tau = 1$ for Eq.(11) and Eq.(13), respectively, some parameters of the BCs can be obtained.

$$\begin{aligned} B_{x,j}(\tau = 0) &= \begin{cases} 1 & j = 0 \\ 0 & j \in [1, n_x] \end{cases} \\ B_{x,j}(\tau = 1) &= \begin{cases} 0 & j \in [0, n_x - 1] \\ 1 & j = n_x \end{cases} \\ B'_{x,j}(\tau = 0) &= \begin{cases} -n_x & j = 0 \\ n_x & j = 1 \\ 0 & j \in [2, n_x] \end{cases} \end{aligned}$$

$$B'_{x,j}(\tau = 1) = \begin{cases} 0 & j \in [0, n_x - 2] \\ -n_x & j = n_x - 1 \\ n_x & j = n_x \end{cases} \quad (14)$$

Combining Eq.(9), Eq.(10), and Eq.(14), we get

$$\begin{aligned} x_i &= x(\tau = 0) = P_{x,0} \\ x_f &= x(\tau = 1) = P_{x,n_x} \\ T\dot{x}_i &= x'(\tau = 0) = n_x(P_{x,1} - P_{x,0}) \\ T\dot{x}_f &= x'(\tau = 1) = n_x(P_{x,n_x} - P_{x,n_x-1}) \end{aligned} \quad (15)$$

Hence, the four Bezier coefficients can be parameterized as

$$\begin{aligned} P_{x,0} &= x_i \quad P_{x,1} = x_i + T\dot{x}_i/n_x \\ P_{x,n_x-1} &= x_f - T\dot{x}_f/n_x \quad P_{x,n_x} = x_f \end{aligned} \quad (16)$$

The Bezier curve can be adjusted by changing the remaining $n_x - 3$ Bezier coefficients in the optimization process. The roots of the m th-degree Legendre polynomial are used to discretize dimensionless time τ .

$$\tau_1 = 0 < \tau_2 < \dots < \tau_{m-1} < \tau_m = 1 \quad (17)$$

The coordinates $[x, y, z]$ of the vehicle flight trajectory are represented in the form of matrix products. For example, x is written as

$$\begin{aligned} [x]_{m \times 1} &= [B_x]_{m \times (n_x+1)} [P_x]_{(n_x+1) \times 1} \\ [x']_{m \times 1} &= [B'_x]_{m \times (n_x+1)} [P_x]_{(n_x+1) \times 1} \\ [x'']_{m \times 1} &= [B''_x]_{m \times (n_x+1)} [P_x]_{(n_x+1) \times 1} \end{aligned} \quad (18)$$

where $[P_x] = [P_{x,0} \quad P_{x,1} \quad [X_x]_{(n_x-3) \times 1}^T \quad P_{x,n_x-1} \quad P_{x,n_x}]^T$ are the Bezier coefficients. $P_{x,0}$, $P_{x,1}$, P_{x,n_x-1} , and P_{x,n_x} are the known coefficients, as described in Eq.(16), and $[X_x]_{(n_x-3) \times 1} = [P_{x,2} \quad \dots \quad P_{x,n_x-2}]^T$ are the unknown coefficients. Matrices $[B_x]_{m \times (n_x+1)}$, $[B'_x]_{m \times (n_x+1)}$ and $[B''_x]_{m \times (n_x+1)}$ can be obtained by substituting $\tau_j (j \in [1, m])$.

$$\begin{aligned} [B_x]_{m \times (n_x+1)} &= \begin{bmatrix} B_{x,0}(\tau_1) & \dots & B_{x,n_x}(\tau_1) \\ \vdots & \ddots & \vdots \\ B_{x,0}(\tau_m) & \dots & B_{x,n_x}(\tau_m) \end{bmatrix} \\ [B'_x]_{m \times (n_x+1)} &= \begin{bmatrix} B'_{x,0}(\tau_1) & \dots & B'_{x,n_x}(\tau_1) \\ \vdots & \ddots & \vdots \\ B'_{x,0}(\tau_m) & \dots & B'_{x,n_x}(\tau_m) \end{bmatrix} \\ [B''_x]_{m \times (n_x+1)} &= \begin{bmatrix} B''_{x,0}(\tau_1) & \dots & B''_{x,n_x}(\tau_1) \\ \vdots & \ddots & \vdots \\ B''_{x,0}(\tau_m) & \dots & B''_{x,n_x}(\tau_m) \end{bmatrix} \end{aligned} \quad (19)$$

Variables n_x , n_y , n_z , and m are required to be determined before beginning the trajectory generation calculation. The coefficient matrices $([B_x]_{m \times (n_x+1)})$, $([B'_x]_{m \times (n_x+1)})$, and $([B''_x]_{m \times (n_x+1)})$ are constant and are calculated once before starting the loop optimization calculation.

If $n_x = n_y = n_z = 3$, the shape of the flight trajectory is fixed by the BCs, and all the coefficients are determined. If $n_x > 3$, $n_y > 3$, and $n_z > 3$, there are unknown coefficients $([X_x]_{(n_x-3) \times 1}, [X_y]_{(n_y-3) \times 1}, [X_z]_{(n_z-3) \times 1})$ which need be optimized to satisfy the dynamic constraints.

From Eq.(5), $[F_T]_{m \times 1}$, $[\alpha]_{m \times 1}$, and $[\phi]_{m \times 1}$ are represented by the coordinates (in the form of Bezier curve functions) and their first and second derivatives by substituting Eqs.(3) and (4) into Eq.(1) and expanding the coordinates (x, y, z) using the BCBS method.

The above-mentioned descriptions of vehicle trajectory generation are based on a single vehicle. In this study, the time limitation problem of cooperative trajectory optimization is that N UAVs begin missions at the same time and reach their respective flight end points simultaneously. The optimization target is to generate the minimum flight time trajectories under the condition that all UAVs satisfy the dynamic constraints (Eq.(6)), and no collisions occur (Eq.(8)). Hence, the trajectory generation NLP problem of N vehicles can be described as

$$\begin{aligned} \min & \quad T \\ & [X_{xi}], [X_{yi}], [X_{zi}], T_i \\ \text{s.t.} & \quad T_i = T \\ & \quad v_{\min} \leq v_i \leq v_{\max}, \gamma_{\min} \leq \gamma_i \leq \gamma_{\max} \\ & \quad a_i \leq a_{\max}, |\alpha_i| \leq \alpha_{\max} \\ & \quad |\phi_i| \leq \phi_{\max}, F_{T \min} \leq F_{Ti} \leq F_{T \max} \\ & \quad \|\mathbf{p}_i(t) - \mathbf{p}_j(t)\|_2^2 \geq E_d^2 \\ & \quad i, j = 1, \dots, N, i \neq j \end{aligned} \quad (20)$$

where $[X_{xi}]$, $[X_{yi}]$, and $[X_{zi}]$ are the unknown Bezier coefficients of the i th vehicle. The constraints come from Eq.(6-8). For the free-time problem of trajectory generation, the number of variables that need be optimized is $N(n_x + n_y + n_z - 9) + N$, which includes the total flight time T_i . The Bezier coefficients can represent the entire trajectory, which is optimized by the Bezier method. In the process of information transmission between the vehicles, only the Bezier coefficients $(n_x + n_y + n_z)$ need to be transferred, instead of the points where the entire curve is discretized in a certain time scale, which greatly reduces the amount of data for information interaction between the vehicles. After the other vehicles receive the Bezier coefficients from the central vehicle, they can quickly calculate the point information at the next moment or the discrete point information of the entire flight curve according to the Bezier function.

B. INITIALIZATION OF UNKNOWN COEFFICIENTS

Before the optimization, initialize the variables ought to be optimized within a reasonable range. This paper discusses the free-time optimization problem. The approximated flight time T_{APP} can be estimated from the median flight speed of a randomly selected vehicle, and the maximum acceleration limit is required to be considered.

$$T_{APP} = \frac{S}{(v_{\min} + v_{\max})/2} \quad (21)$$

where S is the straight line length between the starting and final points.

The approximate coordinates (x_{APP} , y_{APP} , and z_{APP}) at the m Legendre-Gauss discretization points of each vehicle can be estimated by a third-order Bezier curve ($n_x = n_y = n_z = 3$).

$$\begin{aligned}
 x_{APP}(\tau) &= (1 - \tau)^3 P_{x,0} + 3\tau(1 - \tau)^2 P_{x,1} \\
 &\quad + 3\tau^2(1 - \tau) P_{x,2} + \tau^3 P_{x,3} \\
 y_{APP}(\tau) &= (1 - \tau)^3 P_{y,0} + 3\tau(1 - \tau)^2 P_{y,1} \\
 &\quad + 3\tau^2(1 - \tau) P_{y,2} + \tau^3 P_{y,3} \\
 z_{APP}(\tau) &= (1 - \tau)^3 P_{z,0} + 3\tau(1 - \tau)^2 P_{z,1} \\
 &\quad + 3\tau^2(1 - \tau) P_{z,2} + \tau^3 P_{z,3}
 \end{aligned} \tag{22}$$

Considering the BCs (Eq.(9)), the coefficients are as follow.

$$\begin{aligned}
 P_{x,0} &= x_i & P_{x,1} &= x_i + T_{APP}\dot{x}_i/3 \\
 P_{x,2} &= x_f - T_{APP}\dot{x}_f/3 & P_{x,3} &= x_f \\
 P_{y,0} &= y_i & P_{y,1} &= y_i + T_{APP}\dot{y}_i/3 \\
 P_{y,2} &= y_f - T_{APP}\dot{y}_f/3 & P_{y,3} &= y_f \\
 P_{z,0} &= z_i & P_{z,1} &= z_i + T_{APP}\dot{z}_i/3 \\
 P_{z,2} &= z_f - T_{APP}\dot{z}_f/3 & P_{z,3} &= z_f
 \end{aligned} \tag{23}$$

Hence, the discrete approximate coordinates ($[x_{APP}]_{m \times 1}$, $[y_{APP}]_{m \times 1}$, $[z_{APP}]_{m \times 1}$) can be obtained by substituting $[\tau]_{m \times 1}$ into Eq.(22). Then, the initial guess for the unknown coefficients in the Bezier curves can be obtained as

$$\begin{aligned}
 [P_{xAPP}]_{(n_x-3) \times 1} &= ([B_{xAPP}]_{m \times (n_x-3)})^{-1} [x_{APP}]_{m \times 1} \\
 [P_{yAPP}]_{(n_y-3) \times 1} &= ([B_{yAPP}]_{m \times (n_y-3)})^{-1} [y_{APP}]_{m \times 1} \\
 [P_{zAPP}]_{(n_z-3) \times 1} &= ([B_{zAPP}]_{m \times (n_z-3)})^{-1} [z_{APP}]_{m \times 1}
 \end{aligned} \tag{24}$$

IV. NUMERICAL SIMULATION RESULTS

A. CASE STUDY

The proposed method is a framework for the optimization of the cooperative trajectory of the UAVs. A 3D simulation flight mission is set up where three fixed-wing UAVs reach the patrol mission target points from their respective start points. The three UAVs are required to start the mission at the same time and reach the end points simultaneously as well. The goal is to determine the trajectory with the minimum flight time that satisfies the dynamics constraints. The simulation scenarios and some parameters refer to the data in Choe's paper [20]. Choe *et al.* used the PH Bezier curve to describe the UAV trajectories, and they expressed the timing law using the quadratic Bezier polynomials. Through the above method, all variables of the UAV are described in the form of the PH Bezier polynomials. However, Choe *et al.* only consider the kinematics model of the UAVs, and the adjustable range of the curves is relatively small. Our method uses the geometrical characteristics of the Bezier curves combined with dynamic equations to generate trajectories under dynamic constraints and boundary conditions. The control variables

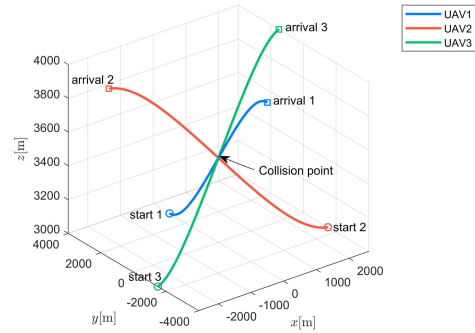


FIGURE 3. Flight trajectories without collision avoidance constraint.

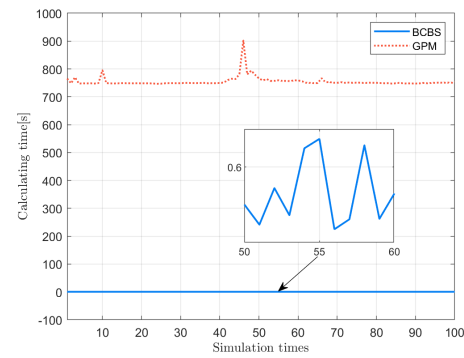


FIGURE 4. 100 Monte Carlo simulations.

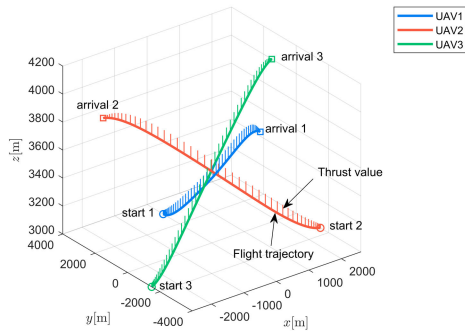
of the UAVs can be directly obtained through BCBS. The two methods have different applications to Bezier curves. The BCs and dynamic constraints are listed in Table 1 and 2. Some aerodynamic parameters are quoted from [10]. From the trajectory in Fig. 3, collisions will occur at the midpoints of the flight trajectories if the collision avoidance constraint is not considered in this simulation scenario.

In the following simulation, the orders of the Bezier curve function are $n_x = n_y = n_z = 6$, and the time parameter τ is discretized into 100 points. The simulation results of the BCBS method are used as the initial values of the GPM. In the trajectories generated by using GPM, the Legendre-Gauss points are set to 50. The NLP of both methods are solved using the sequence quadratic program method. All simulations in the paper are executed on a computer with the following specifications: 16.0 GB RAM, i7-10710U Intel processor at 1.10 GHz.

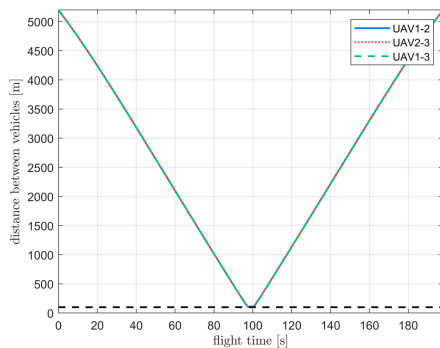
As shown in Fig. 4, in order to avoid the particularity of a single simulation, both methods have performed 100 Monte Carlo simulations. Fig. 5 shows the flight trajectories and space between vehicles calculated using the BCBS method. Further, Fig. 6 shows the results calculated using the GPM. The figure shows that the trajectories of both the methods have a collision avoidance flight tendency, and the distances between the vehicles are greater than the minimum safety distance. The resulting curve of the Bezier method is more detailed because the Bezier method has more discrete points

TABLE 1. Boundary conditions.

parameter	UAV 1	UAV 2	UAV 3
p_i (km)	(0,3.00,3.00)	(2.60,-1.50,3.00)	(-2.60,-1.50,3.00)
v_i (m/s)	25	25	25
γ_i ($^\circ$)	0	0	0
ψ_i ($^\circ$)	-90	150	30
p_f (km)	(0,-3.00,4.00)	(-2.60,1.50,4.00)	(2.60,1.50,4.00)
v_f (m/s)	25	25	25
γ_f ($^\circ$)	0	0	0
ψ_f ($^\circ$)	-90	150	30



(a) Flight trajectories with the Bezier method.



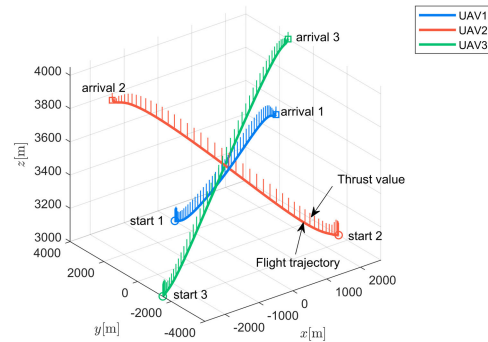
(b) Distance between vehicles with the Bezier method.

FIGURE 5. 3D trajectories of 3 UAVs with the Bezier method.

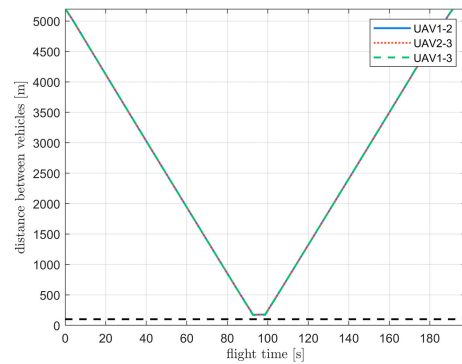
TABLE 2. Dynamic constraints.

parameter	value
M (kg)	2.6
S (m ²)	0.2589
ρ (kg/m ³)	1.225
g (m/s ²)	9.81
C_{D0}	0.01631
K	0.04525
C_α	4.2
v_{max} (m/s)	32
v_{min} (m/s)	18
a_{max} (m/s ²)	10
γ_{min} ($^\circ$)	-20
γ_{max} ($^\circ$)	30
T_{max} (N)	10
T_{min} (N)	0
α_{max} ($^\circ$)	10
ϕ_{max} ($^\circ$)	24

than GPM. If GPM uses more discrete points, the calculation time will increase significantly. The flight time obtained by



(a) Flight trajectories with GPM.



(b) Distance between vehicles with GPM.

FIGURE 6. 3D trajectories of 3 UAVs with GPM.

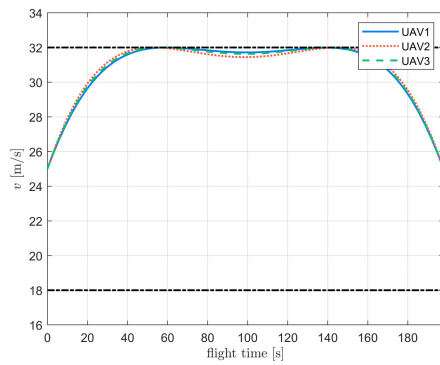
using the Bezier method is 197.8359 s, and that obtained using GPM is 191.444 s. The calculation time spent by the Bezier method and GPM is 0.588 s and 754.7936 s, respectively. The difference between the results calculated by the two methods is about 3.34%; however, the calculation time of the Bezier method is only 0.078% of that used to optimize and generate the trajectories using GPM further.

Figs. 7 and 8 show the dynamics variables of the vehicles with the Bezier method and the GPM, respectively. The figures indicate that the results of using the Bezier method and GPM can satisfy dynamics constraints. However, the results using the Bezier method are smoother, and there are no mutations, which is more favorable for the UAV's flight control.

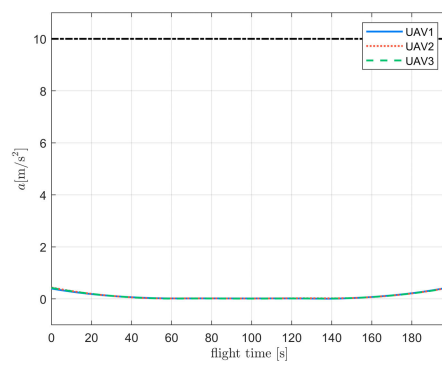
To avoid accidental results in the simulation, some parameters of the simulation cases in Section 4 and performed

TABLE 3. Comparison of results and calculation time.

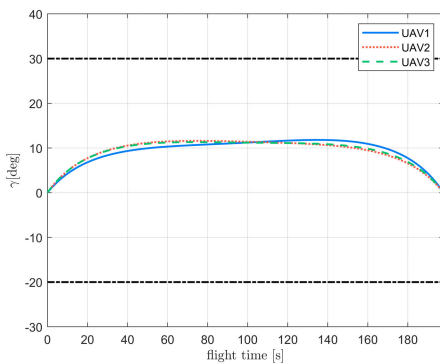
case	changed parameters	value	flight time		$\frac{\Delta}{GPM}$	calculation time		$\frac{Bezier}{GPM}$
			Bezier	GPM		Bezier	GPM	
1	original case in Section 4	0	197.8359	191.444	3.34%	0.548	747.857	0.07%
2	multiple of total flight distance	0.5	150.1409	146.894	2.21%	0.234	276.454	0.08%
3		1.5	242.2994	234.474	3.34%	0.609	737.891	0.08%
4		2	279.7650	270.006	3.61%	0.759	797.092	0.10%
5		$F_{T\max}(N)$	6	246.2240	236.309	4.20%	0.432	974.221
6	$v_i(m/s)$	18	211.8294	200.381	5.71%	0.323	1165.951	0.03%
7		22	201.0632	195.375	2.91%	0.272	1221.367	0.02%
8		28	196.2861	192.183	2.14%	0.729	1057.931	0.07%
9		32	233.2417	232.118	0.48%	0.275	106.748	0.26%
10	$v_{\max}(m/s)$	28	222.1560	218.526	1.66%	0.880	628.799	0.14%
11		30	208.9886	203.710	2.59%	0.290	1419.212	0.02%
12		34	178.4852	171.514	4.06%	0.409	985.204	0.04%
13		40	160.7717	154.785	3.87%	0.276	1356.190	0.02%



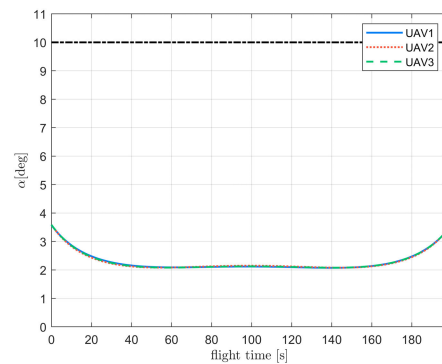
(a) Velocity of vehicles.



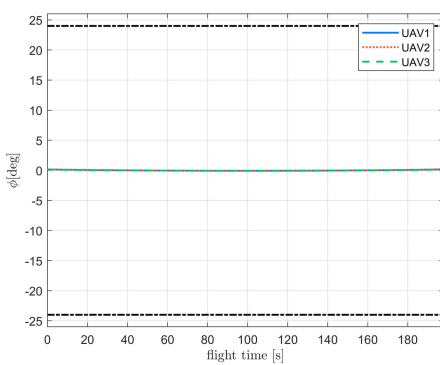
(b) Acceleration of vehicles



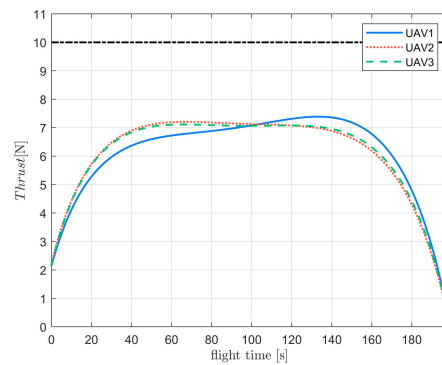
(c) Flight path angle with Bezier method



(d) The angle of attack



(e) Roll angle



(f) Thrust

FIGURE 7. Results of dynamics variables with Bezier method.

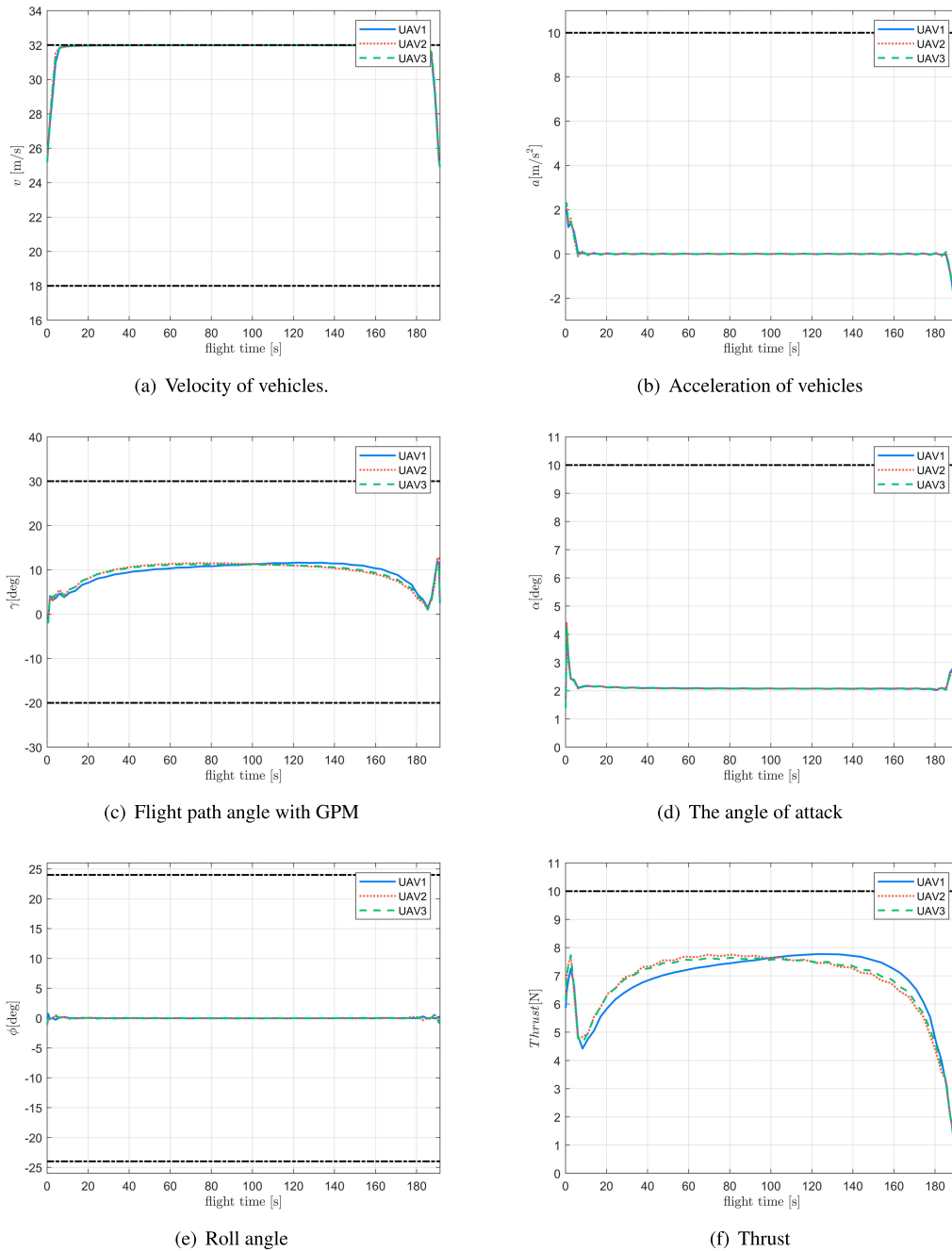


FIGURE 8. Results of dynamics variables with GPM.

more simulations have been changed. The simulation results and calculation times are summarized in Table. 3. Case 1 is the original case in Section 4. Case 2–4 changed multiple coordinates and the total flight distance; Case 5 changed the maximum thrust ($F_{T\max}$); Case 6–9 changed the initial speed (v_i); and Cases 10–13 changed the maximum speed (v_{\max}). $\frac{\text{Bezier}}{\text{GPM}}$ in the table represents the percentage of calculation time obtained by the Bezier curve relative to the calculation

time obtained by GPM. $\frac{\Delta}{\text{GPM}}$ represents the difference of the flight time optimized by the two methods.

In the simulation of the above cases, the Bezier method required less calculation time (on an average, 0.08% of that GPM takes) and received relatively satisfactory results. The average difference of the calculated total flight time was only 3.09%. Further, the GPM uses the calculation result of the Bezier method as the initial value for further optimization.

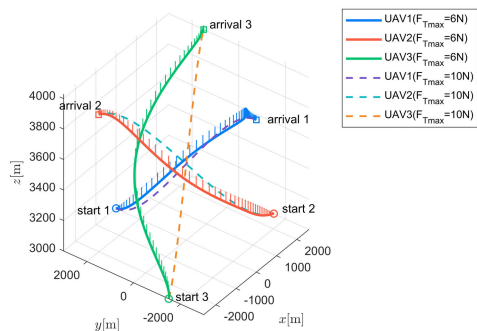


FIGURE 9. Trajectories of Case 5.

TABLE 4. Influence of the orders of Bezier function on the results.

orders of the Bezier function	calculation time (s)	flight time (s)
6	0.5478	197.8359
10	3.8892	193.2861
16	11.0066	192.4758
20	20.5503	191.9353

The calculated results using both methods can satisfy the dynamic constraints. Moreover, the curves calculated by the Bezier method were continuous and smooth because more discrete points were used. Furthermore, a more continuous trajectory was found to be beneficial for avoiding collisions and helpful for flight control. If the GPM has more discrete points, the calculation time will be much longer. The small amount of calculation for the Bezier method was attributed to the coefficient matrix requiring one single calculation before optimization; the result was then substituted into the optimization iteration. Another reason was the geometry of the Bezier method, which does not require a large number of integration operations. In the optimization process, adjusting the Bezier coefficient can make the trajectory meet the requirements. When calculating the coordinated trajectories of more UAVs, the difference in the amount of calculation will increase more times.

In particular, the simulation case 5 changes the maximum thrust limit, which directly reflects the dynamic constraints. The trajectories are shown in Fig. 9, where it can be seen that the UAVs can only reach the target points by circling far because of insufficient thrust. The trajectory simulation of the kinematics model cannot realize the limitation of the thrust and angle of attack, which can only be indirectly restricted by the maximum flight path angle and the turning radius reflected by the dynamics model.

B. INFLUENCE OF BEZIER ORDER

Simulations on the influence of the order of the Bezier function on the calculation results and calculation time have been carried out. In the simulation, only the order is changed, and the other conditions are not changed. The results are shown in Table. 4. Higher-order Bezier functions can obtain better results, however, the calculation time is also significantly increased higher. The difference between the simulation

results of the 6th-order and 20th-order functions is only 3.07%. Higher-order Bezier functions have higher degrees of freedom and can fine-tune the flight trajectory curve. But there are also more coefficients involved in the optimization process, which increases the amount of calculation. In order to improve calculation efficiency, a smaller order needs to be selected. However, if the order is too low, the error of the optimization result will be larger. The selection of the order of the Bezier function requires a balance between the accuracy of the calculation result and the calculation efficiency.

V. CONCLUSION

The rapid generation of 3D cooperative trajectories of multiple UAVs using the proposed BCBS method was presented in this paper. The BCBS method had both the authenticity of dynamics and the quickness of calculation. The trajectory coordinates were expanded into the form of the Bezier function, and the desired trajectories were optimized by adjusting the Bezier coefficient. A simulation was performed, and the results were compared with those of the GPM to verify the feasibility of the method. The Bezier method required less calculation time (0.07%) to generate the cooperative trajectories with the minimum flight time that satisfies the dynamics constraints and avoids collisions. Furthermore, the optimization results were found to be satisfactory, with a difference was only 3.34%. The trajectories obtained by the Bezier method were more continuous and smoother than those obtained using GPM, which is beneficial for flight control and collision avoidance. The relatively small amount of computations required by the Bezier method is of considerable importance for the rapid generation of optimized trajectories with onboard computers that have limited computing power. Therefore, it is expected that the proposed method will be employed for real-time online trajectory generation.

REFERENCES

- [1] W. P. Coutinho, M. Battarra, and J. Fliege, "The unmanned aerial vehicle routing and trajectory optimisation problem, a taxonomic review," *Comput. Ind. Eng.*, vol. 120, pp. 116–128, Jun. 2018.
- [2] G. Afifi and Y. Gadallah, "Autonomous 3-D UAV localization using cellular networks: Deep supervised learning versus reinforcement learning approaches," *IEEE Access*, vol. 9, pp. 155234–155248, 2021.
- [3] K. Bollino, L. R. Lewis, P. Sekhavat, and I. M. Ross, "Pseudospectral optimal control: A clear road for autonomous intelligent path planning," in *Proc. AIAA Infotech Aerospace Conf. Exhib.*, May 2007, p. 2831.
- [4] Q. Gong, L. R. Lewis, and I. M. Ross, "Pseudospectral motion planning for autonomous vehicles," *J. Guid., Control, Dyn.*, vol. 32, no. 3, pp. 1039–1045, May 2009.
- [5] K. Bollino and L. R. Lewis, "Collision-free multi-UAV optimal path planning and cooperative control for tactical applications," in *Proc. AIAA Guid., Navigat. Control Conf. Exhibit*, Aug. 2008, p. 7134.
- [6] S. Zhang, J. Yu, Y. Mei, H. Sun, and Y. Du, "Unmanned aerial vehicle trajectory planning by an integrated algorithm in a complex obstacle environment," *Proc. Inst. Mech. Eng., G, J. Aerosp. Eng.*, vol. 231, no. 11, pp. 2048–2067, Sep. 2017.
- [7] K. Rogowski and R. Maroński, "Optimization of glider's trajectory for given thermal conditions," *Arch. Mech. Eng.*, vol. 58, no. 1, pp. 11–25, Jan. 2011.
- [8] S. Yang, T. Cui, X. Hao, and D. Yu, "Trajectory optimization for a ramjet-powered vehicle in ascent phase via the Gauss pseudospectral method," *Aerosp. Sci. Technol.*, vol. 67, pp. 88–95, Aug. 2017.

- [9] H. Sun, S. Luo, Q. Sun, Z. Chen, W. Wu, and J. Tao, "Trajectory optimization for parafoil delivery system considering complicated dynamic constraints in high-order model," *Aerosp. Sci. Technol.*, vol. 98, Mar. 2020, Art. no. 105631.
- [10] Z. Wang and S. T. McDonald, "Convex relaxation for optimal rendezvous of unmanned aerial and ground vehicles," *Aerosp. Sci. Technol.*, vol. 99, Apr. 2020, Art. no. 105756.
- [11] R. Pepy and B. Hérisse, "An indirect method for optimal guidance of a glider," *IFAC Proc. Vol.*, vol. 47, no. 3, pp. 5097–5102, 2014.
- [12] L. Babel, "Trajectory planning for unmanned aerial vehicles: A network optimization approach," *Math. Methods Oper. Res.*, vol. 74, no. 3, pp. 343–360, 2011.
- [13] W. Yao, N. Qi, J. Zhao, and N. Wan, "Bounded curvature path planning with expected length for Dubins vehicle entering target manifold," *Robot. Auton. Syst.*, vol. 97, pp. 217–229, Nov. 2017.
- [14] W. Yao, N. Qi, C. Yue, and N. Wan, "Curvature-bounded lengthening and shortening for restricted vehicle path planning," *IEEE Trans. Autom. Sci. Eng.*, vol. 17, no. 1, pp. 15–28, Jan. 2020.
- [15] H. Zhang, L. Dou, B. Xin, J. Chen, M. Gan, and Y. Ding, "Data collection task planning of a fixed-wing unmanned aerial vehicle in forest fire monitoring," *IEEE Access*, vol. 9, pp. 109847–109864, 2021.
- [16] M. Shanmugavel, A. Tsourdos, B. White, and R. Żbikowski, "Cooperative path planning of multiple UAVs using Dubins paths with clothoid arcs," *Control Eng. Pract.*, vol. 18, no. 9, pp. 1084–1092, Sep. 2010.
- [17] M. Shanmugavel, A. Tsourdos, R. Żbikowski, B. A. White, C. A. Rabbath, and N. Lechevin, "A solution to simultaneous arrival of multiple UAVs using Pythagorean hodograph curves," in *Proc. Amer. Control Conf.*, Jun. 2006, p. 6.
- [18] S. Subchan, B. A. White, A. Tsourdos, M. Shanmugavel, and R. Żbikowski, "Pythagorean hodograph (PH) path planning for tracking airborne contaminant using sensor swarm," in *Proc. IEEE Instrum. Meas. Technol. Conf.*, May 2008, pp. 501–506.
- [19] M. A. Shah and N. Aouf, "3D cooperative Pythagorean hodograph path planning and obstacle avoidance for multiple UAVs," in *Proc. IEEE 9th Int. Conf. Cybernetic Intell. Syst.*, Sep. 2010, pp. 1–6.
- [20] R. Choe, J. Puig-Navarro, V. Cichella, E. Xargay, and N. Hovakimyan, "Cooperative trajectory generation using Pythagorean hodograph Bézier curves," *J. Guid. Control Dynam.*, vol. 39, no. 8, pp. 1744–1763, Aug. 2016.
- [21] A. E. Petropoulos and J. M. Longuski, "Shape-based algorithm for the automated design of low-thrust, gravity assist trajectories," *J. Spacecraft Rockets*, vol. 41, no. 5, pp. 787–796, Sep. 2004.
- [22] M. Huo, Z. Yu, H. Liu, C. Zhao, T. Lin, Z. Song, and N. Qi, "Initial three-dimensional trajectory design for solar sails using Bezier shaping approach," *IEEE Access*, vol. 7, pp. 150842–150850, 2019.
- [23] K. G. Jolly, R. S. Kumar, and R. Vijayakumar, "A Bezier curve based path planning in a multi-agent robot soccer system without violating the acceleration limits," *Robot. Auto. Syst.*, vol. 57, no. 1, pp. 23–33, Jan. 2009.
- [24] C. Chen, Y. He, C. Bu, J. Han, and X. Zhang, "Quartic Bezier curve based trajectory generation for autonomous vehicles with curvature and velocity constraints," in *Proc. IEEE Int. Conf. Robot. Automat. (ICRA)*, Hong Kong, May 2014, pp. 6108–6113.
- [25] A. Machmudah, S. Parman, and A. Zainuddin, "UAV Bezier curve maneuver planning using genetic algorithm," in *Proc. 12th Annu. Conf. Comp Genetic Evol. Comput. (GECCO)*, New York, NY, USA, Jul. 2010, pp. 2019–2022.
- [26] Z. Habib and M. Sakai, "G2 cubic transition between two circles with shape control," *J. Comput. Appl. Math.*, vol. 223, no. 1, pp. 133–144, Jan. 2009.
- [27] X. Wang, P. Jiang, D. Li, and T. Sun, "Curvature continuous and bounded path planning for fixed-wing UAVs," *Sensors*, vol. 17, no. 9, p. 2155, Sep. 2017.
- [28] J. Connors and G. Elkaim, "Analysis of a spline based, obstacle avoiding path planning algorithm," in *Proc. IEEE 65th Veh. Technol. Conf.*, Apr. 2007, pp. 2565–2569.
- [29] W. Yu, W. Shuo, W. Rui, and T. Min, "Generation of temporal-spatial Bezier curve for simultaneous arrival of multiple unmanned vehicles," *Inf. Sci.*, vol. s. 418–419, pp. 34–45, Dec. 2017.
- [30] M. Shanmugavel, A. Tsourdos, R. Żbikowski, and B. White, "Path planning of multiple UAVs using Dubins sets," in *Proc. AIAA Guid. Navigation Control Conf. Exhib.*, 2005, p. 5827.
- [31] H. Zhang, B. Xin, L.-H. Dou, J. Chen, and K. Hirota, "A review of cooperative path planning of an unmanned aerial vehicle group," *Frontiers Inf. Technol. Electron. Eng.*, vol. 21, no. 12, pp. 1671–1694, Dec. 2020.
- [32] X. M. Cheng, D. Cao, and C. T. Li, "Survey of cooperative path planning for multiple unmanned aerial vehicles," *Appl. Mech. Mater.*, vol. s. 668–669, pp. 388–393, Oct. 2014.
- [33] Y. Liu, X. J. Zhang, Y. Zhang, and X. M. Guan, "Collision free 4D path planning for multiple UAVs based on spatial refined voting mechanism and PSO approach," *Chin. J. Aeronaut.*, vol. 32, pp. 1504–1519, Jun. 2019.
- [34] G. Farin, *Curves and Surfaces for Computer-Aided Geometric Design, A Practical Guide*. Amsterdam, The Netherlands: Elsevier, Jan. 1997.



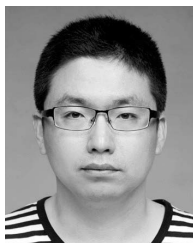
ZE YU received the B.S. degree in aeronautical and astronautical science and technology from the Harbin Institute of Technology, Harbin, China, in 2018. He is currently pursuing the Ph.D. degree with the Department of Aerospace Engineering.

His research interest includes cooperative trajectory optimization of multiple UAVs.



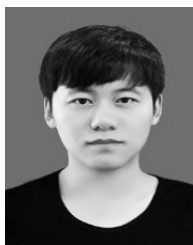
NAIMING QI received the Ph.D. degree in precision instrument and machinery from the Harbin Institute of Technology, Harbin, China, in 2001.

He is currently a Professor with the Department of Aerospace Engineering, Harbin Institute of Technology. His research interests mainly include dynamics and control of spacecraft and trajectory optimization method.



MINGYING HUO received the Ph.D. degree in aeronautical and astronautical science and technology from the Harbin Institute of Technology, Harbin, China, in 2015.

He is currently an Associate Professor with the Department of Aerospace Engineering, Harbin Institute of Technology. His research interests mainly include dynamics and control of electric solar wind sail and shape-based trajectory optimization method.



ZICHEN FAN received the B.S. degree in aeronautical and astronautical science and technology from the Harbin Institute of Technology, Harbin, China, in 2017, where he is currently pursuing the Ph.D. degree in aerospace engineering.

His research interests mainly include dynamics and control of spacecraft, and trajectory optimization method.



WEIRAN YAO (Member, IEEE) received the Ph.D. degree in aeronautical and astronautical science and technology from the Harbin Institute of Technology, Harbin, China, in 2020.

From 2017 to 2018, he was a Visiting Ph.D. Student with the Department of Mechanical and Industrial Engineering, University of Toronto, Toronto, ON, Canada. He is currently a Lecturer with the Department of Control Science and Engineering, Harbin Institute of Technology. His

research interests include path planning for holonomic and nonholonomic robots, multi-robot mission planning, and multiagent control systems.

...