2023 IEEE International Conference on Robotics and Automation (ICRA 2023)
May 29 - June 2, 2023. London, UK

Shunted Collision Avoidance for Multi-UAV Motion Planning with Posture Constraints

Gang Xu[1,3], Deye Zhu[1], Junjie Cao[1,*], Yong Liu[1,*], and Jian Yang[2]

***Abstract*— This paper investigates the problem of fixed-wing unmanned aerial vehicles (UAVs) motion planning with posture constraints and the problem of the more general symmetrical situations where UAVs have more than one optimal solution. In this paper, the posture constraints are formulated in the 3D Dubins method, and the symmetrical situations are overcome by a more collaborative strategy called the shunted strategy. The effectiveness of the proposed method has been validated by conducting extensive simulation experiments. Meanwhile, we compared the proposed method with the other state-of-the-art methods, and the comparison results show that the proposed method advances the previous works. Finally, the practicability of the proposed algorithm was analyzed by the statistic in computational cost. The source code of our method can be available at https://github.com/wuuya1/SCA.**

***Index Terms*— Collision avoidance, multi-UAV motion planning, posture constraints.**

I. INTRODUCTION

The multi-unmanned aerial vehicle (UAV) motion planning is essential for UAV swarm systems. In recent years, the UAV swarm has been in the spotlight as it has been proficiently applied to many fields like cooperative exploration [1], formation [2], [3], collaborative search [4], etc. The fixed-wing UAVs are more qualified for such applications due to their strengths in endurance and load capabilities. Hence, fixed-wing UAVs are considered in this article. However, many practical challenges should be considered. For example, fixed-wing UAVs have to comply with specific requirements in yaw angle or pitch angle, called posture constraints, like turning around, take-off, landing, etc.

The three-dimensional reciprocal velocity obstacles (RVO) method expanded from [5] has great potential to bridge the gap in fixed-wing UAVs' motion planning as its excellent scalability for the physical constraints. However, some symmetrical situations may occur while UAVs are traveling, in which UAVs have more than one optimal solution, resulting in deadlock or oscillation. In analyzing the most promising approaches to geometric collision avoidance in multi-vehicle systems, J. A. Douthwaite et al. [6] raised that the symmetry problem will lead to performance degradation. However, any solutions were not proposed to overcome this challenge.

To bridge the above gap in fixed-wing UAVs' motion planning, we proposed a novel shunted collision avoidance (SCA) method based on our previous work [7]. Unlike our previous work, we expanded the posture constraints to three-dimensional space according to the fixed-wing UAVs motion model. Unlike most of the work in multi-UAV motion planning, this work considers the more practical constraints and the situations of symmetry in a three-dimensional workspace. Here, the posture constraints are overcome by combining the 3D Dubins method [8] and the RVO algorithm [5] in 3D domains (RVO3D). Meanwhile, we consider the situations of symmetry by inspiration from the human traffic rules where the opposite vehicles will choose the same side to pass, such as passing on the right side. Finally, the results of numerous simulations demonstrate the validity and practical potential of our method. The main contributions can be summarized as follows.

- A novel shunted strategy is proposed to improve the smoothness of the UAVs' trajectory.
- We proposed the shunted collision avoidance (SCA) algorithm for fixed-wing UAVs' motion planning, in which UAVs' posture constraints are considered.
- We conducted three simulations on the AirSim platform

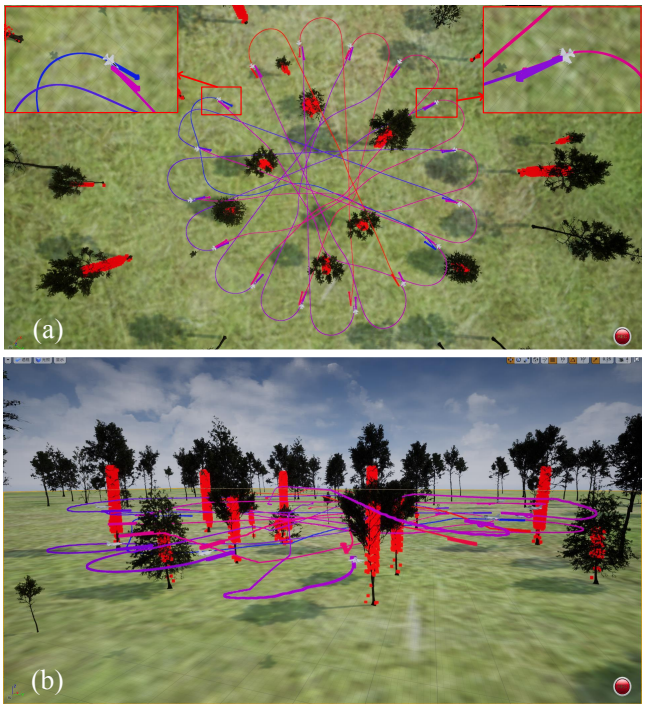

Fig. 1. Simulation results of the low altitude search task. Maximum heading rate: 0.774 rad; Maximum pitch rate: 0.670 rad; ET: 8.2 s (ideally: 4.4 s); ED: 5.683 m (ideally: 4.4 m); AS: 0.989 m/s (ideally: 1.0 m/s); CC: 17.723 ms. (a) Top view. (b) 3D view.

[1]Gang Xu, Deye Zhu, Junjie Cao, and Yong Liu are with the Institute of Cyber-Systems and Control, Zhejiang University, Hangzhou 310027, China.
[2]Jian Yang is with the Research and Development Academy of Machinery Equipment, Beijing 100089, China.
[3]Gang Xu is also with the Taizhou Institute of Zhejiang University, Taizhou 318000, China.
*Yong Liu and Junjie Cao are the corresponding authors (Email: yongliu@iipc.zju.edu.cn, cjunjie@zju.edu.cn).

[9] to present the unique performance of our method and conducted large-scale scenarios to evaluate the advantages of our method.

The rest of this paper is organized as follows. In Section II, we reviev the related works. In Section III, we formulate the UAV motion model, and we summarize the RVO algorithm in 3D domains and its limitations. In Section IV, we introduce our novel method for fixed-wing UAVs' motion planning. In Section V, simulation experiments are conducted, and the conclusion is in Section VI.

II. RELATED WORKS

This section will summarize the related works in multi-vehicle motion planning. Generally, the methods can be categorized into centralized and decentralized approaches.

Most of the centralized approaches are mainly developed by single-vehicle algorithms. For example, both the Push and Swap (PS) method [10] and the Conflict Based Search (CBS) method [11] are developed from the A* algorithm [12], and the Random sampling algorithm for cooperation [13] is extended from the Rapidly-exploring Random Tree (RRT) algorithm [14]. In addition, many approaches have been proposed to solve planning problems with non-holonomic constraints [15], [16]. Although the centralized approaches are logically simple, the methods are usually limited by the number of vehicles and perfect sensing, resulting in great difficulty for practice.

Different from centralized methods, each vehicle travels by observing neighbors' behavior without communication in decentralized approaches. In this case, controlling the multi-vehicle system may be more convenient. As a result, a variety of decentralized methods are developed to address the problems of multi-vehicle motion planning, including potential fields [17], sampling-based method [18], reinforcement learning [19], [20], and velocity obstacle approaches like [5], [21], [22], [23], and so forth.

Here, the velocity obstacle approaches are the most associated with our method in this paper. The early improved version of the velocity obstacles (VO) method [21] takes a total effort for collision avoidance like [24], leading to trajectory oscillation. The reciprocal velocity obstacles (RVO) method [5] overcomes this oscillation problem. Sequentially, the hybrid reciprocal velocity obstacle (HRVO) algorithm [23] enabled the vehicle to pass on the same side. Among the velocity obstacle approaches, one of the most popular is the optimal reciprocal collision avoidance (ORCA) algorithm [22] as it enhances the performance of collision avoidance and computational efficiency.

In addition, some related work considered vehicles' kinematic constraints, dynamic constraints, etc. For example, D. Bareiss et al. [25] consider non-homogeneous robot systems in the real world. Instead of simply regarding vehicles as a circle, R. Mao et al. [26] worked on the collision-free navigation of wheeled robots by integrating the linear model predictive control method into ORCA. J. Liu et al. [27] proposed the Dubins-RVO algorithm for aircraft on the deck by considering the posture constraints. Despite these above works can theoretically be expanded to three-dimensional space, they are only implemented in the two-dimensional plane. Fortunately, some motion planning methods based on geometric collision avoidance have also been investigated for UAVs. For instance, J. Snape et al. [28] proposed the ORCA3D method in a 3D workspace for simple airplanes without any kinematic and dynamic constraints. C. Y. Tan et al. [29] extended the velocity obstacle method for three-dimensional collision avoidance but was limited to the original concept of the velocity obstacles, which may lead to oscillation.

Although some above works have considered some practical constraints, most of them cannot be deployed in general 3D workspaces for fixed-wing UAVs.

III. BACKGROUND

In this section, the RVO method in 3D domains and its limitations are summarized briefly.

A. UAV Model

We assume that there are n fixed-wing UAVs in a three-dimensional workspace. At the same time, we denote the $\mathbf{p}_i = [x_i, y_i, z_i]^T$ as the position of UAV i and the $\chi_i = [\psi_i, \gamma_i, \phi_i]^T$ as the Euler angles (heading angle, flight-path angle, and bank angle) of UAV i in a north-east-down (NED) frame. The UAV motion model can be described as

$$\begin{aligned} \dot{x}_i &= V_i \cos\gamma_i \cos\psi_i \\ \dot{y}_i &= V_i \cos\gamma_i \sin\psi_i \\ \dot{z}_i &= -V_i \sin\gamma_i \\ \dot{\psi}_i &= \frac{g}{V_i}\tan\phi_i \end{aligned} \tag{1}$$

where $i \in N = \{1,2,3,...,n\}$ represents the index of UAV, V_i represents the airspeed of UAV i, g is the acceleration due to gravity at sea level. Then, we defined $\mathbf{q}_i = [x_i, y_i, z_i, \psi_i, \gamma_i]^T$ and $\mathbf{u}_i = [V_i, \dot{\psi}_i, \gamma_i]^T$ as the state and the control input of UAV i, respectively. Thus, we can expand the Eqn. (1) as

$$\dot{\mathbf{q}}_i = f(\mathbf{q}_i, \mathbf{u}_i) \tag{2}$$

To UAVs can be controlled by computer systems, we discrete the Eqn. (2) as

$$\mathbf{q}_{i,k+1} = f_t(\mathbf{q}_{i,k}, \mathbf{u}_{i,k}) \tag{3}$$

where $f_k(\mathbf{q}_{i,k}, \mathbf{u}_{i,k}) = \mathbf{q}_{i,k} + \Delta\mathbf{q}_{i,k}$, $\Delta\mathbf{q}_{i,k} = [\Delta x_{i,k}, \Delta y_{i,k}, \Delta z_{i,k}, \Delta\psi_{i,k}, \Delta\gamma_{i,k}]^T$, $\Delta x_{i,k} = T_s V_{i,k}\cos\gamma_{i,k}\cos\psi_{i,k}$, $\Delta y_{i,k} = T_s V_{i,k}\cos\gamma_{i,k}\sin\psi_{i,k}$, $\Delta z_{i,k} = -T_s V_{i,k}\sin\gamma_{i,k}$, $\Delta\psi_{i,k} = \psi_{i,k} - \psi_{i,k-1}$, $\Delta\gamma_{i,k} = \gamma_{i,k} - \gamma_{i,k-1}$, and T_s represents the sampling period.

Furthermore, we also consider the posture constraints of UAVs. Based on the work [8], the posture constraints are formulated as

$$\Delta\psi_{min} \le \Delta\psi_{i,k} \le \Delta\psi_{max}, \Delta\gamma_{min} \le \Delta\gamma_{i,k} \le \Delta\gamma_{max} \tag{4}$$

where $\Delta\psi_{min}$, $\Delta\psi_{max}$, $\Delta\gamma_{min}$, and $\Delta\gamma_{max}$ are determined by the physical properties of UAVs. Especially, the constraints of yaw angle rate can be described by the minimum turning radius ρ_{min} of the trajectory. To simplify, we regard the UAV

as a sphere whose radius is 0.5 m in this paper. At the same time, we also assume $\rho_{min} = 1.5$ m, $\Delta\gamma_{min} = -45°$, and $\Delta\gamma_{max} = 45°$. It should be reasonable as the values of the actual constraint are not so restricted.

B. RVO Method in 3D Domains

We regard UAV A as a sphere whose reference position is $\mathbf{p}_A$ and radius is r_A, moving at $\mathbf{v}_A$ in a 3D workspace. Similarly, UAV B was also regarded as a sphere whose reference position is $\mathbf{p}_B$ and radius is r_B, moving at $\mathbf{v}_B$ in the same space.

The mathematical formulation of the VO3D algorithm could be summarized as follows. We denote a sphere with $D(\mathbf{p_0}, r)$ in Eqn. (5), where the center is $\mathbf{p_0}$, the radius is r, and $\mathbf{p}$ is expressed as any point in space. In addition, the ray $\lambda(\mathbf{p}, \mathbf{v})$ is denoted in Eqn. (6) where the starting point is $\mathbf{p}$, the direction is $\mathbf{v}$, and τ represents the extension parameter. And then, we get the obstacles velocity set $VO_B^A(\mathbf{v}_B)$ in Eqn. (7), where $\mathbf{v}_{AB}$ is $\mathbf{v}_A - \mathbf{v}_B$ and r_{AB} is $r_A + r_B$.

$$D(\mathbf{p_0}, r) = \{\mathbf{p} \mid \|\mathbf{p} - \mathbf{p_0}\| < r\}, \tag{5}$$

$$\lambda(\mathbf{p}, \mathbf{v}) = \{\mathbf{p} + \tau\mathbf{v} \mid \tau \geq 0\}, \tag{6}$$

$$VO_B^A(\mathbf{v}_B) = \{\mathbf{v}_A \mid \lambda(\mathbf{p}_A, \mathbf{v}_{AB}) \cap D(\mathbf{p}_B, r_{AB}) \neq \emptyset\}. \tag{7}$$

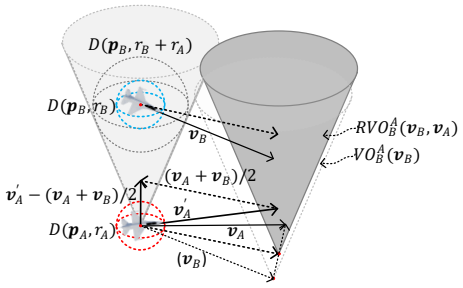

Fig. 2. The RVO3D concept geometric representation of reciprocal velocity obstacles set $RVO_B^A(\mathbf{v}_B, \mathbf{v}_A)$ of UAV A to UAV B.

Consequently, the basic idea of the RVO concept can be briefed as follows. When UAV A calculates the collision avoidance velocity $\mathbf{v}'_A$ at the $t+1$ moment, it believes that UAV B will also make a specific effort for collision avoidance. Hence, UAV A no longer holds that the velocity of UAV B at the moment of $t+1$ must be $\mathbf{v}_B$ (VO3D is), but another velocity after considering the motion of UAV A, such as $\frac{\mathbf{v}_A+\mathbf{v}_B}{2}$. Geometrically, the cone-zone inscribed in the sphere $D(\mathbf{p}_B, r_A + r_B)$ shown in Fig. 2 is translated at $\frac{\mathbf{v}_A+\mathbf{v}_B}{2}$ for expressing the reciprocal velocity obstacles set $RVO_B^A(\mathbf{v}_B, \mathbf{v}_A)$ of UAV A to UAV B (the cone-zone of dark-gray in Fig. 2). Here, when UAV A calculates the velocity of collision-free, its relative velocity also changes to $\mathbf{v}'_A - \frac{\mathbf{v}_A+\mathbf{v}_B}{2}$. Hence, the RVO3D concept holds that if the ray $\mathbf{v}'_A - \frac{\mathbf{v}_A+\mathbf{v}_B}{2}$ intersects the cone-zone of dark-gray in Fig. 2, UAV A and UAV B will certainly collide in the future. Otherwise, the two UAVs will be collision-free. Finally, the formulation of RVO3D is shown as

$$RVO_B^A(\mathbf{v}_B, \mathbf{v}_A) = \{\mathbf{v}'_A \mid (2\mathbf{v}'_A - \mathbf{v}_A) \in VO_B^A(\mathbf{v}_B)\}. \tag{8}$$

C. Limitations of the RVO Method

The RVO concept can guarantee that both UAVs automatically choose the same side to pass, which is named the Same Side lemma and is the key to achieving collision avoidance. The Same Side lemma had been proven and deduced in Formula (9), in which $\mathbf{v}_A + \mathbf{u}$ is the optimal velocity outside UAV B's reciprocal velocity obstacle for UAV A, and $\mathbf{v}_B - \mathbf{u}$ is the optimal velocity UAV A's reciprocal velocity obstacle for UAV B.

$$\mathbf{v}_A + \mathbf{u} \notin RVO_B^A(\mathbf{v}_B, \mathbf{v}_A) \Leftrightarrow \mathbf{v}_B - \mathbf{u} \notin RVO_A^B(\mathbf{v}_A, \mathbf{v}_B). \tag{9}$$

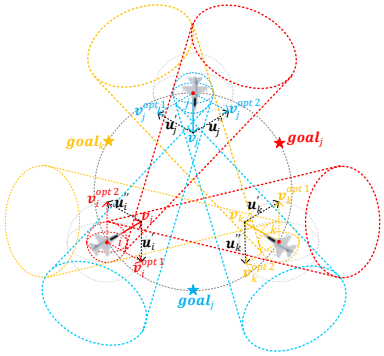

Fig. 3. The RVO3D method cannot guarantee that every UAV passes on the same side.

However, symmetrical situations are always more complex in more practical situations. Let us consider Fig. 3 for example, where both UAVs' vectors of velocity minimum change are no longer opposite vectors of each other, such as $\mathbf{u}'_i \neq -\mathbf{u}'_j$. In this case, the RVO3D method cannot guarantee that every UAV passes on the same side. In a word, this above lemma of passing on the same side is only limited to special symmetrical situations where are only two UAVs. In addition, the RVO3D method is limited as it only considers the simple kinematic constraints without more practical constraints. However, fixed-wing UAVs have some physical constraints in both yaw angle and pitch angle, which we define these constraints as posture constraints.

IV. METHODOLOGY

This section presents our shunted collision avoidance (SCA) for the motion planning of fixed-wing UAVs.

A. The Shunted Strategy

First, we introduce the notation U, which represents UAVs' universal velocity set. At the same time, let RVO be the set of reciprocal velocity obstacles and AV be the feasible velocity set. The UAV will achieve collision avoidance as long as the UAV selects the velocity among the AV. Considering any pair of UAV i and UAV j in the same 3D environment, the set $AV_j^i(\mathbf{v}_j, \mathbf{v}_i)$ of UAV i can be derived by Eqn. (10), in which the $\mathbf{v}_i$ is the current velocity of UAV i and the $\mathbf{v}_j$ is the current velocity of UAV j. Based on this above derivation, the feasible velocity set AV^i considered all the neighbors of UAV i could be formulated as the Eqn.

(11). Then, one optimal velocity $\mathbf{v}_i^{opt}$ is calculated through Eqn. (12), where $\mathbf{v}_i'$ is a feasible velocity among $AV^i$and $\mathbf{v}_i^{pref}$ is the desired velocity of UAV i. Finally, let OV_{SCA}^i be the optimal velocity set of UAV i. It can be deduced by Eqn. (13), where a threshold parameter Δv_{opt}, empirically is 0.03, will be set slightly greater than zero.

$$AV_j^i(\mathbf{v}_j,\mathbf{v}_i) = U_j^i(\mathbf{v}_j,\mathbf{v}_i) - RVO_j^i(\mathbf{v}_j,\mathbf{v}_i), \tag{10}$$

$$AV^i = \bigcap_{j\neq i} AV_j^i(\mathbf{v}_j,\mathbf{v}_i), \tag{11}$$

$$\mathbf{v}_i^{opt} = \arg\min_{\mathbf{v}_i'\in AV^i}(\|\mathbf{v}_i' - \mathbf{v}_i^{pref}\|), \tag{12}$$

$$OV_{SCA}^i = \{\mathbf{v}_i' \mid \|\mathbf{v}_i' - \mathbf{v}_i^{opt}\| < \Delta v_{opt}, \mathbf{v}_i' \in AV^i\}. \tag{13}$$

Then we develop the shunted strategy to guarantee that UAV i passes on the same side (right side). The details are as follows. We project the current velocity $\mathbf{v}_i$ and all the optimal velocities from three-dimensional space onto the XY plane and compute their angle θ in the XY plane, where $\theta \in [-\pi,\pi]$. Theoretically, the angle of $\mathbf{v}_i$ must be larger than or equal to the θ_{right} of the optimal velocity on the right side of $\mathbf{v}_i$. It also must be smaller than or equal to the θ_{left} of the optimal velocity on the left side of $\mathbf{v}_i$. Hence, the relation is expressed as $\theta_{right} \leq \theta(\mathbf{v}_i) \leq \theta_{left}$, in which $\theta_{right} \leq \theta_{left}$ is always true. These steps can be formulated as Eqn. (14), where $\mathbf{v}_i^{new}$ is the most feasible velocity.

$$\mathbf{v}_i^{new} = \arg\min_{\mathbf{v}_i^{cand}\in OV_{SCA}^i}(\theta(\mathbf{v}_i^{cand})). \tag{14}$$

The shunted strategy is summarized as the following four steps. The above details are presented in Algorithm 1, where the N is the set of UAV i's neighbors.

1) Let a UAV in the 3D domains be UAV i. All feasible velocity sets AV_j^i between UAV i and neighbor UAV j are computed using the RVO3D algorithm.
2) Then, the intersection operation for all AV_j^i achieves the feasible velocity set AV^i of UAV i.
3) An optimal velocity $\mathbf{v}_i^{opt}$ of UAV i can be calculated by Eqn. (12). Then, the Eqn. (13) is exploited to determine the optimal velocity set OV_{SCA}^i of UAV i.
4) Finally, the most feasible optimal velocity $\mathbf{v}_{new}'$ can be solved by using the idea of the shunted strategy.

Algorithm 1: The Shunted Strategy

Input : $\mathbf{p}_i$, $\mathbf{v}_i$, $\mathbf{v}_i^{pref}$, $\bigcup\mathbf{p}_j$, $\bigcup\mathbf{v}_j$ ($j \in N$, $j \neq i$) – UAV i config.

Output: The most feasible velocity $\mathbf{v}_i'$ of UAV i

```
for j in N do
    RVO_j^i ← OperationRVO3D(p_i, v_i, p_j, v_j) ;
end
for j in N do
    AV_j^i ← U_j^i(v_j, v_i) − RVO_j^i(v_j, v_i) ;
end
AV^i = ⋂_{j≠i} AV_j^i(v_j, v_i) ;
v_i^opt ← arg min_{v_i'∈AV^i}(‖v_i' − v_i^pref‖) ;
for v_i' in AV^i do
    OV_SCA^i ← {v_i' | ‖v_i' − v_i^opt‖ < Δv_opt} ;
end
v_i^new ← arg min_{v_i^cand∈OV_SCA^i}(θ(v_i^cand)) ;
return v_i' ;
```

B. The 3D Dubins Path with Collision Avoidance

In fact, the posture constraints are general requirements in motion planning for fixed-wing UAVs, and the 3D Dubins method is mainly exploited to solve this problem.

Váňa Petr et al. [8] addressed the shortest 3D Dubins curve with curvature bounds and pitch angle constraints, improving the calculation speed. We extend this work by integrating the RVO3D algorithm, in which the static and dynamic obstacles are investigated in the 3D Dubins method. The details of implementation are summarized as follows two points. Firstly, the UAV will recalculate the new 3D Dubins path if the current path is different from the old 3D Dubins path. Secondly, calculating the desired velocity in the RVO3D algorithm is replaced by calculating the 3D Dubin curve's adjacent nodes near the UAV's current position.

We can recognize collision-avoidance action in the first point with the Eqn. (15), where $\mathbf{v}_i^{pref}(t-1)$ is the desired velocity of any UAV i at the last moment, and $\mathbf{v}_i(t)$ is the current velocity of the UAV i. This Eqn. (15) follows the fact: For UAV i, it will execute the desired velocity when there are no obstacles. The second point can be formulated as follows. For UAV i, let $\mathbf{p}_i$ be the current position, $\mathbf{p}_i^{goal}$ be the goal position, and v_i^{pref} is the preferred scalar speed. We can calculate the desired velocity $\mathbf{v}_i^{pref}(t)$ at t moment using Eqn. (16) in the RVO3D algorithm. At the same time, we express the 3D Dubins path node at any time t as $D(t)$, where the distance between adjacent nodes indicates the sampling size of the 3D Dubins curve. In this case, the desired velocity $\mathbf{v}_i^{pref}(t)$ at t moment can be computed using Eqn. (17). In a word, the Eqn. (16) will be replaced with Eqn. (17) when the 3D Dubins method is introduced into the RVO3D algorithm.

$$\mathbf{v}_i^{pref}(t-1) = \mathbf{v}_i(t), \tag{15}$$

$$\mathbf{v}_i^{pref}(t) = v_i^{pref}\cdot(\mathbf{p}_i^{goal} - \mathbf{p}_i), \tag{16}$$

$$\mathbf{v}_i^{pref}(t) = v_i^{pref}\cdot(D(t+1) - \mathbf{p}_i). \tag{17}$$

The process of implementing posture constraints can be summarized in the following three steps.

1) First, we judge whether the $\mathbf{v}_i^{pref}(t-1)$ is parallel to $\mathbf{v}_i(t)$ and whether both vectors are in the same direction.
2) If both conditions above are true, UAV i will calculate the desired velocity with the old 3D Dubins curve. Otherwise, UAV i will update the 3D Dubins curve to calculate the desired velocity.
3) Meanwhile, the 3D Dubins curve will not be updated when the distance between the current position and

goal position is less than d, where d is $3.0 \cdot \rho_{min}$ (ρ_{min} is the minimum turning radius of the UAV).

V. EXPERIMENTAL RESULTS

In this section, we conducted three simulation experiments to validate the effectiveness of the proposed algorithm on the AirSim platform. Then, we compared our method with other SOTA (state-of-the-art) methods. The results of the experiments show the validity of the proposed method.

A. Simulation Setup

The parameters of algorithms were set as follows. The preferred and maximum speeds were set as $v^{pref} = 1$ m/s and $v^{max} = 1$ m/s, respectively. The maximum observation range was set as $d^{max} = 10$ m. The maximum number of neighbors was set as $N^{max} = 15$. The time horizon and the time step were set as $\tau = 10$ s and $\Delta t = 0.1$ s, respectively. All simulations were conducted on an Intel(R) Core(TM) i7-12700K 3.61 GHz NVIDIA GeForce RTX 3090 device.

Five metrics for quantitative evaluation were listed as follows.

- Success Rate (SR): the ratio of the number of UAVs successfully reaching their goals in a specific time limit without any collision to the total number of UAVs.
- Extra Time (ET): the average extra travel time that per UAV spends on its actual traveling time compared with going straight toward its goal.
- Extra Distance (ED): the average extra travel distance per UAV spends on its actual trajectories length compared with going straight toward its goal.
- Average Speed (AS): the average speed of all UAVs during operation.
- Computational Cost (CC): the average computational time per UAV spends on the time of solving new velocity.

B. Simulation Experiments

In this part, three simulation experiments are conducted on the AirSim platform [9]. The maximum yaw rate and pitch rate are used to evaluate the performance of trajectories.

We experimented with simulating an air patrol task with our SCA algorithm in the first simulation. Here, fourteen fixed-wing UAVs are deployed uniformly on a circle at a specific altitude whose radius is 10 m to execute the air patrol task. They are required to turn around automatically back to the start position once the patrol task is completed. The experiment results are presented in Fig. 4. These figures show that all UAVs completed the air patrol task and returned to the start position with the initial heading angle. Meanwhile, these metrics in Fig. 4 show that the smoothness of all UAVs' trajectories is excellent, demonstrating our proposed method's practical potential in the real world. Note that the ideal metrics values are computed by the Dubins trajectories in ideal conditions, the same as the following simulations.

The second simulation was conducted to simulate the process of take-off and landing with the SCA algorithm. Here, eight fixed-wing UAVs are located uniformly on a

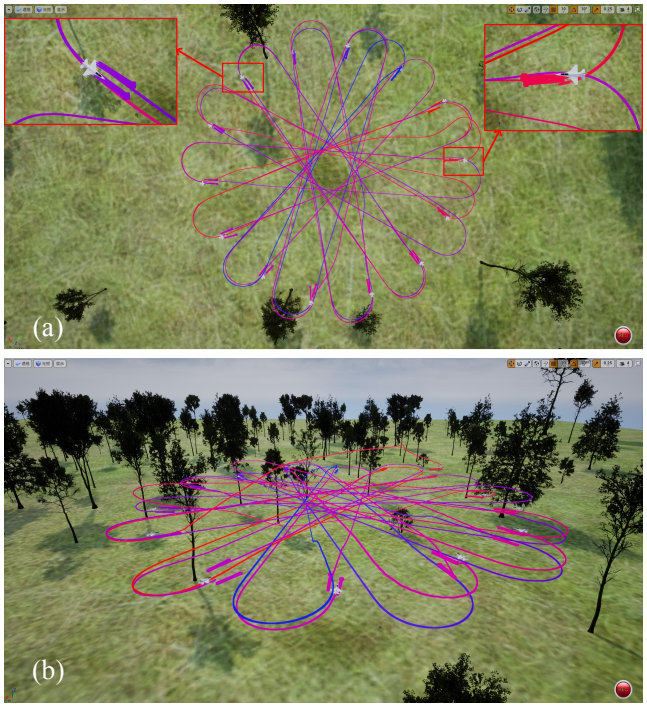

Fig. 4. Simulation results of air patrol task. Maximum yaw rate: 0.566 rad; Maximum pitch rate: 0.547 rad; ET: 11.0 s (ideally: 8.8 s); ED: 10.914 m (ideally: 8.8 m); AS: 0.998 m/s (ideally: 1.0 m/s); CC: 19.987 ms. (a) Top view. (b) 3D view.

circle at a specific altitude, and another eight UAVs are similar at another specific altitude. Both circles' radius is 4 m, and their height difference is 10 m. Now, all two UAVs in a vertical position will exchange their position for simulating the take-off and landing, respectively. Note that obstacle-free and obstacle cases are considered in this experiment. The simulation results are presented in Fig. 5. From these figures, it can be observed that all UAVs successfully finished the take-off and landing process in both cases. In particular, from these metrics in Fig. 5, it also can be observed that the pitch angle constraints are satisfied during all the take-off and landing processes. The simulation also verified the potential practice of our proposed method.

The third simulation was conducted with our SCA method for simulating a low-altitude search task. Around the outside of a jungle area whose radius is 10 m, sixteen fixed-wing UAVs travel through the jungle area at a lower altitude and exchange their position and heading angles with each other. Note that all the trunks of trees expressed by multiple voxels are regarded as obstacles (red cube in Fig. 1), and the leaves are not regarded as obstacles. The experiment results are shown in Fig. 1. From these figures, it can be found that all UAVs successfully reached their goal positions and heading angles. These metrics in Fig. 1 show that the smoothness of all UAVs' trajectories is excellent and is possible for UAVs to follow in the real world.

C. Performance Evaluation

For fairness, the S-RVO3D (Ours.) was compared with the RVO3D as well as the ORCA3D. Note that we do not

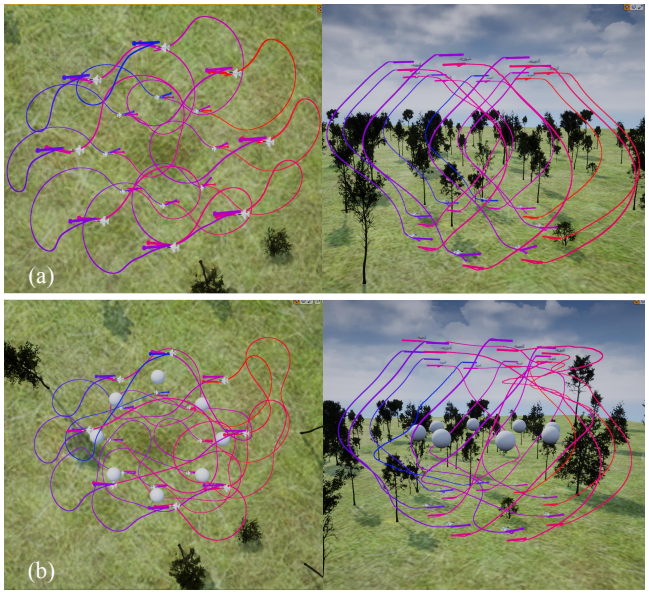

Fig. 5. Simulation results of take-off and landing. (a) obstacle-free case: top view on the left and 3D view on the right. Maximum yaw rate: 0.694 rad; Maximum pitch rate: 0.679 rad; ET: 9.1 s (ideally: 5.6 s); ED: 6.489 m (ideally: 5.6 m); AS: 0.863 m/s (ideally: 1.0 m/s); CC: 15.966 ms. (b) obstacle case: top view on the left and 3D view on the right. Maximum yaw rate: 0.916 rad; Maximum pitch rate: 0.783 rad; ET: 15.4 s (ideally: 5.6 s); ED: 10.129 m (ideally: 5.6 m); AS: 0.685 m/s (ideally: 1.0 m/s); CC: 14.792 ms.

TABLE I

COMPARED THE PROPOSED METHOD WITH OTHER METHODS.

Scenarios	Metric	RVO3D	ORCA3D	S-RVO3D
Circle	Success Rate	0.91	0.99	**1.00**
	Extra Time	7.4	10.5	**5.5**
	Extra Distance	2.741	2.943	**2.298**
	Average Speed	0.893	0.837	**0.924**
	Computational Cost	20.655	**9.521**	19.228
Ball	Success Rate	0.86	0.93	**1.00**
	Extra Time	8.3	6.5	**5.5**
	Extra Distance	3.524	2.454	**2.368**
	Average Speed	0.917	0.927	**0.943**
	Computational Cost	20.838	**17.461**	19.845
Random	Success Rate	0.99	**1.00**	**1.00**
	Extra Time	**0.2**	1.2	0.3
	Extra Distance	**0.118**	0.319	0.144
	Average Speed	**0.996**	0.961	0.995
	Computational Cost	25.565	**8.461**	25.872

compare our method with the above variants of RVO and ORCA as they are developed only in a two-dimensional workspace. Three scenarios are considered, including circle (radius is 18 m), ball (radius is 25 m), and random (space is 30 m $\times$ 30 m $\times$ 30 m, where 10 times independent experiments would give an average result rather than results from only one test.) scenarios. The number of UAVs is 100. Table I presents the comparison results, and the best results are highlighted in bold. It can be seen that our method yields the best performance in most cases except for the Computational Cost. In the metric of computational cost, the best results were yielded by the ORCA3D in all cases due to the introduction of the linear program, but the S-RVO3D can still provide a real-time response. In the random benchmark, as the S-RVO3D considers a safer strategy, the RVO3D method yields the best performance in three metrics except for Success Rate and Computational Cost.

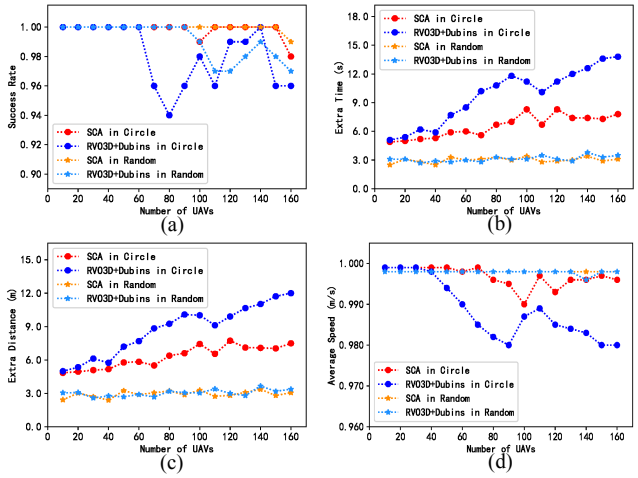

Fig. 6. The performance evaluated for the SCA and RVO3D+Dubins methods on the circle and random scenarios.

Considering the posture constraints, the SCA was compared with the RVO3D+Dubins in a circle (radius is 40 m) and random (space is 80 m $\times$ 80 m $\times$ 10 m, where 10 times independent experiments would give an average result rather than results from only one test.) scenarios. Fig. 6 presents the experimental results. In Fig. 6, it can be observed that our SCA method yields the best performance in almost all cases. In the circle scenario, although the Success Rate under the SCA is higher than its competitor, it drops to 99% when the number of UAVs is 100. It should be that the Dubins path cannot be computed when UAVs reach the near goal position and encounter other UAVs, resulting in the stuck. In the random scenario, except for the Success Rate, both methods yield comparable performances as the complexity of the random scenario is less than the circle scenario, resulting in the path distance of both methods being close to the optimal path distance.

Overall, all experimental results demonstrated the effectiveness and the practical potential of our proposed method.

VI. CONCLUSIONS

This paper proposes a novel SCA algorithm for fixed-wing UAVs' motion planning in a three-dimensional workspace. The posture constraints of fixed-wing UAVs were defined as the minimum turning radius and pitch angle constraints. The extensive simulations show our proposed method's effectiveness and practical potential. On the other hand, the proposed algorithm only regards the UAVs as a sphere, and introducing the 3D Dubins method also affects the computational power. In the future, we will mainly improve these shortcomings and consider the more practical constraints, applying the SCA algorithm to fixed-wing UAVs in the real world.

REFERENCES

[1] K. Cesare, R. Skeele, S.-H. Yoo, Y. Zhang, and G. Hollinger, "Multi-uav exploration with limited communication and battery," in *2015 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2015, pp. 2230–2235.

[2] Y. Wang, Y. Yue, M. Shan, L. He, and D. Wang, "Formation reconstruction and trajectory replanning for multi-uav patrol," *IEEE/ASME Transactions on Mechatronics*, vol. 26, no. 2, pp. 719–729, 2021.

[3] X. Wang, S. Baldi, X. Feng, C. Wu, H. Xie, and B. De Schutter, "A fixed-wing uav formation algorithm based on vector field guidance," *IEEE Transactions on Automation Science and Engineering*, 2022.

[4] E. T. Alotaibi, S. S. Alqefari, and A. Koubaa, "Lsar: Multi-uav collaboration for search and rescue missions," *IEEE Access*, vol. 7, pp. 55 817–55 832, 2019.

[5] J. Van den Berg, M. Lin, and D. Manocha, "Reciprocal velocity obstacles for real-time multi-agent navigation," in *2008 IEEE International Conference on Robotics and Automation*. IEEE, 2008, pp. 1928–1935.

[6] J. A. Douthwaite, S. Zhao, and L. S. Mihaylova, "Velocity obstacle approaches for multi-agent collision avoidance," *Unmanned Systems*, vol. 7, no. 01, pp. 55–64, 2019.

[7] G. Xu, Y. Chen, J. Cao, D. Zhu, W. Liu, and Y. Liu, "Multivehicle motion planning with posture constraints in real world," *IEEE/ASME Transactions on Mechatronics*, 2022.

[8] P. Váňa, A. A. Neto, J. Faigl, and D. G. Macharet, "Minimal 3d dubins path with bounded curvature and pitch angle," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 8497–8503.

[9] S. Shah, D. Dey, C. Lovett, and A. Kapoor, "Airsim: High-fidelity visual and physical simulation for autonomous vehicles," in *Field and Service Robotics: Results of the 11th International Conference*. Springer, 2018, pp. 621–635.

[10] R. J. Luna and K. E. Bekris, "Push and swap: Fast cooperative path-finding with completeness guarantees," in *Twenty-Second International Joint Conference on Artificial Intelligence*, 2011.

[11] G. Sharon, R. Stern, A. Felner, and N. R. Sturtevant, "Conflict-based search for optimal multi-agent pathfinding," *Artificial Intelligence*, vol. 219, pp. 40–66, 2015.

[12] P. E. Hart, N. J. Nilsson, and B. Raphael, "A formal basis for the heuristic determination of minimum cost paths," *IEEE transactions on Systems Science and Cybernetics*, vol. 4, no. 2, pp. 100–107, 1968.

[13] S. Kamio and H. Iba, "Random sampling algorithm for multi-agent cooperation planning," in *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2005, pp. 1265–1270.

[14] S. M. LaValle *et al.*, "Rapidly-exploring random trees: A new tool for path planning," 1998.

[15] K. Solovey, O. Salzman, and D. Halperin, "Finding a needle in an exponential haystack: Discrete rrt for exploration of implicit roadmaps in multi-robot motion planning," *The International Journal of Robotics Research*, vol. 35, no. 5, pp. 501–513, 2016.

[16] L. Wen, Y. Liu, and H. Li, "Cl-mapf: Multi-agent path finding for car-like robots with kinematic and spatiotemporal constraints," *Robotics and Autonomous Systems*, vol. 150, p. 103997, 2022.

[17] C. W. Warren, "Multiple robot path coordination using artificial potential fields," in *Proceedings., IEEE International Conference on Robotics and Automation*. IEEE, 1990, pp. 500–505.

[18] V. R. Desaraju and J. P. How, "Decentralized path planning for multi-agent teams in complex environments using rapidly-exploring random trees," in *2011 IEEE International Conference on Robotics and Automation*. IEEE, 2011, pp. 4956–4961.

[19] T. Fan, P. Long, W. Liu, and J. Pan, "Distributed multi-robot collision avoidance via deep reinforcement learning for navigation in complex scenarios," *The International Journal of Robotics Research*, vol. 39, no. 7, pp. 856–892, 2020.

[20] D. Wang, T. Fan, T. Han, and J. Pan, "A two-stage reinforcement learning approach for multi-uav collision avoidance under imperfect sensing," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 3098–3105, 2020.

[21] P. Fiorini and Z. Shiller, "Motion planning in dynamic environments using velocity obstacles," *The International Journal of Robotics Research*, vol. 17, no. 7, pp. 760–772, 1998.

[22] J. Van Den Berg, S. J. Guy, M. Lin, and D. Manocha, "Reciprocal n-body collision avoidance," in *Robotics research*. Springer, 2011, pp. 3–19.

[23] J. Snape, J. Van Den Berg, S. J. Guy, and D. Manocha, "The hybrid

reciprocal velocity obstacle," *IEEE Transactions on Robotics*, vol. 27, no. 4, pp. 696–706, 2011.
[24] F. Lamarche and S. Donikian, "Crowd of virtual humans: a new approach for real time navigation in complex and structured environments," in *Computer graphics forum*, vol. 23, no. 3. Wiley Online Library, 2004, pp. 509–518.
[25] D. Bareiss and J. Van den Berg, "Generalized reciprocal collision avoidance," *The International Journal of Robotics Research*, vol. 34, no. 12, pp. 1501–1514, 2015.
[26] R. Mao, H. Gao, and L. Guo, "A novel collision-free navigation approach for multiple nonholonomic robots based on orca and linear mpc," *Mathematical Problems in Engineering*, vol. 2020, 2020.
[27] J. Liu, W. Han, X. Wang, and J. Li, "Research on cooperative trajectory planning and tracking problem for multiple carrier aircraft on the deck," *IEEE Systems Journal*, vol. 14, no. 2, pp. 3027–3038, 2019.
[28] J. Snape and D. Manocha, "Navigating multiple simple-airplanes in 3d workspace," in *2010 IEEE International Conference on Robotics and Automation*. IEEE, 2010, pp. 3974–3980.
[29] C. Y. Tan, S. Huang, K. K. Tan, and R. S. H. Teo, "Three dimensional collision avoidance for multi unmanned aerial vehicles using velocity obstacle," *Journal of Intelligent & Robotic Systems*, vol. 97, no. 1, pp. 227–248, 2020.