Information Sciences 418–419 (2017) 34–45

Contents lists available at ScienceDirect

Information Sciences

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

Generation of temporal–spatial Bezier curve for simultaneous arrival of multiple unmanned vehicles

Wang Yu [a], Wang Shuo [a,b,*], Wang Rui [a,b], Tan Min [a,b]

[a] State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China

[b] University of Chinese Academy of Sciences, Beijing 100049, China

ARTICLE INFO

Article history:
Received 5 September 2016
Revised 18 July 2017
Accepted 26 July 2017
Available online 29 July 2017

Keywords:
Simultaneous arrival
Bezier curve
Path planning

ABSTRACT

This paper investigates simultaneous arrival planning of multiple unmanned vehicles. Problem of simultaneous arrival planning is introduced and formulated. First, a Temporal–Spatial (T–S) Bezier curve is constructed by considering time as a one-dimensional variable of Bezier curve. Second, constraints of minimum curvature radius, tangent acceleration and velocity are addressed to design the suboptimal T–S Bezier curve. Subsequently, the suboptimal simultaneous arrival time for multiple unmanned vehicles is calculated and obtained. Finally, simultaneous arrival planning by using a T–S Bezier is summarized. Simulations and comparisons are conducted to demonstrate the effectiveness of the proposed method.

1. Introduction

Autonomy corresponds to the basic capacity of unmanned vehicles. Thus, autonomous systems and algorithms are rapidly developed to substitute for human beings to accomplish different types of tasks including rescue [22], reconnaissance and surveillance [26], search [10], and strike [14]. In most tasks, it is necessary for unmanned vehicles to sail from one point to another point. This requires path planning to generate a feasible trajectory that connects the two points.

Recently, several numerical methods were employed for path planning. Mellinger et al. employed mixed-integer quadratic programs to generate optimal paths for air vehicles by considering collision avoidance [15]. Portasa et al.provided the performance comparisons of genetic algorithms, PSO algorithms, and differential evolution for UAV path planning [18]. Analogously, Roberge et al. listed the comparisons and performance analysis of genetic algorithm and particle swarm optimization algorithm, which were both devoted to producing feasible and quasi-optimal trajectories for fixed wing UAVs in a complex 3D environment while considering the dynamic properties of a vehicle [20]. Ma et al.proposed improved PSO-based methods Con-Per-PSO and SA-PSO for path planning of multiple mobile robots under the time-varying double-warehouse environment [16]. Rashid et al. presented a visibility binary tree algorithm to generate the shortest path between the initial point and the final point by considering circular obstacles [19]. Kothari et al. applied an exploring random tree algorithm to design probabilistically robust path planning in uncertain environments with dynamic obstacles [9]. Moreover, A^* algorithm was applied to search for the optimal path. Lu et al. proposed an extension of the Lifelong Planning A^* algorithm by using

* Corresponding author at: State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China.
E-mail address: shuo.wang@ia.ac.cn (W. Shuo).

http://dx.doi.org/10.1016/j.ins.2017.07.031
0020-0255/

a multiscale decomposition of the environment, which solved dynamic shortest path-planning problems [12]. Analogously, MacAllister et al. utilized an anytime planner based on A^* algorithm to get close-to-optimal paths for a quadrotor [13]. Additionally, another method involves using smooth curves to achieve path planning. Lekkas et al. designed a path planner based on a monotone cubic Hermite spline to realize better shape control and to avoid wiggles and zigzags [11]. Bianco et al. utilized η^3-splines to accomplish path planning for wheeled mobile robots [1]. Ghilardelli et al. used a η^4-spline to shape a smooth feasible curve between two arbitrary dynamic configurations of the articulated vehicle [6]. Brezak et al. employed a clothoids curve, which corresponded to a G^2 continuous path with linearly changing curvature, to smooth piecewise linear paths [2]. Our group proposed a Real-time Dynamic Dubins-Helix (RDDH) method to achieve smooth transition between the initial pose and the final pose [23], and an autonomous approach to a moving ship for unmanned vehicles based on the RDDH method was realized [24]. Furthermore, a type of curves with a closed-form expression of position such as B-splines [5,8] and Bezier curves [3,25] were applied for path planning.

This article focuses on the simultaneous arrival of multiple unmanned vehicles by path planning. It is necessary to satisfy the physical constraints of unmanned vehicles to guarantee the feasibility for simultaneous arrival of multiple unmanned vehicles. The acceleration constraint that is decomposed into centripetal acceleration and tangential acceleration should be addressed. Centripetal acceleration constraint is closely related to curvature radius, and thus, the minimum curvature radius constraint instead of centripetal acceleration constraint is usually taken into account. Furthermore, it is necessary to satisfy the vehicle's velocity constraint.

Several types of feasible curves were used to achieve the simultaneous arrival of vehicles. Dai et al. devised parameterized Cornu-Spirals for a group of vehicles to implement simultaneous arrival [4]. Shanmugavel et al. used Dubins curves to achieve the simultaneous arrival of multiple unmanned vehicles [21]. Oh et al. presented path shaping with two constant curvature segments to achieve simultaneous arrival, and this provided a path with longer length albeit increased flexibility and fewer discontinuity points on a curvature command when compared with the widely-used Dubins path [17]. In the above methods, the main idea for simultaneous arrival is to generate paths with the same length based on the assumption that the speed of the vehicles is the same.

The main contribution of this paper is to design Temporal-Spatial (T–S) Bezier curve for simultaneous arrival planning of multiple unmanned vehicles. The T–S Bezier curve is constructed by considering time as a one-dimensional variable of the Bezier curve. This novel design allows a vehicle to spend an assigned time sailing from the initial pose to the final one. Another notable advantage of T–S Bezier curve is its smoothness in terms of curvature, heading angle, and velocity. Thus, it is easy especially for fixed-wing UAVs to track the T–S Bezier. In order to obtain the suboptimal T–S Bezier curve (in terms of path length), a genetic algorithm is employed by considering an unmanned vehicle's physical constraints of minimum curvature radius, tangent acceleration and velocity. Moreover, the suboptimal simultaneous arrival time for multiple unmanned vehicles is calculated. Simultaneous arrival planning is realized by planning T–S Bezier curves for each vehicle once the suboptimal simultaneous arrival time is set. Comparisons are given to illustrate the feasibility of T–S Bezier curve for simultaneous arrivals.

The organization of the rest of the paper is given as follows. The problem of simultaneous arrival planning is stated and the proposed solution is described in Section 2. Section 3 details the T–S Bezier curve for simultaneous arrival planning. The design and generation of T–S Bezier curve is presented in Section 3.1. The suboptimal T–S Bezier curve that considers physical constraints and the suboptimal arrival time are proposed and obtained in Section 3.2 and Section 3.3, respectively. Simultaneous arrival planning is subsequently summarized in Section 3.4. Simulations and comparisons of simultaneous arrival are conducted in Section 4. The conclusion is given in Section 5.

2. Problem statement

The problem schematic is shown in Fig. 1. N unmanned vehicles sail from their individual initial poses to the final poses. The initial and final poses are assumed to be known a priori. It is necessary to implement the simultaneous arrival of multiple vehicles.

In order to achieve the simultaneous arrival of multiple vehicles, path generation is implemented. In a Cartesian coordinate frame, the trajectory of an unmanned vehicle is generated based on the following equations:

$$\begin{cases} \dot{x} = v\cos\theta \\ \dot{y} = v\sin\theta \end{cases} \tag{1}$$

where v denotes the vehicle speed, θ denotes the heading angle. The feasible trajectory of the ith vehicle that connects its initial pose to its final pose is given by path planning. This trajectory is represented by C_i in which the length is represented by s_i. Hence, the problem formulation drawn from [17] is given as follows:

$$F_{s,i}(x_{s,i}, y_{s,i}, \theta_{s,i}, v_{s,i}) \xrightarrow{C_i} F_{g,i}(x_{g,i}, y_{g,i}, \theta_{g,i}, v_{g,i}) \tag{2}$$

where $F_{s,i}$ denotes the initial configuration of the ith vehicle, $F_{g,i}$ denotes the final configuration of the ith vehicle, $(x_{s,i}, y_{s,i})$, and $(x_{g,i}, y_{g,i})$ represent the initial position and final position of the ith vehicle, respectively, and $\theta_{s,i}$ and $\theta_{g,i}$ represent the initial heading and final heading angles of the ith vehicle, respectively. $v_{s,i}$ and $v_{g,i}$ represent the initial and final speeds of the ith vehicle, respectively. It is assumed that all the vehicles correspond to the same type, thus the values of physical constraints are fixed. Each vehicle trajectory C_i should satisfy the correspond to following physical constraints.

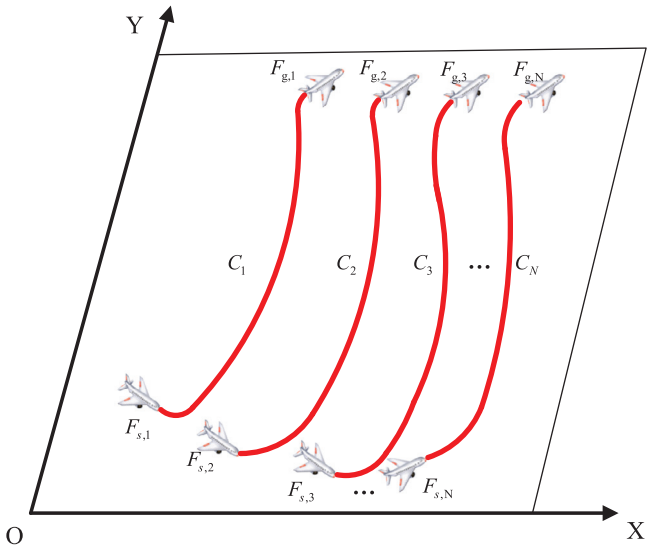

Fig. 1. Problem formulation.

(1) Minimum curvature radius constraint. The curvature radius constraint is taken into account as follows:

$$R_i(s_{a,i}) \geq R_{min} \tag{3}$$

where $s_{a,i}$ denotes the curvilinear abscissa of the ith vehicle trajectory, R_i denotes the curvature radius of the ith vehicle trajectory, and $R_{\min}$ denotes the minimum curvature radius. This constraint is closely related to the centripetal acceleration constraint along the path. Evidently, the trajectory planned for the vehicle should be smooth, and the radius along the entire path should be continuous. Big tracking errors exist when discontinuities occur [17].

(2) Tangent acceleration constraint. The tangent acceleration at any position along the path must be smaller than the vehicle's maximal tangent acceleration, which is expressed as follows:

$$|a_{i,t}(t)| \leq a_{\max,t} \tag{4}$$

where $a_{i,t}$ denotes the tangent acceleration of the ith vehicle, $a_{max,t}$ denotes the maximum tangent acceleration of the vehicles.

(3) Velocity constraint.

$$v_{\min} \leq v_i(t) \leq v_{\max} \tag{5}$$

where v_i denotes the speed of the ith vehicle, $v_{\min}$ and $v_{\max}$ denote the minimum and maximum velocities of the vehicles, respectively.

(4) Minimum separation distance constraint. The distance between any two vehicles at the same time should exceed a threshold value to avoid collision. Thus, the following expression is obtained:

$$D(i, j, t) \geq D_{safe}.(\forall i, j, t.\ i, j \in [1, N]) \tag{6}$$

where $D(i, j, t)$ represents the distance between the ith vehicle and the jth vehicle at any time t, D_{safe} represents the safe distance avoiding collision.

In order to achieve the simultaneous arrival of multiple unmanned vehicles, the following equation constraint is obtained:

$$t_g = t_{g,1} = t_{g,2} = \cdots = t_{g,N} \tag{7}$$

where t_g denotes the arrival times of multiple vehicles, and $t_{g,i}(i = 1, 2 \ldots N)$ denotes the time spent by the ith vehicle to sail from the initial pose to the target pose. The shortest simultaneous arrival time is considered as follows:

$$\min t_g \tag{8}$$

3. T–S Bezier curve for simultaneous arrival planning

There are $N(N \geq 3)$ vehicles that sail to their respective target points at the same time and form a certain formation at the end and subsequently maintain the formation. The T–S Bezier curve is designed by considering time as a one-dimensional variable of the Bezier curve. This novel manner achieves the arrival of a vehicle at a fixed time. The vehicle's physical constraints are considered, and the suboptimal T–S Bezier path is obtained by selecting the control points by using genetic algorithm. It contributes to the search for a suboptimal simultaneous arrival time.

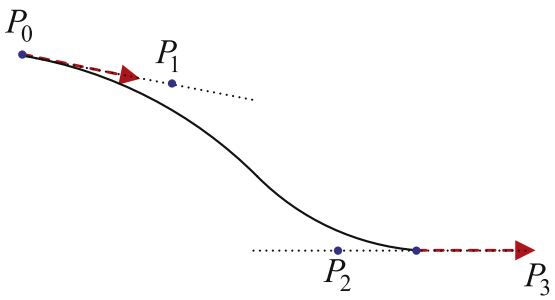

Fig. 2. A cubic Bezier curve.

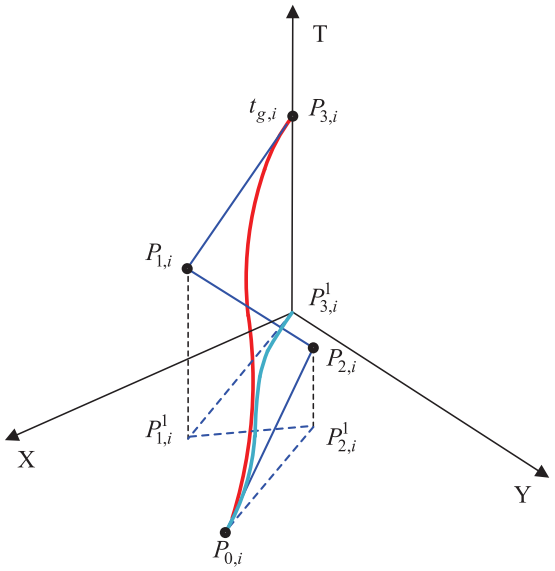

Fig. 3. T–S Bezier curve.

3.1. Design and generation of the T–S Bezier curve

The n-degree Bezier curve with $n+1$ control points $(P_0, P_1 \ldots P_{n+1})$ is defined as

$$P(\tau) = \sum_{k=0}^{n} B_{k,n}(\tau) P_k, B_{k,n}(\tau) = C_n{}^k (I-\tau)^{n-k} \tau^k \tag{9}$$

where $\tau \in [0, 1]$. $B_{i,n}(\tau)$ is Bernstein polynomials. Here, Bezier curve determined by the four control points is selected to realize path planning of multiple unmanned vehicles. A cubic Bezier curve is depicted in Fig. 2, where the four control points (denoted as $P_1 \sim P_4$) are marked. There are two important properties of a cubic Bezier curve that are useful for a T–S Bezier curve:

(1) Endpoint interpolation

$$P(0) = P_0$$
$$P(1) = P_3$$

(2) Tangent vectors at endpoints

$$P'(0) = 3(P_1 - P_0)$$
$$P'(1) = 3(P_3 - P_2)$$

The four control points of the ith vehicle Bezier curve are represented as $(P_{0,i}, P_{1,i}, P_{2,i}, P_{3,i})$. Time is considered as a one-dimensional variable of the Bezier curve. Thus, a Temporal–Spatial Bezier curve is created. As illustrated in Fig 3, the red curve corresponds to the T–S Bezier curve, $P_{0,i}$ represents initial position that is denoted as $(x_{s,i}, y_{s,i}, 0)$, and $P_{3,i}$ represents the target point denoted as $(x_{g,i}, y_{g,i}, t_{g,i})$. $P_{1,i}$ and $P_{2,i}$ are the selected points which determine the shape of Bezier curve that is denoted as $(x_{1,i}, y_{1,i}, t_{1,i})$ and $(x_{2,i}, y_{2,i}, t_{2,i})$, respectively. Hence, $P_{1,i}^1, P_{2,i}^1, P_{3,i}^1$ denote the projection points of $P_{1,i}, P_{2,i}, P_{3,i}$ on

the $x-y$ plane, respectively. Therefore, the following expression is obtained:

$$\begin{cases} x_i(\tau) = a_{x,i}\tau^3 + b_{x,i}\tau^2 + c_{x,i}\tau + d_{x,i} \\ y_i(\tau) = a_{y,i}\tau^3 + b_{y,i}\tau^2 + c_{y,i}\tau + d_{y,i} \\ t_i(\tau) = a_{t,i}\tau^3 + b_{t,i}\tau^2 + c_{t,i}\tau + d_{t,i} \end{cases} \tag{10}$$

Here,

$$\begin{cases} a_{x,i} = -x_{s,i} + 3x_{1,i} - 3x_{2,i} + x_{3,i} \\ b_{x,i} = 3x_{s,i} - 6x_{1,i} + 3x_{2,i} \\ c_{x,i} = -3x_{s,i} + 3x_{1,i} \\ d_{x,i} = x_{s,i} \\ a_{y,i} = -y_{s,i} + 3y_{1,i} - 3y_{2,i} + y_{3,i} \\ b_{y,i} = 3y_{s,i} - 6y_{1,i} + 3y_{2,i} \\ c_{y,i} = -3y_{s,i} + 3y_{1,i} \\ d_{y,i} = y_{s,i} \\ a_{t,i} = 3t_{1,i} - 3t_{2,i} + t_{g,i} \\ b_{t,i} = 6t_{1,i} + 3t_{2,i} \\ c_{t,i} = 3t_{1,i} \\ d_{t,i} = 0 \end{cases} \tag{11}$$

Based on the property of the Bezier curve, the second control point is located on the direction of the velocity at the initial point, and this is determined by $\theta_{s,i}$. Moreover, the third control point is located on the direction opposite to the velocity at the target point, and this is determined by $\theta_{g,i}$. Therefore, we can get

$$\begin{aligned} (x_{1,i}, y_{1,i}) &= (x_{s,i}, y_{s,i}) + k_{1,i} \times (cos\theta_{s,i}, sin\theta_{s,i}), k_{1,i} > 0 \\ (x_{2,i}, y_{2,i}) &= (x_{g,i}, y_{g,i}) - k_{2,i} \times (cos\theta_{g,i}, sin\theta_{g,i}), k_{2,i} > 0 \end{aligned} \tag{12}$$

where

$$\begin{aligned} k_{1,i} &= \sqrt{(x_{1,i} - x_{s,i})^2 + (y_{1,i} - y_{s,i})^2}, \\ k_{2,i} &= \sqrt{(x_{g,i} - x_{2,i})^2 + (y_{g,i} - y_{2,i})^2}, \end{aligned} \tag{13}$$

Moreover, the velocity of the vehicle is given as follows:

$$\begin{aligned} v_i(\tau) &= \sqrt{{v_{x,i}}^2(\tau) + {v_{y,i}}^2(\tau)} \\ &= \frac{\sqrt{(3a_{x,i}\tau^2 + 2b_{x,i}\tau + c_{x,i})^2 + (3a_{y,i}\tau^2 + 2b_{y,i}\tau + c_{y,i})^2}}{3a_{t,i}\tau^2 + 2b_{t,i}\tau + c_{t,i}} \end{aligned} \tag{14}$$

By substituting $\tau = 0$ and $\tau = 1$ into (14), we can obtain

$$\begin{aligned} v_{s,i} &= \frac{\sqrt{{c_{x,i}}^2 + {c_{y,i}}^2}}{c_{t,i}} \\ &= \frac{\sqrt{(x_{1,i} - x_{s,i})^2 + (y_{1,i} - y_{s,i})^2}}{t_{1,i}} \\ &= \frac{k_{1,i}}{t_{1,i}} \end{aligned} \tag{15}$$

$$\begin{aligned} v_{g,i} &= \frac{\sqrt{(3a_{x,i} + 2b_{x,i} + c_{x,i})^2 + (3a_{y,i} + 2b_{y,i} + c_{y,i})^2}}{3a_{t,i} + 2b_{t,i} + c_{t,i}} \\ &= \frac{\sqrt{(x_{g,i} - x_{2,i})^2 + (y_{g,i} - y_{2,i})^2}}{t_{g,i} - t_{2,i}} \\ &= \frac{k_{2,i}}{t_{g,i} - t_{2,i}} \end{aligned} \tag{16}$$

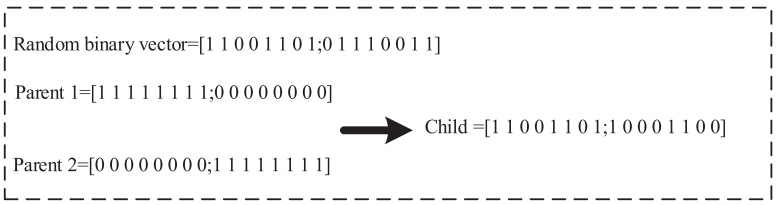

Fig. 4. Crossover operator. Parent 1: $k_1 = 255, k_2 = 0$, Parent 2: $k_1 = 0, k_2 = 255$.

Consequently, the final formula for the selection of the second control point and the third control point is obtained as follows:

$$\begin{aligned} P_{1,i} &= (x_{s,i} + k_{1,i} \times cos\theta_{s,i}, y_{s,i} + k_{1,i} \times sin\theta_{s,i}, k_{1,i}/v_{s,i}) \\ P_{2,i} &= (x_{g,i} - k_{2,i} \times cos\theta_{g,i}, y_{g,i} - k_{2,i} \times \sin\theta_{g,i}, t_{g,i} - k_{2,i}/v_{g,i}) \end{aligned} \tag{17}$$

3.2. The suboptimal T–S Bezier curve considering physical constraints

The optimization problem is considered to find a suboptimal T–S Beizer curve with physical constraints including curvature radius constraint, tangent acceleration constraint, and velocity constraint. Thus, path length is selected as an optimization objective. Hence, T–S Bezier curves are generated by controlling the four points, and the ith vehicle's path length is a function of $k_{1,i}$ and $k_{2,i}$ as follows:

$$\begin{aligned} g &= \int_0^1 \sqrt{\left(\frac{dx_i(\tau)}{d\tau}\right)^2 + \left(\frac{dy_i(\tau)}{d\tau}\right)^2}\, d\tau + \lambda \cdot \delta \\ &= f(k_{1,i}, k_{2,i}) \end{aligned} \tag{18}$$

where λ corresponds to a very large number, and δ is calculated as follows:

$$\delta = \begin{cases} 0, & if\ (3) - (5)\ are\ satisfied \\ 1, & else \end{cases} \tag{19}$$

Thus this problem can be written as follows:

$$\begin{aligned} &\min_{k_{1,i}, k_{2,i}} f(k_{1,i}, k_{2,i}) \\ &s.t. \quad 0 \le k_{1,i} \le d_i \\ &\qquad 0 \le k_{2,i} \le d_i \\ &\qquad 0 < t_{1,i} < t_{2,i} < t_{g,i} \end{aligned} \tag{20}$$

where d_i denotes the distance between $(x_{s,i}, y_{s,i})$ and $(x_{g,i}, y_{g,i})$.

In order to solve this, a genetic algorithm is employed [7], which consists of the following three basic components: selection, crossover, and mutation. A fitness function is selected as $\frac{1}{f(k_{1,i}, k_{2,i})}$. The binary coding method is adopted to express the population. The initial population that includes 20 individuals is randomly created, and this satisfies the bounds. In a selection operator, Roulette Wheel Selection is utilized to locate the individuals, and the probability of individual selection considered as proportional to the fitness of an individual. A crossover operator combines two parents to form a new individual in which Scattered Crossover is applied as illustrated in Fig. 4. A binary vector is randomly generated. Genes are selected from the first parent if the corresponding position of the binary vector is '1' while genes are selected from the second parent if the corresponding position is '0'. Thus, genes are combined to produce the child. The aim of a mutation operator involves adding to the diversity of the population in which an adaptive operator is adopted. Hence, probability of mutation decreases along with the increase of the fitness of the generation.

3.3. Determination of the suboptimal arrival time

As the optimization of the T–S Bezier curve considering path length is obtained, we can get the suboptimal arrival time of the ith vehicle (denoted as $t_{min,i}$), which is summarized as Algorithm 1. In Algorithm 1, μ is a small value which reflects the precision of the search for suboptimal arrival time. $t_{ref,i}$ serves as a lower bound of possible arrival time, which is discussed as three cases based on two reference length values: $s_{i,1}$ and $s_{i,2}$, which can be respectively calculated as follows:

$$s_{i,1} = \frac{v_{\max}^2 - v_{s,i}^2}{2a_{\max}} + \frac{v_{\max}^2 - v_{g,i}^2}{2a_{\max}}, \tag{21}$$

$$s_{i,2} = \frac{|v_{g,i}^2 - v_{s,i}^2|}{2a_{\max}}. \tag{22}$$

Algorithm 1 Determination of the suboptimal arrival time of the ith vehicle.

```
Compute t_{ref,i} according to (23);
m = 0;
while (1) do
    Get min f(k_1, k_2) optimizing (20) using genetic algorithm when t_{g,i} = t_{ref,i} + μ * m;
    if min f(k_1, k_2) > λ · δ then
        m = m + 1
    else
        Break;
    end if
end while
t_{min,i} = t_{g,i};
```

d_i is the distance between $(x_{s,i}, y_{s,i})$ and $(x_{g,i}, y_{g,i})$. $t_{ref,i}$ can be obtained as

$$t_{ref,i} = \begin{cases} \frac{\sqrt{a_{\max} d_i + \frac{v_{s,i}^2 + v_{g,i}^2}{2}} - v_{s,i}}{a_{\max}} \\ \qquad + \frac{\sqrt{a_{\max} d_i + \frac{v_{s,i}^2 + v_{g,i}^2}{2}} - v_{g,i}}{a_{\max}}, & s_{i,1} \geq d_i \geq s_{i,2} \\ t_{i,1} + \frac{(d_i - s_{i,1})}{v_{\max}}, & d_i > s_{i,1} \\ \frac{|v_{g,i} - v_{s,i}|}{a_{\max}}, & d_i < s_{i,2} \end{cases} \tag{23}$$

where $t_{i,1}$ is calculated as follows:

$$t_{i,1} = \frac{v_{\max} - v_{s,i}}{a_{\max}} + \frac{v_{\max} - v_{g,i}}{a_{\max}}. \tag{24}$$

3.4. Summary for simultaneous arrival planning

Simultaneous arrival planning of multiple vehicles is listed in Algorithm 2. First, the suboptimal arrival time of each

Algorithm 2 Simultaneous arrival planning of multiple vehicles.

```
for i = 1... N do
    Compute t_{min,i} by Algorithm 1;
end for
t_g = max(t_{min,i});
for i = 1... N do
    Obtain k_{1,i}, k_{2,i} optimizing (20) using genetic algorithm in case that the simultaneous arrival time is t_g;
    Get the two control points of the ith vehicle by (17) and the corresponding T–S Bezier curve;
end for
if (6) is not satisfied then
    Search the new control points near the previous control points with regard to the intersected paths;
end if
Discretize and obtain position and velocity related to time.
```

vehicle is obtained by Algorithm 1. Thus, the suboptimal simultaneous arrival time of the multiple vehicles (denoted as t_g) is determined, and corresponds to the maximum of $t_{\min,i}$. Subsequently, the two control points of the ith vehicle are given by the genetic algorithm in case that the simultaneous arrival time is set as t_g. In order to avoid collision between unmanned vehicles, Simultaneous arrival planning should satisfy (6). If this equation is not satisfied, it is necessary to search the two control points of responding intersected paths again to realize collision avoidance.

Simultaneous arrival planning is implemented by T–S Bezier curve, which is designed by considering time as a one-dimensional variable of the Bezier curve. The T–S Bezier curve is a feasible and safe path within the constraints of minimum curvature radius, tangent acceleration, and velocity, which are taken into consideration by genetic algorithm. Therefore, the path and velocity curves of vehicle are concurrently planned. This novel design allows the vehicle to spend an appointed time sailing from the initial pose to the final pose. The simultaneous arrival of multiple unmanned vehicles corresponds to a time-optimal problem. Therefore, the suboptimal simultaneous arrival time of multiple unmanned vehicles should be found. Algorithm 1 depicts the search for the suboptimal arrival time of the ith vehicle. The search begins with an initial arrival time until the suboptimal T–S Bezier curve meeting physical constraints is determined by a genetic algorithm. The initial arrival time is computed by (23). Hence, the maximum value of the suboptimal arrival times of all the vehicle is selected as

Table 1
The initial and final configurations of unmanned vehicles.

	$F_{s,i}(x_{s,i}, y_{s,i}, \theta_{s,i}, v_{s,i})$	$F_{g,i}(x_{g,i}, y_{g,i}, \theta_{g,i}, v_{g,i})$
Vehicle 1	(−100, 0, 0°, 12 m/s)	(500, 300, 0°, 20 m/s)
Vehicle 2	(−100, 300, 0°, 25 m/s)	(600, 450, 0°, 20 m/s)
Vehicle 3	(−100, 600, 0°, 18 m/s)	(600, 550, 0°, 20 m/s)
Vehicle 4	(−100, 900, 0°, 10 m/s)	(500, 700, 0°, 20 m/s)

Table 2
Specific parameters of simultaneous arrival planning-a.

	$t_{ref,i}$	$t_{min,i}$	t_g	$k_{1,i}$	$k_{2,i}$
Vehicle 1	27.61 s	30.81 s	34.26 s	115.42	77.86
Vehicle 2	28.74 s	29.74 s	34.26 s	125.07	136.99
Vehicle 3	30.27 s	34.26 s	34.26 s	75.01	36.07
Vehicle 4	26.30 s	29.69 s	34.26 s	147.33	110.30

Table 3
Specific parameters of simultaneous arrival planning-b.

	$P_{1,i}$	$P_{2,i}$	Path length
Vehicle 1	(15.42, 0, 9.62)	(422.14, 300, 30.37)	674.55 m
Vehicle 2	(25.08, 300, 5.00)	(463.01, 450, 27.41)	717.09 m
Vehicle 3	(−24.99, 600, 4.17)	(563.93, 550, 32.46)	701.82 m
Vehicle 4	(47.33, 900, 14.73)	(389.70, 700, 28.75)	635.37 m

the suboptimal simultaneous arrival time of multiple unmanned vehicles. Once t_g is determined, the suboptimal T–S Bezier curve for the ith unmanned vehicle is obtained by using a genetic algorithm. The proposed method that uses a T–S Bezier curve is effective when the distance between initial and final poses are large. Otherwise, it is necessary to add way-points between the initial and final poses, and two or more T–S Bezier curves may be used for path planning.

4. Numerical results

4.1. Results of simultaneous arrival planning

The proposed method for the simultaneous arrival of multiple unmanned vehicles is evaluated by using numerical simulations. The four unmanned vehicles 1–4 are selected to achieve simultaneous arrival from their respective initial configurations to corresponding final configurations, which are tabulated in Table 1. The physical constraints are selected as follows:

$$\begin{aligned} &R_i(s_{a,i}) \geq 30 \text{ m}, \\ &|a_{i,t}(t)| \leq 5 \text{ m/s}^2, \\ &5 \text{ m/s} \leq v_i(t) \leq 25 \text{ m/s}, \\ &D(i,j,t) \geq 15 \text{ m}. (\forall i,j,t.\ i,j \in [1,N]). \end{aligned} \tag{25}$$

The suboptimal arrival time of ith vehicle (denoted as $t_{min,i}$) should be obtained by Algorithm 1 beginning with an initial value $t_{ref,i}$. The suboptimal simultaneous arrival time is subsequently calculated ($t_g = 34.26$ s). Thus, the two control points of the ith vehicle are given by a genetic algorithm, and the suboptimal T–S Bezier curve is generated for the ith unmanned vehicle as seen in Fig. 5. t_g denotes the simultaneous arrival time of multiple vehicles. The T–S Bezier curve of the ith vehicle connects the respective initial pose to the respective final pose smoothly (See Fig. 5(d)). Additionally, $t_{ref,i}$, $t_{min,i}$, t_g, $k_{1,i}$, and $k_{2,i}$ are tabulated in sequence in Table 2. $P_{1,i}$, $P_{2,i}$, the length of T–S Bezier curve planned for the ith vehicle are tabulated in Table 3 in sequence. Fig. 6 gives the description of the T–S Bezier curve including time history of curvature, heading angle, velocity and tangent acceleration. Evidently, the generated trajectory satisfies the constraints of minimum curvature radius, tangent acceleration, and velocity. Moreover, the trajectory is smooth in terms of radius curvature, heading angle and velocity. The four vehicles can satisfy the respective final configurations including the final poses and the final velocities. Hence, T–S Bezier curve is effective in achieving simultaneous arrival of multiple vehicles.

4.2. Comparisons and discussions

To the best of the authors' knowledge, only a few studies applied path planning methods to achieve the simultaneous arrival of multiple unmanned vehicles. Shanmugavel et al. applied Dubins curves with the same path length to realize simultaneous arrival of multiple unmanned vehicles, which can satisfy the radius curvature constraint [21]. When compared with

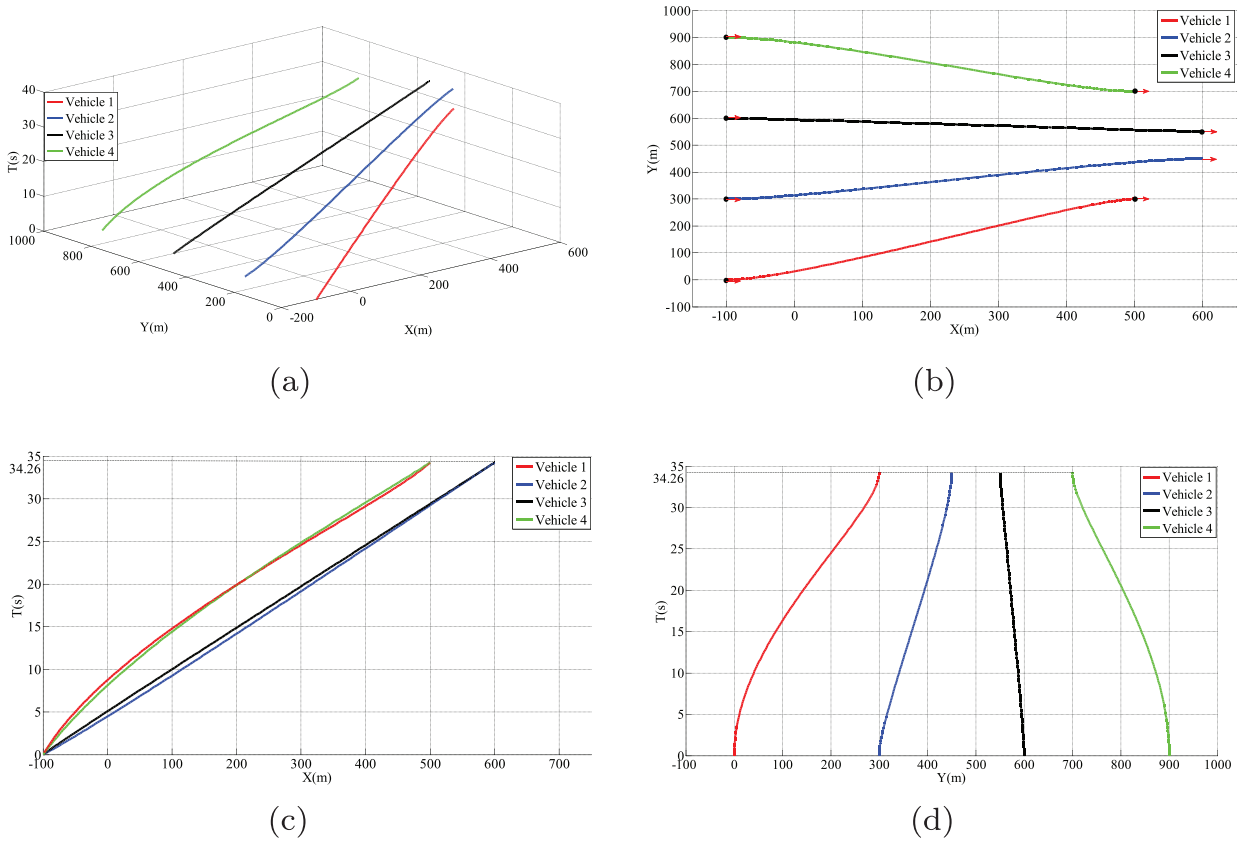

Fig. 5. T–S Bezier curves for simultaneous arrival. (a) T–S Bezier curves for the multiple vehicle at a constant arrival time corresponding to t_g. (b) The projected trajectories of T–S Bezier curves on the $X-Y$ plane. (c) The projected trajectories of T–S Bezier curves on the $X-T$ plane. The projected trajectories of T–S Bezier curves on the $Y-T$ plane.

Table 4
Final configurations of vehicles-a.

	Vehicle 1			Vehicle 2		
	$P_{g,1}$	$\theta_{g,1}$	$v_{g,1}$	$P_{g,2}$	$\theta_{g,2}$	$v_{g,2}$
1	(800, 300)	300°	20 m/s	(800, 400)	300°	20 m/s
2	(500, 300)	0°	20 m/s	(600, 450)	0°	20 m/s
3	(500, 300)	0°	20 m/s	(500, 400)	0°	20 m/s
4	(900, −200)	300°	20 m/s	(700, −200)	300°	20 m/s
5	(200, 1000)	45°	20 m/s	(450, 1000)	45°	20 m/s
6	(1000, −600)	315°	20 m/s	(1000, −400)	315°	20 m/s
7	(800, 500)	45°	20 m/s	(600, 500)	45°	20 m/s
8	(1000, 500)	45°	20 m/s	(800, 500)	45°	20 m/s

Dubins curves, the method in [17] provides a path with more flexibility and fewer discontinuity points on a curvature command although with a longer path length. Therefore, the simultaneous arrival time obtained by Shanmugavel et al. [21] is shorter than the one obtained by Oh et al. [17] on the same conditions. Here we just give the comparisons of the proposed method and the method in [21], which are evaluated by simultaneous arrival time. The initial poses of multiple vehicles are the same as shown in Section 4.1, and the initial velocities of all the vehicles are 20 m/s. Eight sets of data with final configurations of vehicles are tabulated in Tables 4 and 5. The simultaneous arrival times calculated by the proposed method and method in [21] are listed in Table 6. The differences $e(\%)$ in Table 6 are given to compare simultaneous arrival times computed by the proposed method with the ones computed by using the approach proposed by Shanmugavel et al. [21]. As depicted in Fig. 7, both the simultaneous arrival curves generated by the two methods are planned according to the first line of the data in Tables 4 and 5. Fig. 8 illustrates the time history of the curvature of these two kinds of curves for simultaneous arrival. The T–S Bezier curve is continuous in terms of curvature. However, the Dubins curve is composed of two arcs and a tangent line between arcs, and thus two discontinuities in curvature occur (See Fig. 8(b)). Hence, the Dubins curve is not desirable especially for fixed-wing UAVs because they follow an approximated continuous trajectory and do not precisely

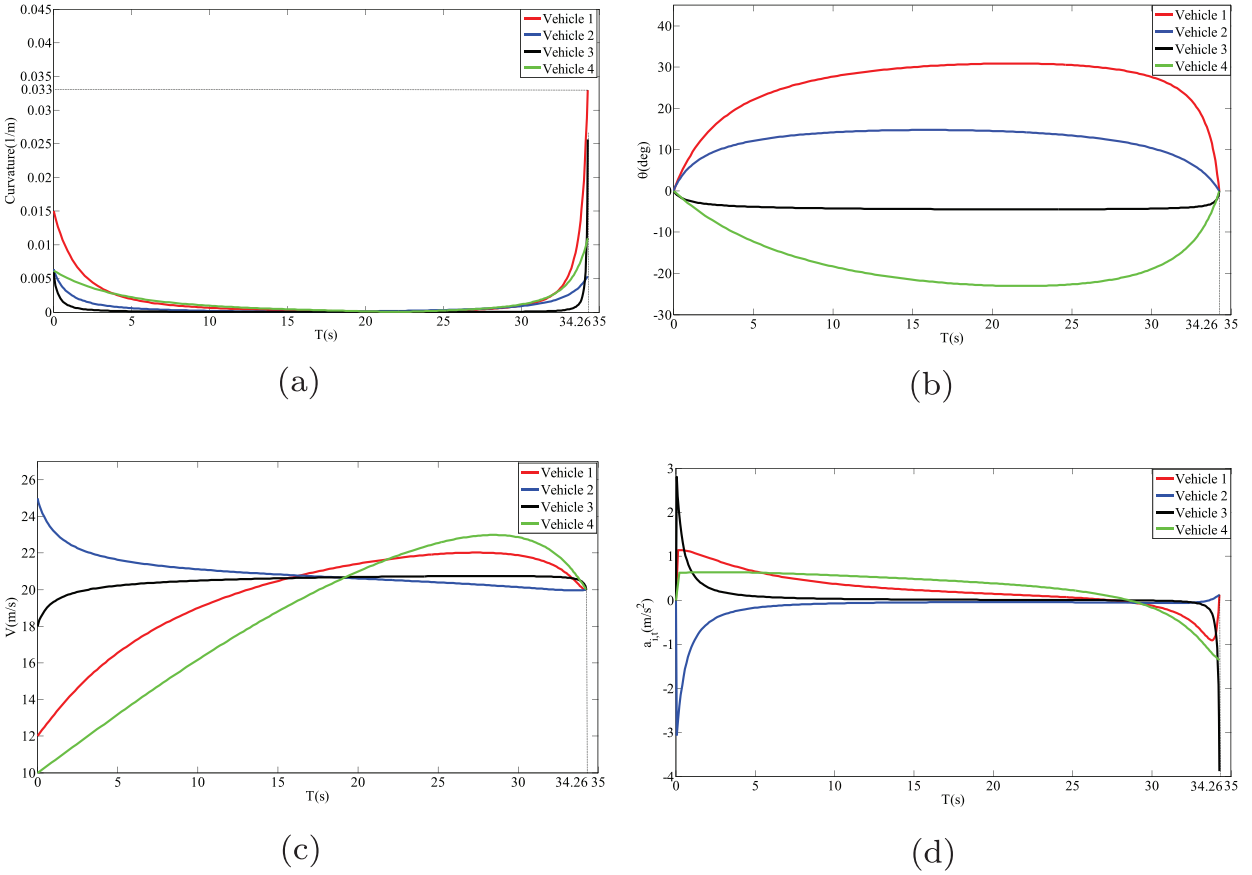

Fig. 6. The description of the T–S Bezier curve. (a) Time history of curvature. (b)Time history of heading angle. (c) Time history of velocity. (d) Time history of tangent acceleration.

Table 5
Final configurations of vehicles-b.

	Vehicle 3			Vehicle 4		
	$P_{g,3}$	$\theta_{g,3}$	$v_{g,3}$	$P_{g,4}$	$\theta_{g,4}$	$v_{g,4}$
1	(800, 500)	300°	20 m/s	(800, 600)	300°	20 m/s
2	(600, 550)	0°	20 m/s	(500, 700)	0°	20 m/s
3	(500, 500)	0°	20 m/s	(500, 600)	0°	20 m/s
4	(500, −200)	300°	20 m/s	(500, −200)	300°	20 m/s
5	(700, 1000)	45°	20 m/s	(950, 1000)	45°	20 m/s
6	(1000, −200)	315°	20 m/s	(1000, 0)	315°	20 m/s
7	(600, 700)	45°	20 m/s	(800, 700)	45°	20 m/s
8	(800, 700)	45°	20 m/s	(1000, 700)	45°	20 m/s

Table 6
The comparison of the two methods.

	Simultaneous arrival time		
	The proposed method	method in [21]	e(%)
1	45.25 s	48.04	6.17%
2	29.84 s	35.80 s	19.97%
3	29.83 s	33.59 s	12.60%
4	50.92 s	58.96 s	15.79%
5	47.06 s	52.82 s	12.24%
6	59.75 s	71.14 s	19.06%
7	43.18 s	51.52 s	19.31%
8	50.33 s	60.45 s	20.11%

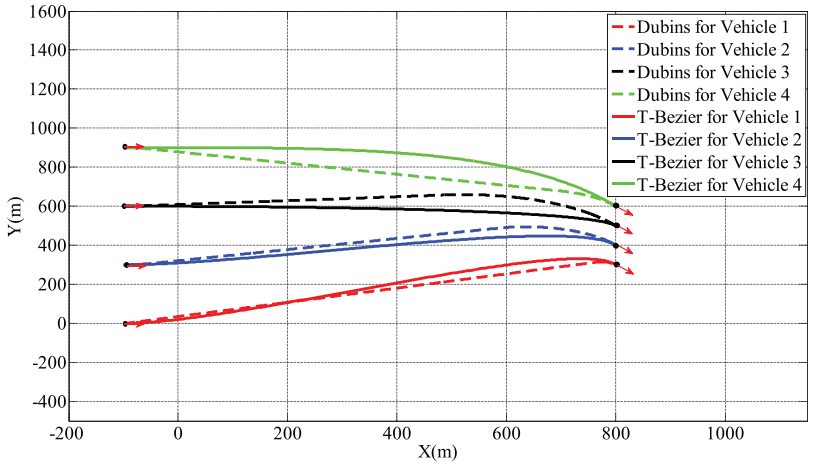

Fig. 7. T–S Bezier and Dubins curves for simultaneous arrival of multiple unmanned vehicles, respectively.

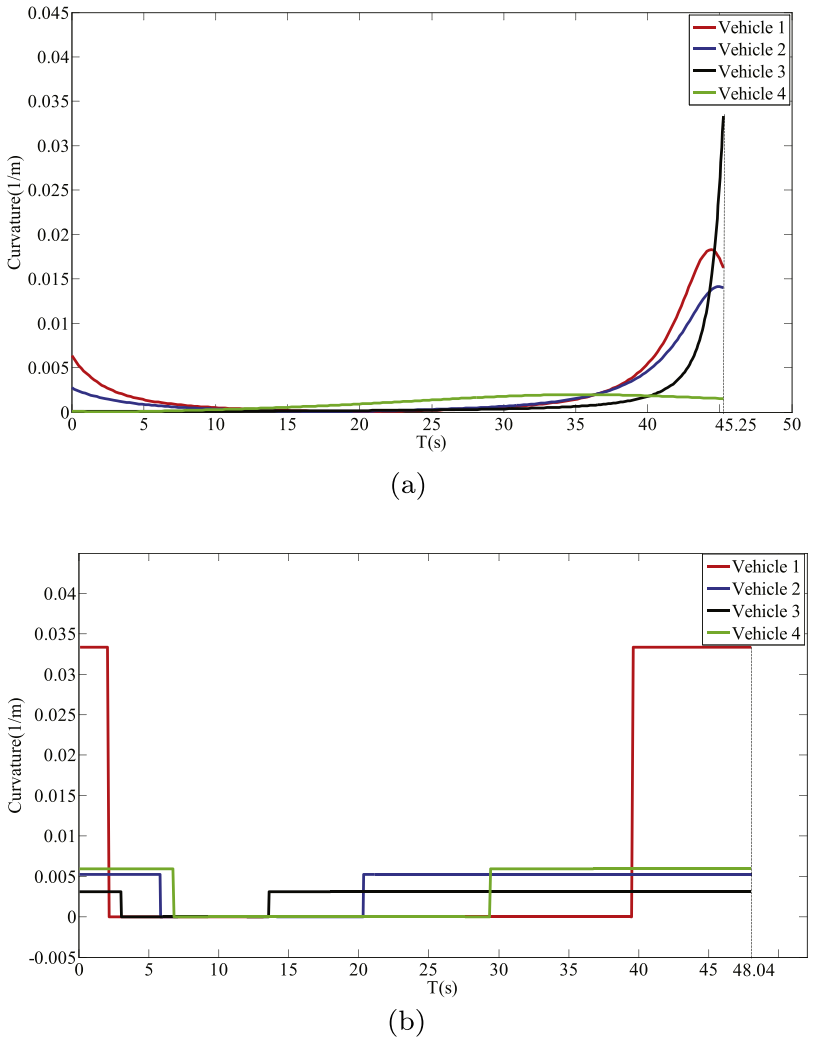

Fig. 8. Time history of the curvature of two types of curves for simultaneous arrival. (a) Time history of the curvature of T–S Bezier curve. (b)Time history of the curvature of Dubins curve.

track trajectories with discontinuous curvature [17]. With respect to the comparisons, the following is summarized: Under the same conditions, the simultaneous arrival time calculated by the proposed method is shorter than the one calculated by the methods proposed by Shanmugavel et al. [21]. Moreover, discontinuity curvature doesn't occur on the path provided by the proposed method while discontinuous curvature command occurs twice on the path proposed by Shanmugavel et al. [21] and once on the path proposed by Oh et al. [17].

5. Conclusion

In this paper, we have presented a novel curve named Temporal–Spatial (T–S) Bezier for simultaneous arrival of multiple unmanned vehicles. Optimization is adopted to search for the shortest T–S Bezier curve within the constraints of minimum curvature radius, tangent acceleration and velocity. The suboptimal time of simultaneous arrival achieved by T–S Bezier curves is determined. Numerical results have proved that T–S Bezier curve is effective and feasible to achieve the simultaneous arrival of multiple unmanned vehicles.

Acknowledgements

This work was supported in part by the National Natural Science Foundation of China under Grant 61233014, 61333016, 61603388, 61633017, in part by the basic research program under Grant B132011XX, and in part by the Foundation for Innovative Research Groups of the National Natural Science Foundation of China under Grant 61421004.

References

[1] C.G.L. Bianco, O. Gerelli, Generation of paths with minimum curvature derivative with η^3-splines, IEEE Trans. Automat. Sci. Eng. 7 (2) (2010) 249–256.
[2] M. Brezak, I. Petrovic, Path smoothing using clothoids for differential drive mobile robots, in: Proc. World Congr. Int. Federat. Automat. Control, Milano, Italy, 2011, pp. 1133–1138.
[3] C. Chen, Y. He, C. Bu, J. Han, X. Zhang, Quartic Bezier curve based trajectory generation for autonomous vehicles with curvature and velocity constraints, in: Proc. 51st IEEE Conf. Robotics and Automation, Hong Kong, China, 2014, pp. 6108–6113.
[4] R. Dai, J.E. Cochran, Path planning for multiple unmanned aerial vehicles by parameterized Cornu-Spirals, in: Proc. American Control Conf, St. Louis, USA, 2009, pp. 2391–2396.
[5] M. Elbanhawi, M. Simic, R.N. Jazar, Continuous path smoothing for car-like robots using B-spline curves, J. Intell. Robot. Syst. (2015) 1–34.
[6] F. Ghilardelli, G. Lini, A. Piazzi, Path generation using η^4-splines for a truck and trailer vehicle, IEEE Trans. Automat. Sci. Eng. 11 (1) (2014) 187–203.
[7] D.E. Goldberg, J.H. Holland, Genetic algorithms and machine learning, Mach. Learn. 3 (2) (1988) 95–99.
[8] D. Jung, P. Tsiotras, On-line path generation for unmanned aerial vehicles using B-spline path templates, J. Guid. Control, Dyn. 36 (6) (2013) 1642–1653.
[9] M. Kothari, I. Postlethwaite, A probabilistically robust path planning algorithm for UAVs using rapidly-exploring random trees, J. Intell. Robot. Syst. 71 (2) (2013) 231–253.
[10] L.C. Lin, M. Goodrich, Hierarchical heuristic search using a Gaussian mixture model for UAV coverage planning, IEEE Trans. Cybern. 44 (12) (2014) 2532–2544.
[11] A.M. Lekkas, T. Fossen, Integral LOS path following for curved paths based on a monotone cubic Hermite spline parameterization, IEEE Trans. Control Syst. Technol. 22 (6) (2014) 2287–2301.
[12] Y. Lu, X. Huo, O. Arslan, P. Tsiotras, Incremental multi-scale search algorithm for dynamic path planning with low worst-case complexity, IEEE Trans. Syst. Man, Cybern. B 41 (6) (2011) 1556.
[13] B. MacAllister, J. Butzke, A. Kushleyev, H. Pandey, M. Likhachev, Path planning for non-circular micro aerial vehicles in constrained environments, in: Proc. 50st IEEE Conf. Robotics and Automation, Karlsruhe, Germany, 2013, pp. 3933–3940.
[14] C.C. Murray, W.C. Park, Incorporating human factor considerations in unmanned aerial vehicle routing, IEEE Trans. Syst. Man Cybern. 43 (4) (2014) 860–874.
[15] D. Mellinger, A. Kushleyev, V. Kumar, Mixed-integer quadratic program trajectory generation for heterogeneous quadrotor teams, in: Proc. 49st IEEE Conf. Robotics and Automation, Saint Paul, USA, 2012, pp. 477–483.
[16] Y. Ma, H. Wang, Y. Xie, M. Guo, Path planning for multiple mobile robots under double-warehouse, Inf. Sci. 278 (2014) 357–379.
[17] H. Oh, D. Turchi, S. Kim, A. Tsourdos, L. Pollini, B. White, Coordinated standoff tracking using path shaping for multiple UAVs, IEEE Trans. Aerosp. Electron. Syst. 50 (1) (2014) 348–363.
[18] E.B. Portasa, D.D.L. Torre, A. Moreno, J.L.R. Martn, On the performance comparison of multi-objective evolutionary UAV path planners, Inf. Sci. 238 (2013) 111–125.
[19] A.T. Rashid, A.A. Ali, M. Frasca, L. Fortuna, Path planning with obstacle avoidance based on visibility binary tree algorithm, Robot. Auton. Syst. 61 (12) (2013) 1440–1449.
[20] V. Roberge, M. Tarbouchi, G. Labont, Comparison of parallel genetic algorithm and particle swarm optimization for real-time UAV path planning, IEEE Trans. Ind. Informat. 9 (1) (2013) 132–141.
[21] M. Shanmugavel, A. Tsourdos, R. Zbikowski, B.A. White, Path planning of multiple UAVs using Dubins sets, in: Proc. AIAA Guidance, Navigation, Control Conf., San Francisco, USA, 2005, pp. 1–17.
[22] T. Tomic, K. Schmid, P. Lutz, A. Domel, M. Kassecker, E. Mair, I.L. Grixa, F. Ruess, M. Suppa, D. Burschka, Research platform for indoor and outdoor urban search and rescue, IEEE Robot. Autom. Mag. 19 (3) (2012) 46–56.
[23] Y. Wang, S. Wang, M. Tan, Q. Wei, Real-time dynamic Dubins-Helix method for 3-D trajectory smoothing, IEEE Trans. Control Syst. Technol. 23 (2) (2015a) 730–736.
[24] Y. Wang, S. Wang, M. Tan, Path generation of autonomous approach to a moving ship for unmanned vehicles, IEEE Trans. Ind. Electron. (2015b), doi:10.1109/TIE.2015.2405904.
[25] K. Yang, S. Sukkarieh, An analytical continuous-curvature path-smoothing algorithm, IEEE Trans. Robot. 26 (3) (2010) 561–568.
[26] H. Yu, K. Meier, M. Argyle, R.W. Beard, Cooperative path planning for target tracking in urban environments using unmanned air and ground vehicles, IEEE/ASME Trans. Mechatron. 20 (2) (2015) 541–552.