ENGINEERING Artificial Intelligence
https://doi.org/10.2738/ENGAI.2026.0009

Original Research

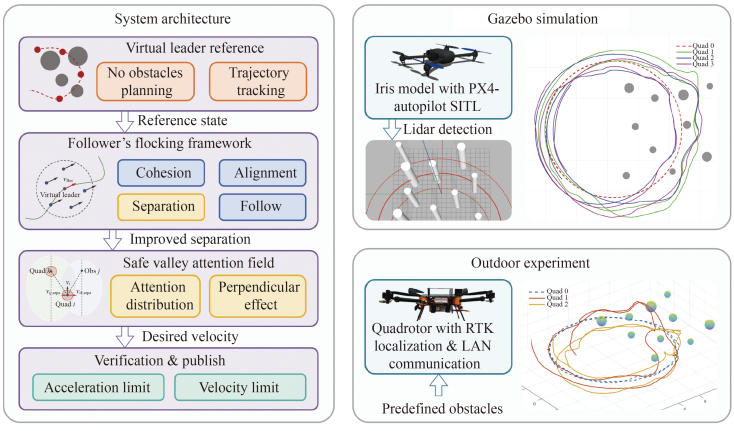
A Distributed Motion Planner for Cooperative Flight of Quadrotor Swarms via Flocking Behavior

Hongtao Liang[1,2], Yaonan Wang[1,2], Hang Zhong[1,2], VanCuong Pham[3], Hao Chang[1,2], Xin Chen[1,2], Hui Zhang[1,2], Hean Hua (✉)[1,2]

1 School of Artificial Intelligence and Robotics, Hunan University, Changsha 410082, China
2 National Engineering Research Center for Robot Visual Perception and Control Technology, Hunan University, Changsha 410082, China
3 HaNoi University of Industry, Hanoi 11914, Vietnam

© The Author(s) 2026. This article is published by Higher Education Press.

GRAPHICAL ABSTRACT



HIGHLIGHTS

- A distributed motion planner for quadrotor swarms is formulated in the flocking behavior framework, which organizes cohesion, alignment, separation and follow-motion into one local velocity command.
- A safe valley attention field is proposed for the separation part, which converts the attention distribution of a human driver into a geometric boundary that reacts before the collision boundary is reached.
- A Gazebo simulation and an outdoor experiment through ten obstacles both show cooperative flight with clearances kept above the collision thresholds.

ABSTRACT In this paper, a distributed cooperative motion planning method is proposed for quadrotor swarm via flocking behavior framework, which achieves cooperative flight with maintained separation in the considered obstacle areas. The motion planning method consists of 4 parts: cohesion, alignment, separation and follow-motion. Firstly, the cohesion part is designed to gather quadrotors into a proper swarm size. Besides, the alignment part is formulated to achieve velocity consensus among neighboring quadrotors, which is based on the constant acceleration braking model. Then, the safe valley attention field is proposed in the separation part to achieve proactive separation in obstacle environments, which is inspired by the distribution of human attention when driving. In addition, a virtual quadrotor is used to generate the follow-motion part. Finally, a swarm with 4 Iris-quadrotor models is constructed in Gazebo simulation platform, to validate the effectiveness of the proposed method. For practical application, an outdoor experiment with 3 quadrotors is also conducted, in which the swarm follows the virtual leader through ten predefined virtual spherical obstacles; this experiment validates distributed execution and trajectory tracking with known obstacle geometry rather than onboard obstacle perception. The results show that the proposed method achieves cooperative flight with maintained separation in both the simulated and the outdoor scenarios considered here.

Received 3 Aug 2026; accepted 22 Sep 2026

E-mail: huahean@hnu.edu.cn (Hua H.).

KEYWORDS Multiple quadrotors; Motion planning; Cooperative flight; Distributed system; Flocking framework; Safe valley potential field

1 Introduction

Recently, quadrotors have been widely applied in various tasks including express delivery [1,2], infrastructure inspection [3,4], agricultural monitoring [5] and aerial photography [6]. Compared to a single quadrotor, a swarm has attracted the concentration of researchers due to their potential advantages on efficiency and flexibility in the above-mentioned large-scale and complex tasks [7,8]. Cooperative flight is one of the key research topics on quadrotor swarms. A cooperative flight method not only enhances the autonomy and robustness of the swarm system, but also significantly improves its adaptability and mission success rate even in complex environments. To achieve this, various methods have been researched and proposed. Most methods are based on distributed architecture, for the superior scalability, fault tolerance, reduced latency and better communication efficiency [9,10]. These methods can be summarized into two categories. For the first category, each quadrotor is assigned with its own target, so it is significant to achieve a suitable target assignment for cooperative flight. While the second category is assignment-free for quadrotors, which divides the cooperative flight into unified-motion layer and adaptive-motion layer.

Several methods have been published to achieve cooperative flight of distributed quadrotor swarm, which are classified as the first category [11–16]. In Ref. [11], a collision-free trajectory planning method is proposed for quadrotor swarms, which combines the visual vector field function with the modified Optimal Reciprocal Collision Avoidance (ORCA) algorithm. In Ref. [12], a gradient-based path planner is proposed for swarm flight, which assigns different passage priorities for quadrotors to avoid collision and improve efficiency. In Ref. [13], Senthil et al. present the Chance-Constrained Optimization (CCO) method, which achieves collision avoidance swarms flight under uncertain state estimation to neighbor quadrotors. In Ref. [16], an innovative approach is proposed based on deep generative model, to generate passable paths for flight shows, where each quadrotor navigates to its target with reactive collision-avoiding algorithm. However, it is usually a complex multi-constraint optimization problem to achieve an effective target assignment in obstacle environments, which is computationally expensive and difficult to solve in real time. Therefore, the methods above need to make great effort in assigning targets or priorities for quadrotors, which fundamentally limits their scalability and adaptability in cooperative flight.

For the foregoing reason, studies on assignment-free cooperative flight have been receiving much attention, which can be classified as the second category [17–21]. For instance, in Ref. [17], a formation motion planning method is proposed to avoid obstacles based on the affine transformation of the swarm shape, without requiring individual replanning for each quadrotor. In Ref. [19], a method is proposed based on the minimizing-energy virtual tube to generate a safe and smooth channel among obstacles, where a quadrotor swarm can flow cooperatively like water flowing in pipe. In Ref. [20], a swarm level collision-avoidance method is published, which introduces an adaption mechanism to divide the swarm adaptively, to achieve cooperative flight. In Ref. [21], a shape assembly method for swarms is proposed based on the idea of mean-shift exploration, which enables the swarm to adapt to different shape of unoccupied regions while passing through obstacle environments. However, the reach-and-wander motion pattern is usually used for collision avoidance in these methods, which may result in the entire swarm being immobilized when certain quadrotors get stuck in confined spaces between obstacles. Therefore, the possibility of getting stuck becomes a critical limitation of the assignment-free methods, particularly in obstacle environments.

Inspired by the attention distribution of drivers in crowded streets, the safe valley attention field is designed to form a novel separation algorithm, which enables proactive and smooth collision avoidance in swarm flight. And then, a distributed motion planning method is proposed for cooperative quadrotor swarm flight in obstacle environments via the flocking behavior framework, which incorporates the separation algorithm to achieve proactive and smooth collision avoidance. Finally, the planning method is respectively validated on a simulation platform and an outdoor experiment platform. The main contributions are summarized as follows:

- A distributed motion planning method is proposed for quadrotor swarms via the flocking behavior framework, which achieves cooperative flight in obstacle environments.
- A novel separation algorithm is presented to achieve proactive and smooth collision avoidance in swarm flight, which is based on the safe valley attention field, inspired by the attention distribution of human driving.

The rest of this paper is introduced as follows: in Section 2, the quadrotor model and the flocking behavior framework are introduced, which divide the cooperative flight problem into three adaptive-motion parts and a follow-motion part. Section 3 presents the proposed method with its 4 parts detailed in sequence, where the safe valley attention field is proposed in Subsection 3.3 to form the separation part. Section 4 presents the system architecture, the simulation and the outdoor experiment of the proposed distributed motion planning method, together with the log processing used for the results and a discussion of the scope and limitations. Section 5 summarizes the main works of the paper.

2 Preliminaries and Problem Formulation

In this section, the model of a quadrotor is defined as a ball with multi-layers. Then, the flocking behavior framework is introduced, which is used to formulate the constitution of the motion planning method.

2.1 Quadrotor model

Each quadrotor of the swarm is regarded as a ball shown in Fig. 1. Here, r_{safe} denotes the collision radius of quadrotors. r_{free} is the free-distance, which means the minimum distance at which no separation effect is needed. r_{sen} represents the effective perception distance; the three quantities above are distances expressed in meters. Therefore, quadrotor i is regarded as a ball centered on r_i, with r_{safe} as the radius. The separation term of quadrotor i becomes active when the center distance to an object is less than the free-distance $r_{ij,\text{free}}$, whereas a collision is considered to occur when this distance falls below the collision distance r_{off}. The quadrotor can't find obstacles at a distance greater than r_{sen} from itself, so that distant obstacles have no interaction and effect on it. The clearance between quadrotor i and quadrotor j is defined as $d_{ij} = \|r_i - r_j\| - 2r_{\text{safe}}$, and the clearance between quadrotor i and an obstacle centered at c_o with radius r_o is $d_{io} = \|r_i - c_o\| - r_o$. Therefore, a collision occurs when the corresponding clearance becomes negative, that is, when the center distance is less than the collision distance r_{off}, which equals $2r_{\text{safe}}$ between a pair of quadrotors and r_{safe} between a quadrotor and an obstacle point ($r_{\text{safe}} + r_o$ for a spherical obstacle of radius r_o).

In addition to the proposed ball model of quadrotors, it is essential to incorporate velocity and acceleration constraints in the model, to keep the commanded motions within the platform envelope. For every quadrotor, the maximum velocity and maximum acceleration are

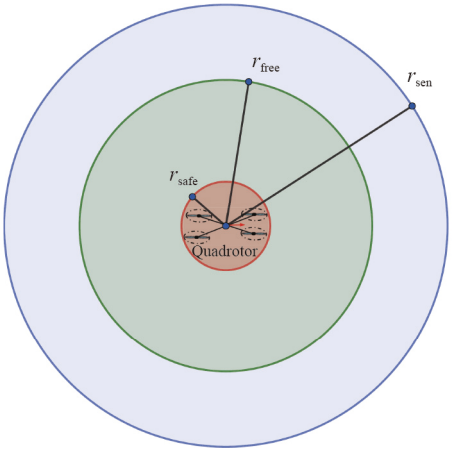


Fig. 1 Multi-layer ball model of a quadrotor.

respectively defined as $v_{\max}$ and $a_{\max}$, given in meters per second and meters per second squared. Hence, the desired velocity $v_{i,\text{des}}$ is constrained within the set below:

$$v_{i,\text{des}} \in \mathcal{V}_{i,\text{des}},$$
$$\mathcal{V}_{i,\text{des}} = \left\{ v \in \mathbb{R}^3 \,\middle|\, \begin{array}{l} v = v_i + \widehat{u} \cdot a_{\max}\delta t, \\ \widehat{u} \in \mathbb{S}^2, \\ \|v\| \leqslant v_{\max} \end{array} \right\}, \tag{1}$$

where v_i is the current velocity of the quadrotor, δt denotes the time step of the planning method in seconds, $\mathbb{S}^2 = \{u \in \mathbb{R}^3 \mid \|u\| = 1\}$ denotes the unit sphere in $\mathbb{R}^3$. For a bound $\rho > 0$, the saturation operator is defined as

$$\text{sat}_\rho(x) = x \min\left\{1, \frac{\rho}{\|x\|}\right\}, \text{ if } x \neq 0, \qquad \text{sat}_\rho(0) = 0, \tag{2}$$

so that a zero vector is mapped to a zero vector rather than to an undefined unit vector. The two constraints of Eq. (1) are enforced through this operator, the velocity increment being limited first and the speed afterwards, as detailed in Subsection 3.4. The bounds of Eq. (1) are therefore applied through the saturation operator, so that the commanded velocity respects the speed and increment limits of the platform. Throughout this paper, positions and distances are expressed in meters, velocities in meters per second, accelerations in meters per second squared, and all gains are dimensionless.

2.2 Flocking swarm framework

According to the flocking behavior framework introduced in Ref. [22], a distributed and cooperative swarm is established and maintained through 3 rules on every member: cohesion, alignment, and separation. The cohesion rule drives individuals toward the swarm center, which promotes the swarm aggregation. The alignment rule encourages to reconcile the velocities of nearby neighbors, promoting consistent motion across the

swarm. The separation rule generates repulsive forces to avoid collisions, both among swarm members and with external obstacles, thereby providing the separation mechanism studied in this paper.

Based on the framework, the cooperative flight problem is formulated as a combination of 4 motion parts: the cohesion, alignment and separation parts to keep the swarm and adapt the environment, supplemented by a follow-motion part to track the global path. To proceed, a swarm is established consisting of multiple quadrotors, where every physical quadrotor is guided by the proposed planning method to achieve cooperative flight. And a virtual leader quadrotor is introduced, which ideally follows the predefined global path and broadcasts its velocity as a follow-motion reference. In particular, the desired velocity for quadrotor i is

$$v_{i,\mathrm{des}} = v_{i,\mathrm{cohe}} + v_{i,\mathrm{aligned}} + v_{i,\mathrm{sepa}} + v_{\mathrm{floc}}, \tag{3}$$

where $v_{i,\mathrm{cohe}}$, $v_{i,\mathrm{aligned}}$, $v_{i,\mathrm{sepa}}$ and v_{floc} represent the desired velocity for cohesion, alignment, separation, and follow-motion respectively, as shown in Fig. 2. All four terms are desired-velocity contributions expressed in m/s, and the gains that scale them are dimensionless; the contributions of all eligible neighbors and objects are accumulated first, so that the intermediate sum may exceed $v_{\max}$, and the limits are enforced once at the end, as detailed in Subsection 3.4. In particular, $v_{i,\mathrm{cohe}}$ acts in the gray region in Fig. 2A, and it gathers the quadrotor into the swarm shape. $v_{i,\mathrm{aligned}}$ is used to reconcile the velocities of nearby quadrotors with an orange implicit effective range in Fig. 2B. $v_{i,\mathrm{sepa}}$ avoids collision within the green attention region in Fig. 2C while passing through obstacles. And then, all quadrotors in the white sphere like Fig. 2D are requested to move on the base velocity v_{floc} to follow the virtual leader. The follow-motion term is a measured reference velocity, namely the horizontal velocity of the virtual leader together with an altitude regulation term, and it carries no gain.

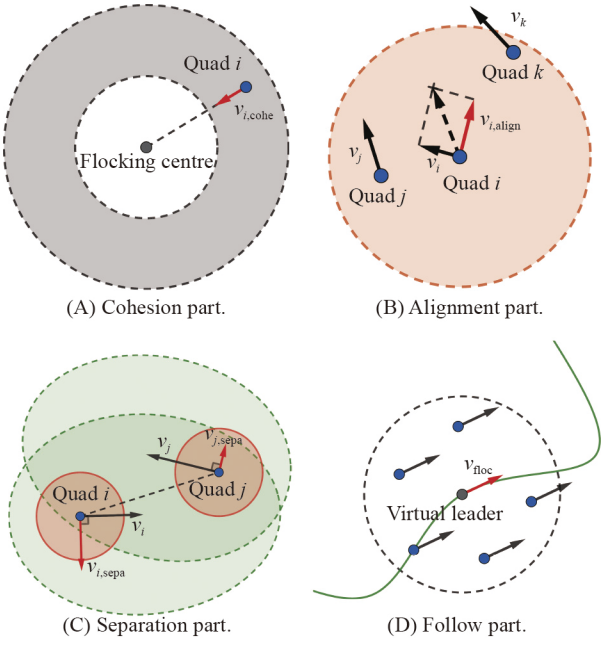


Fig. 2 The 4 motion parts that constitute the cooperative flight of quadrotor swarms.

3 Methods

This section details the proposed motion planning method to achieve cooperative flight of quadrotor swarms in obstacle environments. It is organized into 4 subsections that respectively corresponding to the cohesion, alignment, separation, and follow-motion parts.

3.1 Cohesion

The cohesion region is defined as a sphere centered on the virtual leader r_0. And then, the velocity algorithm $v_{i,\mathrm{cohe}}$ is designed for guiding quadrotor i to the desired region. That is, $v_{i,\mathrm{cohe}}$ guides quadrotor i toward the center when it's out of the region, while do nothing within the region. Besides an inner border to accommodate the swarm, an outer border is considered to confine the effect of cohesion, so that the swarm does not affect distant and isolated quadrotors. Since r_{i0} is strictly positive within the active interval $r_{\mathrm{in}} < r_{i0} < r_{\mathrm{out}}$, the unit vector used in the cohesion term is always well defined. In particular,

$$v_{i,\mathrm{cohe}} = \begin{cases} 0, & \text{if } r_{i0} \leqslant r_{\mathrm{in}} \\ -p_{\mathrm{cohe}} \sqrt{\dfrac{r_{i0} - r_{\mathrm{in}}}{r_{\mathrm{out}} - r_{\mathrm{in}}}} \cdot \widehat{r_{i0}}, & \text{if } r_{\mathrm{in}} < r_{i0} < r_{\mathrm{out}} \\ 0, & \text{otherwise} \end{cases}. \tag{4}$$

Here, $r_{i0} = \|r_i - r_0\|$ denotes the distance between quadrotor i and the virtual leader (quadrotor 0 in other words), $\widehat{r_{i0}} = \dfrac{r_i - r_0}{r_{i0}}$ is the unit vector that represents the direction from the leader to the quadrotor. $p_{\mathrm{cohe}} > 0$ is the dimensionless gain of the cohesion; the square-root factor equals zero at $r_{i0} = r_{\mathrm{in}}$ and one at $r_{i0} = r_{\mathrm{out}}$, so that $\|v_{i,\mathrm{cohe}}\| \leqslant p_{\mathrm{cohe}}$. r_{in} and r_{out} are respectively the inner and outer radii of the cohesion-effective region.

Every quadrotor of the swarm will be driven to gather into the cohesion region on radius r_{in}. So the radius should be specified to accommodate all the quadrotors. To assemble the swarm simply and regularly, the Simple Cubic Packing in Crystallography is referenced as a formation to estimate the proper radius, as shown in Fig. 3. In the figure, every quadrotor is assigned a spherical

space of radius r_{pack}. The figure shows a swarm of 27 quadrotors which is arranged into a Simple Cubic. In other words, the cohesion region is voxelated as the Simple Cubic Packing of all quadrotors. Radius of the sphere is calculated as below:

$$\begin{aligned} r_{\text{in}} &= r_{\text{pack}}\left(\frac{n}{\Phi_{\text{sc}}}\right)^{\frac{1}{3}},\ n\in\mathbb{N}, \\ r_{\text{pack}} &\geqslant \frac{1}{2}\left(r_{ij,\text{free}}\Big|_{e_i=0}^{j\text{ is a quadrotor}}\right)=\frac{1}{2}(b+2r_{\text{safe}}). \end{aligned} \tag{5}$$

Where n is the number of quadrotors of the swarm (the virtual leader is not included), $\mathbb{N}$ represents the set of natural numbers. r_{pack} denotes the packing radius, which allows every quadrotor to hover without separation effect. $\Phi_{\text{sc}}=\dfrac{\pi}{6}$ is the packing efficiency of the Simple Cubic Packing on space, approximately 52%. $r_{ij,\text{ free}}\Big|_{e_i=0}^{j\text{ is a quadrotor}}$ means the free-distance between two hovering neighboring quadrotors.

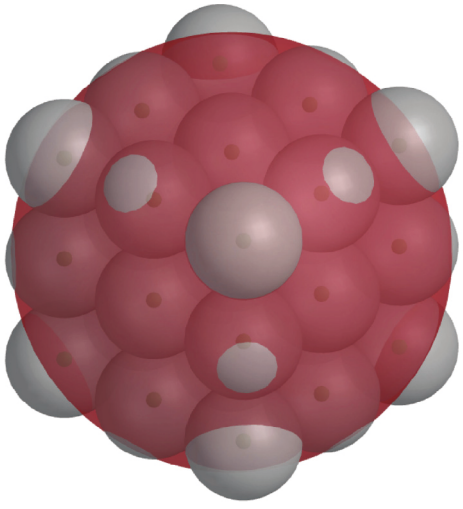


Fig. 3 Gather quadrotors by the Simple Cubic Packing.

3.2 Alignment

The alignment part is designed to achieve velocity consensus among neighboring quadrotors, promoting cooperative swarm behavior. For a pair of neighboring quadrotors i and j, it is implemented as a motion algorithm which limits the relative velocity within a range, as follow:

$$v_{ij,\text{aligned}}=\begin{cases}0, & \text{if } v_{ij}\leqslant v_{ij,\max}\\ -p_{\text{aligned}}\dfrac{v_{ij}-v_{ij,\max}}{2v_{\max}-v_{ij,\max}}\cdot\widehat{v_{ij}}, & \text{otherwise}\end{cases}. \tag{6}$$

Here, $v_{ij,\text{aligned}}$ denotes the desired alignment velocity for quadrotor i, which is influenced by quadrotor j. The relative velocity is defined as $v_{ij}=v_i-v_j$, with the norm $v_{ij}=\|v_{ij}\|$ being the value of the current velocity difference between the two quadrotors, and $\widehat{v_{ij}}=v_{ij}/v_{ij}$ the corresponding unit vector, which is well defined whenever $v_{ij}>0$, that is, whenever the alignment term is active. $p_{\text{aligned}}>0$ is the dimensionless gain of the alignment effect, and the sign that reduces the velocity difference is written explicitly in the vector expression above. $v_{ij,\max}$ is the maximum acceptable velocity difference at distance r_{ij}. $2v_{\max}$ is the maximum velocity difference between the two quadrotors. Therefore, the desired alignment velocity for quadrotor i can be calculated as the sum of alignment effect from other quadrotors: $v_{i,\text{aligned}}=\sum_{j\neq i}v_{ij,\text{aligned}}$.

A model of the limit braking curve is proposed to formulate $v_{ij,\max}$, which considers the maximum acceleration of quadrotors as an isotropic constant. The limit braking curve is identified as the function below:

$$(v_{\text{acp,max}})^2=2a_{\max}\cdot s_{\text{brk,min}},\ s_{\text{brk,min}}\geqslant 0, \tag{7}$$

where $v_{\text{acp,max}}$ is the maximum acceptable value between two quadrotors, $s_{\text{brk,min}}$ is the minimum braking distance in need, $a_{\max}$ is half of the maximum relative acceleration of the two quadrotors. Although a larger acceleration $2a_{\max}$ can be achieved in some cases, it is considered to cause an unreachable combined acceleration probably, so that the distance is too short to brake. Besides the limit braking curve, a minimum velocity difference is permitted over short distances, which enables rapid separation. So, the limitation of relative velocity is formulated as follow:

$$V_{ij}=\max\left\{(v_{\text{res}})^2,\ 2a_{\max}\cdot(r_{ij}-r_{\text{res}})\right\}, \tag{8}$$

$$v_{ij,\max}=\begin{cases}0, & \text{if } r_{ij}\leqslant 2r_{\text{safe}}\\ v_{\text{res}}, & \text{if } 2r_{\text{safe}}<r_{ij}\leqslant r_{\text{res}}\\ \sqrt{V_{ij}}, & \text{if } r_{\text{res}}<r_{ij}<r_{\text{aligned}}\\ 2v_{\max}, & \text{otherwise}\end{cases}. \tag{9}$$

Here, v_{res} is the permitted minimum velocity, $r_{ij}=\|r_i-r_j\|$ is the relative distance between quadrotor i and j, r_{res} is the residual distance at which the minimum permitted relative velocity v_{res} applies. r_{aligned} is an implicit border, where the alignment algorithm is effective. It is calculated as follow:

$$r_{\text{aligned}}=\frac{(2v_{\max})^2}{2a_{\max}}+2r_{\text{safe}}+r_{\text{res}}, \tag{10}$$

so that the limitation does not affect distant quadrotors farther away than r_{aligned}.

The braking curve above is a local design rule, and its

validity relies on the following assumptions. (1) Bounded relative deceleration: the closing motion can be reduced by at most $a_{\max}$ along the line of sight, with $a_{\max}$ taken as half of the maximum relative acceleration of the two quadrotors; this is a conservative bound that is assumed to hold throughout the flight. (2) Sampling: the terms of the method are re-evaluated at every control step of duration δt, and no sample is assumed to be lost. (3) Sensing and communication: the states of the neighbors and of the leader are assumed to be available at the update instant, with the GPS-time clock offset corrected as described in Subsection 4.1; the delay, the packet loss and the jitter of the link were not measured in this work, so this assumption remains unproved. (4) Tracking: the velocity setpoint is assumed to be tracked by the onboard controller within one control step; the trajectories of Section 4 support this approximately in the tested operating range, while the tracking error was not quantified. (5) Neighbor behavior: the closing motion of a pair is bounded by the same rule, and no adversarial behavior of a neighbor is assumed. Accordingly, the braking curve provides a local, velocity-dependent separation geometry rather than a formal proof of collision avoidance, and the safety margin is supported by the experiments reported in Section 4 and discussed in Subsection 4.5.

A visual representation of the velocity limitation can be seen in Fig. 4. In the figure, the X-axis is quantified with relative distance r_{ij}. The Y-axis is quantified with v_{ij}^2 to express the relationship more intuitively, so that the equal slopes mean a same relative acceleration. The green area is the set of all acceptable relative velocities on different distances. The blue polyline is the set border. In the case shown, the relative velocity v_{ij} is over the border, so the alignment equation will reduce it to $v_{ij,\max}$.

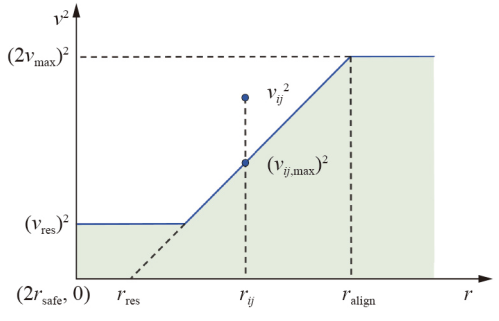


Fig. 4 The limitation of velocity difference at different distances.

Therefore, the alignment algorithm for quadrotor i is simplified as the sum of the individual alignment effect from neighboring quadrotors, given by:

$$\begin{aligned} v_{i,\text{aligned}} &= \sum_{j\in\mathcal{N}_{i,\text{aligned}}} v_{ij,\text{aligned}}, \\ \mathcal{N}_{i,\text{aligned}} &= \{j \mid j \neq i, r_{ij} < r_{\text{aligned}}\}. \end{aligned} \tag{11}$$

Here, $v_{i,\text{aligned}}$ is the desired alignment velocity for quadrotor i. $\mathcal{N}_{i,\text{aligned}}$ denotes the set of neighboring quadrotors within distance r_{aligned} around quadrotor i.

3.3 Separation based on safe valley attention field

A driver pays different attention to road conditions in different directions:

- The driver focuses most on the direction of velocity, with the farthest care distance.
- The secondary is attention to the left and right sides.
- The opposite direction receives the least attention with the shortest distance, which is not shorter than the distance covered by the car.
- The extent of attention difference also correlates with the car's speed—the faster it travels, the farther his or her focus extends to the heading direction.

Inspired by the attention distribution of drivers in crowded streets, the safe valley attention field is proposed to construct a novel separation algorithm, which guides the quadrotor to turn proactively around obstacles and other quadrotors. The case in Fig. 5 shows a typical scenario when quadrotor i is sandwiched between obstacle j and quadrotor k, while a swarm passing through obstacle environments. In the case, the separation affects only the perpendicular direction of quadrotor i's velocity, which forms a valley-shaped guidance field between obstacle point j and quadrotor k. For the design of separation region is inspired by the attention distribution of human driving, the field is named safe valley attention field.

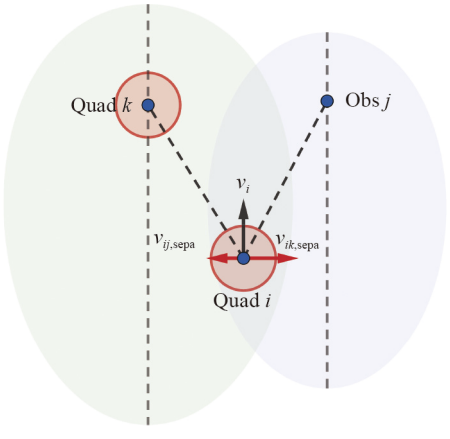


Fig. 5 The condition where a quadrotor fly in a safe valley.

Therefore, for quadrotor i, border of the attention region $r_{ij,\text{free}}$ is designed as an egg, which is the

superposition of a rotational ellipsoid and a ball, as seen in Fig. 6.

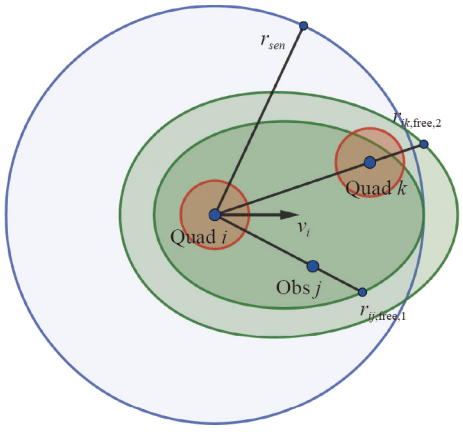


Fig. 6 The attention region of quadrotor i on velocity $\boldsymbol{v}_i|_{\|\boldsymbol{v}_i\|=v_{\max}}$.

In Fig. 6, the magnitude of quadrotor i's velocity equals to $v_{\max}$, so the smaller green border $r_{ij,\text{free},1}$ to obstacles reaches the blue sensing radius r_{sen} at $\theta_{ij}=0$. The other green border $r_{ik,\text{free},2}$ to other quadrotors can exceed the sensing boundary, because quadrotors are communicable in a larger distance. But the individuals farther away than the larger border are ignored in the algorithm.

By treating an obstacle as a cloud of points, the separation shape is formulated in a polar coordinate system as follow:

$$r_{ij,\text{free}}=\frac{b\sqrt{1-e_i^2}}{1-e_i\cos\theta_{ij}}+r_{ij,\text{off}}, \tag{12}$$

$$\cos\theta_{ij}=\frac{-v_i\cdot r_{ij}}{v_i r_{ij}},\ \theta_{ij}\in(-\pi,\pi], \tag{13}$$

$$r_{ij,\text{off}}=\begin{cases}r_{\text{safe}}, & \text{if } j \text{ is an obstacle point}\\ 2r_{\text{safe}}, & \text{if } j \text{ is a quadrotor}\end{cases}. \tag{14}$$

In the separation-region equation, the first part of $r_{ij,\text{free}}$ is a rotational ellipsoid whose focus is r_i. b is a specific semi-minor axis of the rotational ellipsoid. e_i is the eccentricity that determined by the value of quadrotor i's current velocity. The polar axis $\theta_{ij}=0$ is aligned with the direction of the velocity. $\boldsymbol{r}_{ij}=\boldsymbol{r}_i-\boldsymbol{r}_j$ is the relative position vector from quadrotor j to i. And the second part is the ball of radius $r_{ij,\text{off}}$, to keep the shortest distance from collision. With the focus at the quadrotor center and the polar axis along the velocity direction, the boundary of Eq. (12) is the offset of an ellipse whose semi-major axis, semi-minor axis, focal distance and semi-latus rectum are $a_i=b/\sqrt{1-e_i^2}$, $b_i=b$, $c_i=a_ie_i=be_i/\sqrt{1-e_i^2}$ and $\ell_i=a_i(1-e_i^2)=b\sqrt{1-e_i^2}$. The forward, rearward and lateral boundaries of the field are therefore $r_{ij,\text{free}}(0)=\ell_i/(1-e_i)+r_{ij,\text{off}}$, $r_{ij,\text{free}}(\pi)=\ell_i/(1+e_i)+r_{ij,\text{off}}=r_{ij,\text{free,min}}$ and $r_{ij,\text{free}}(\pm\pi/2)=\ell_i+r_{ij,\text{off}}$, as illustrated in Fig. 6. It is emphasized that $r_{ij,\text{free}}$ is the boundary where the separation term becomes active, whereas $r_{ij,\text{off}}$ is the collision boundary defined in Subsection 2.1. When an obstacle is represented by a sphere of radius r_o instead of a point, the offset radius becomes $r_{ij,\text{off}}=r_{\text{safe}}+r_o$; for a point obstacle $r_o=0$ holds and the expression reduces to the case given in Eq. (14).

In addition, e_i is considered to reach the maximum when quadrotor i is moving at $v_{\max}$, but the free-distance to an obstacle point should not be longer than the effective perception distance r_{sen}. In other words, $\forall e_i, r_{ij,\text{free}}|_{j\text{ is an obstacle point}}\leqslant r_{\text{sen}}$. Based on the limits, $e_{\max}$ and e_i can be calculated in closed form as follow:

$$\frac{b\sqrt{1-e_{\max}^2}}{1-e_{\max}\cos 0}+r_{\text{off}}=r_{\text{sen}}, \tag{15}$$

$$K=\left(\frac{r_{\text{sen}}-r_{\text{off}}}{b}\right)^2,\ e_{\max}=\frac{K-1}{K+1}, \tag{16}$$

$$e_i=\begin{cases}e_0, & \text{if } v_i<\varepsilon_v\\ e_{\max}\dfrac{v_i}{v_{\max}}, & \text{otherwise}\end{cases}, \tag{17}$$

where $r_{\text{off}}=r_{\text{safe}}$ for an obstacle point and $r_{\text{off}}=r_{\text{safe}}+r_o$ for an obstacle of radius r_o, as given in Eq. (14). The first equation, Eq. (15), is the sensing constraint evaluated at $\theta_{ij}=0$, and Eq. (16) is its closed-form solution, in which the semi-latus rectum is written as $\ell_i=b\sqrt{1-e_i^2}$. For the low-speed case, the polar axis is taken as the body forward axis (the first column of the rotation matrix reconstructed from the attitude quaternion), and the two constants are set to $\varepsilon_v=0.05v_{\max}$ and $e_0=0.05$ in all the experiments reported in Section 4. The eccentricity therefore satisfies $0<e_0\leqslant e_i\leqslant e_{\max}<1$, so that the field is defined for every state of the swarm. When the obstacle is a sphere, the forward boundary may exceed the sensing radius r_{sen} by r_o; such objects lie beyond the sensing range and are therefore not evaluated by the planner.

Therefore, the safe valley effect from object j to quadrotor i is formulated to guide proactive turning, which is perpendicular to the direction of quadrotor i's velocity and away from object j, as shown below:

$$\begin{aligned}&v_{ij,\text{sepa}}=p_{\text{sepa}}\sqrt{\frac{r_{ij,\text{free}}-r_{ij}}{r_{ij,\text{free,min}}-r_{ij,\text{off}}}}\cdot\widehat{r_{ij,\perp}},\ \text{if } r_{ij}<r_{ij,\text{free}},\\ &r_{ij,\text{free,min}}=r_{ij,\text{free}}|_{\theta_{ij}=\pi}=\frac{b\sqrt{1-e_i^2}}{1+e_i}+r_{ij,\text{off}}.\end{aligned} \tag{18}$$

Here, $p_{\text{sepa}}>0$ is the dimensionless gain of the

separation effect. To meet different objects including obstacle points and other quadrotors, the gain is divided into different values, which are $p_{\text{sepa,obs}} > 0$ and $p_{\text{sepa,quad}} > 0$ for obstacle points (or spherical obstacles) and for other quadrotors, respectively. $\widehat{r_{ij,\perp}}$ is a unit vector to represents the direction that perpendicular to the polar axis and away from the object, in particular,

$$\begin{aligned} \widehat{r_{ij,\perp}} &= \frac{r_{ij,\perp}}{\|r_{ij,\perp}\|}, \\ r_{ij,\perp} &= r_{ij} - \left(\frac{r_{ij}\cdot v_i}{v_i^2}\right) v_i. \end{aligned} \tag{19}$$

The escape direction is undefined when the object lies on the velocity axis, that is, when $\|r_{ij,\perp}\|$ vanishes; this includes the important case of an obstacle directly ahead of quadrotor i. Two thresholds are therefore introduced to make the field well defined. For $\|v_i\| < \varepsilon_v$, the polar axis is replaced by the body forward axis and the eccentricity is set to e_0, as given in Eq. (17). For $\|r_{ij,\perp}\| < \varepsilon_\perp$, the escape direction is replaced by the deterministic lateral direction $\widehat{u}_{ij} = (\widehat{v_i} \times \widehat{z})/\|\widehat{v_i} \times \widehat{z}\|$, or by $\widehat{u}_{ij} = (\widehat{v_i} \times \widehat{x})/\|\widehat{v_i} \times \widehat{x}\|$ if the former is also ill-conditioned, that is, for vertical motion. Objects whose centers coincide with the quadrotor, namely $r_{ij} < 10^{-3}$ m, are excluded from the sum. With these rules every term of the field is well defined, and an object directly ahead results in a lateral escape with a reproducible sign convention; the projection threshold is set to $\varepsilon_\perp = 10^{-3}$ m in the experiments reported in Section 4.

Hence, the separation algorithm for quadrotor i is sum of the safe valley effect from every object in the attention region:

$$\begin{aligned} v_{i,\text{sepa}} &= \sum_{j \in \mathcal{N}_{i,\text{sepa}}} v_{ij,\text{sepa}}, \\ \mathcal{N}_{i,\text{sepa}} &= \{ j \mid j \neq i, r_{ij} < r_{ij,\text{free}} \}. \end{aligned} \tag{20}$$

where, $v_{i,\text{sepa}}$ is the desired separation velocity for quadrotor i. $\mathcal{N}_{i,\text{sepa}}$ denotes the set of neighboring objects within distance $r_{ij,\text{free}}$ around quadrotor i, including obstacle points, spherical obstacles and other quadrotors, where $r_{ij,\text{free}}$ is evaluated with the direction-dependent expression of Eq. (12). In addition, the sensing radius r_{sen} restricts the obstacle points that are evaluated, so that objects beyond the perception distance have no effect on the quadrotor.

3.4 Follow-motion and verification

In addition to the cohesion, alignment and separation parts that constitute the adaptive-motion layer, every quadrotor is also guided by the virtual node (quadrotor 0) on velocity v_{floc}, to accomplish the follow-motion part. Therefore, the desired velocity for quadrotor i is combined as the four-component equation above. Then, to meet the limits from $a_{\max}$ and $v_{\max}$, the published desired velocity must be verified to prevent the response of autopilot from becoming saturated. Using the saturation operator defined in Eq. (2), the velocity increment is limited first and the speed afterwards:

$$\delta v_i = v_{i,\text{des}} - v_i, \tag{21}$$

$$v_{i,\text{tmp}} = v_i + \text{sat}_{a_{\max}\delta t}(\delta v_i), \tag{22}$$

$$v_{i,\text{pub}} = \text{sat}_{v_{\max}}(v_{i,\text{tmp}}). \tag{23}$$

Here δv_i, given by Eq. (21), is the change of the desired velocity during the step time δt, $v_{i,\text{tmp}}$ is a temporary vector whose increment is limited by $a_{\max}\delta t$, and $v_{i,\text{pub}}$ denotes the published desired velocity, whose norm is limited by $v_{\max}$. Since sat maps the zero vector to the zero vector, no unit vector has to be formed at this step, and the two limits are applied in the order given above.

The resulting procedure for one control step of quadrotor i is summarized below.

Algorithm 1: flocking step of quadrotor i at time t

(1) Read the own position r_i, velocity v_i and attitude, together with the leader and neighbor states.
(2) Compute the cohesion term $v_{i,\text{cohe}}$ of Eq. (4), which is inactive when $r_{i0} \leqslant r_{\text{in}}$ or $r_{i0} \geqslant r_{\text{out}}$.
(3) Sum the pairwise alignment terms $v_{ij,\text{aligned}}$ of Eq. (6) over the set $\mathcal{N}_{i,\text{aligned}}$.
(4) Set the polar axis and the eccentricity: if $\|v_i\| < \varepsilon_v$, take the body forward axis and $e_i = e_0$; otherwise take $\widehat{v_i} = v_i/\|v_i\|$ and $e_i = e_{\max}\|v_i\|/v_{\max}$.
(5) Sum the pairwise separation terms $v_{ij,\text{sepa}}$ of Eq. (18) over the objects inside their respective free-distance boundaries, applying the rules given in Subsection 3.3 for the degenerate cases.
(6) Compute the follow-motion term v_{floc} from the leader velocity and the altitude regulation.
(7) Accumulate the four contributions as $v_{i,\text{des}} = v_{i,\text{cohe}} + v_{i,\text{aligned}} + v_{i,\text{sepa}} + v_{\text{floc}}$.
(8) Limit the increment and the speed as $v_{i,\text{pub}} = \text{sat}_{v_{\max}}(v_i + \text{sat}_{a_{\max}\delta t}(v_{i,\text{des}} - v_i))$.
(9) Publish $v_{i,\text{pub}}$ as the velocity setpoint of the quadrotor.

4 Experiment Settings and Results

This section presents the settings and results of the proposed motion planning method in both simulation and outdoor experiment. First, a system architecture for the quadrotor swarm is designed to implement the proposed motion planning method. Subsequently, an obstacle environment is constructed in the Gazebo simulator, where a swarm of 4 Iris-quadrotor models is used to validate the method's effectiveness. Finally, the method is reimplemented on an outdoor swarm of 3 quadrotors,

to achieve cooperative flight under real-world conditions.

4.1 Architecture of the quadrotor swarm

The architecture adopts the flocking behavior framework to enable the cooperative flight of a quadrotor swarm, shown in Fig. 7A. Here, every quadrotor of the swarm is guided by the same proposed motion planning method. It receives the states of the configured neighbors, and broadcasts its own state to the local communication network. It only detects obstacles within the effective perception distance r_{sen}. In addition, none of the quadrotors know the global path. Therefore, a virtual leader quadrotor is introduced, which ideally follows the predefined global path. It broadcasts its position as the swarm center r_0, and the current velocity as a follow-motion reference v_{floc}.

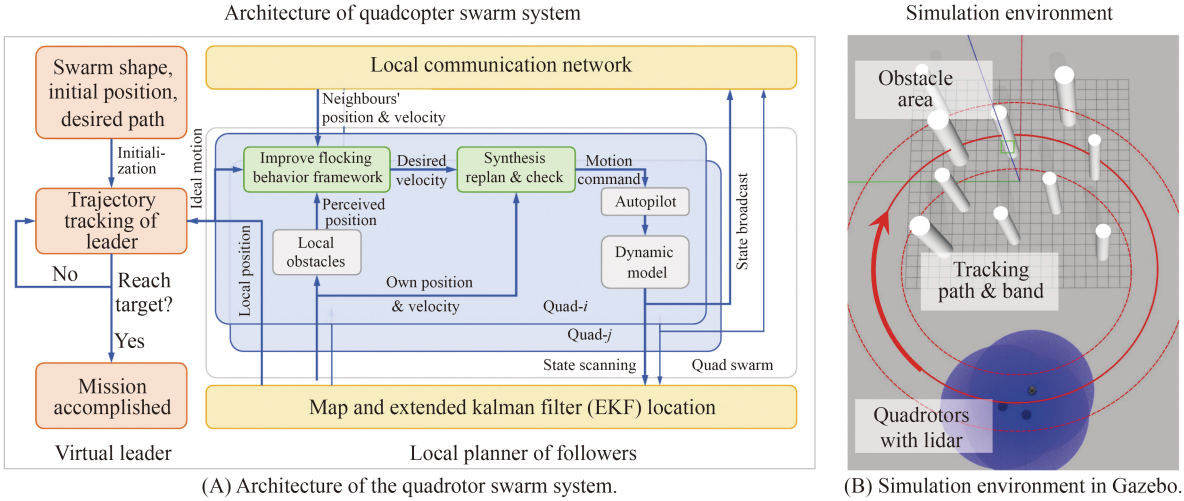
To make the implementation reproducible, the execution specification of the swarm system is summarized as follows.

- Update rate: the planner runs on the onboard computer with Ubuntu 20.04 and executes its control loop at 30 Hz, publishing one command per iteration.
- Inputs: the own state, the own odometry, the odometry of the configured neighbors and the GPS time reference.
- Communication topology: each quadrotor runs its own Robot Operating System (ROS) master, the states of the configured neighbors are exchanged over the local network, and the leader broadcasts its own state to the followers.
- Commands: velocity setpoints during the flocking phase, while position setpoints are used only in the hover and formation phases.
- Mode gating: commands are produced only in the offboard mode with the flocking submode selected.
- Time synchronization: every received state is corrected by a GPS-time clock offset, $T_{\text{own}} = T_{\text{other}} - \text{bias}_{\text{other}} + \text{bias}_{\text{own}}$.
- Frame synchronization: a common ENU origin is published by the leader and adopted by the other quadrotors.
- Localization: RTK GNSS with the common origin.
- The neighbor set is given by the configuration: in the outdoor flight, the two follower quadrotors exchange their states with each other and both receive the state of the leader.
- Not measured: the state-publication rate of the flight stack, the packet-loss rate, the delay distribution, the update jitter and the number of link reconfigurations.

Accordingly, the items above are the confirmed facts of the implementation, whereas no communication guarantee and no latency value are claimed in this paper.

4.2 Simulation

The simulation platform is constructed using the ROS on Ubuntu 20.04, together with PX4 Autopilot Firmware v1.13. The simulation environment and Iris quadrotor model is established in the Gazebo simulator. As shown in Fig. 7B, 10 cylinders on varying radii are randomly distributed in the 16 m × 16 m square area, which create an obstacle environment. The global path is predefined as a circle that repeatedly pass through the obstacle area. An Iris quadrotor flies above the obstacles and tracks the global path, as the virtual leader. There Iris quadrotors equipped with lidars run the proposed method with the same parameters, fly in the environment and follow the



(A) Architecture of the quadrotor swarm system. (B) Simulation environment in Gazebo.

Fig. 7 Architecture and simulation settings of the multi quadrotor system.

virtual leader. The lidars are also provided by plug-ins in Gazebo, to detect nearby obstacles.

With the parameters shown in Table 1, the setups provide a benchmark for testing the method with the sensing and dynamic limits of the platform.

Table 1 Parameters of the method in simulation and outdoor experiment.

Parameter	Simulation	Outdoor experiment	Unit	Category
v_{max}, max. velocity	5.0	2.0	m/s	changed
a_{max}, max. acceleration	15.0	10.0	m/s^2	changed
r_{safe}, collision distance	0.3	0.45	m	changed
r_{sen}, perception distance	5.0	5.0	m	shared
r_{pack}, packing radius	1.05	0.945	m	changed
v_{res}, min. velocity difference	0.5	0.2	m/s	changed
r_{res}, residual distance	0.5	0.5	m	shared
b, semi-minor axis	1.0	1.0	m	shared
p_{cohe}, cohesion gain	1.5	1.0	–	changed
$p_{aligned}$, alignment gain	0.6	0.5	–	changed
$p_{sepa,quad}$, quadrotor	0.3	0.3	–	shared
$p_{sepa,obs}$, obstacle	0.05	0.5	–	changed

In the table, all values other than the obstacle-separation gain are set to match the size of the real quadrotors and to keep the outdoor flight safe. The separation gain to an obstacle point $p_{sepa,obs}$ is the only parameter that is changed because of the obstacle representation, since the simulation evaluates the term for every sensed obstacle point of the point cloud, whereas the outdoor planner evaluates one term for each spherical obstacle. Consequently, the two experiments use different obstacle models, and their results are evidence of feasibility under two deployment conditions rather than a controlled comparison of tuned performance.

The performance of the proposed method in simulation platform is post-processed in MATLAB from the flight logs, as shown in Figs. 8–10. In Fig. 8, the red curve with Quad 0 tag is trajectory of the virtual leader. It is a circle on $x(t) = 11\left(1-\cos\left(2\pi\cdot\frac{t}{60}+\phi\right)\right)-20$ m, $y(t) = 11\sin\left(2\pi\cdot\frac{t}{60}+\phi\right)$m, $z(t) = 5$ m, where t is the flight time, ϕ is used to adjust the initial positions. Then, the other curves with Quad 1-3 tags are trajectories of the 3 quadrotors, which are guided by the proposed planning method. Figs. 8A and 8B present the ideal trajectory and 3 actual trajectories while tracking, respectively on 3D and top view. Fig. 8B also shows that the quadrotors choose different branches in path furcation to pass through obstacles in different conditions.

Fig. 9 records the position of quadrotors on 3 axes during the flight in simulation. It can be seen that the quadrotors get together in empty area, and deviate from the circle trajectory as the separation term becomes active in the obstacle area. Besides, according to Fig. 10, the magenta line with Collision Distance tag is the minimum distance for two quadrotors to collide with each other.

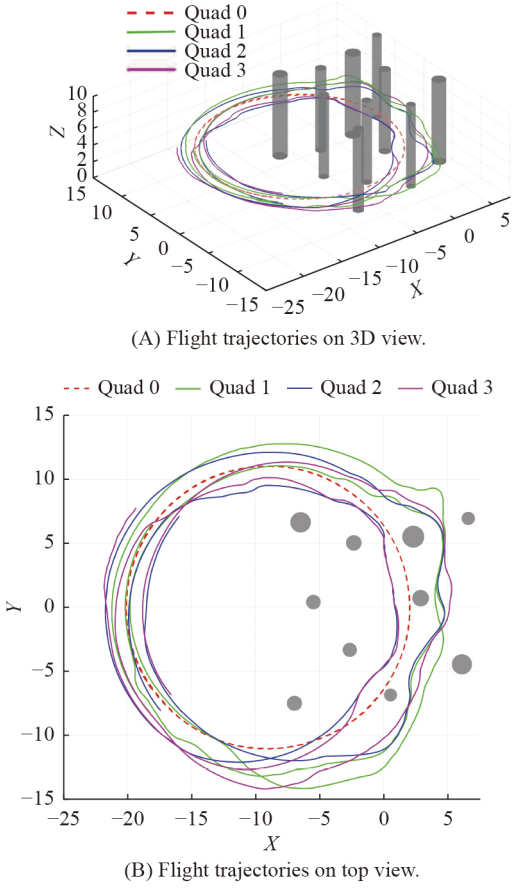


Fig. 8 Flight trajectories in simulation.

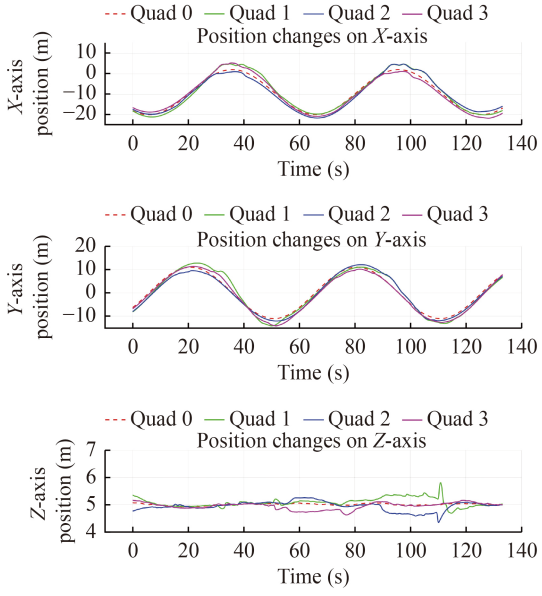


Fig. 9 Positions of quadrotors on 3 axes during the flight in simulation.

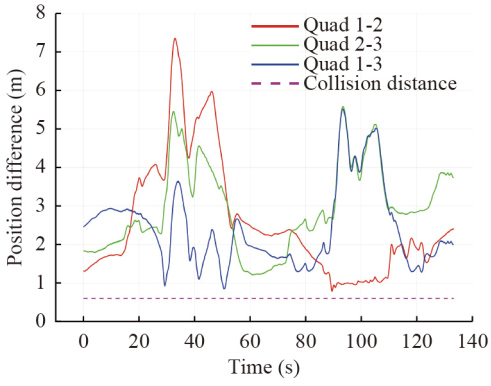


Fig. 10 Position-differences between quadrotors during the flight in simulation.

The other curves are distances between each pair of quadrotors in the swarm. Therefore, the logged pairwise center distances remain above the collision distance throughout the analysed interval, which is the separation evidence reported here.

4.3 Outdoor experiment

The outdoor experiment platform is constructed with 3 quadrotors. Every quadrotor, as shown in Fig. 11, is equipped with a MicoAir743v2 Autopilot for flight control, an RTK (Real-Time Kinematic) module for high-precision localization, and an NVIDIA Jetson Orin NX onboard computer for swarm communication and implementing the proposed method. In the platform, one quadrotor flies above the obstacle area as the leader quadrotor. Its state information is mapped to the corresponding state inside the obstacle area, serving as the virtual leader. The other 2 quadrotors operate as followers in the obstacle area. The followers receive the mapped virtual leader state as the follow-motion reference and receive the states of the other followers. The initial conditions are as follows: the two followers start from the formation offsets $(2.0, 0, -2.0)$ m and $(0, -1.732, -2.0)$ m relative to the leader, their altitude is regulated to 6 m, and the leader flies a clockwise circle of 5 m radius at 0.5 m/s at an altitude of 8 m above the obstacle field. The leader state used by the planner is the vertical projection of the physical leader onto the obstacle plane, that is, the followers hold their altitude at the cruise height and take the horizontal position and velocity of the leader as the virtual-leader reference. The interval analysed below starts 150 s after the earliest sample of the recorded logs and lasts 130 s.



Fig. 11 A quadrotor platform in the outdoor experiment.

The proposed method is implemented as a ROS package in Python, which runs on the Ubuntu 20.04 system installed on the onboard computer. The outdoor experiment environment captured by the camera is shown in Fig. 12. To simplify the system, the quadrotor does not carry an actual environment sensing hardware such as LiDAR, and no onboard camera is used for obstacle detection. Instead, an $8\,\text{m} \times 8\,\text{m}$ obstacle area is predefined in the package, with 10 virtual spherical obstacles randomly distributed in the same horizontal plane as the virtual leader, whose centers lie on the cruise plane of the followers. Each obstacle is given by a center and a radius between 0.25 m and 0.43 m, and these parameters are prior inputs of the planner rather than measurements: they are used directly in the separation part, in place of obstacle point clouds. Therefore, the experiment validates the distributed execution and the trajectory tracking of the planner under known obstacle geometry, and not an online obstacle perception or mapping function. As the obstacle representation differs from the simulation, the gain of separation to an obstacle takes a different value in the two experiments, as reported in Table 1, whereas the remaining parameters are set for the real airframe and for a safe flight. Every follower quadrotor runs the proposed method with the same parameter table.

The performance of the proposed method in outdoor

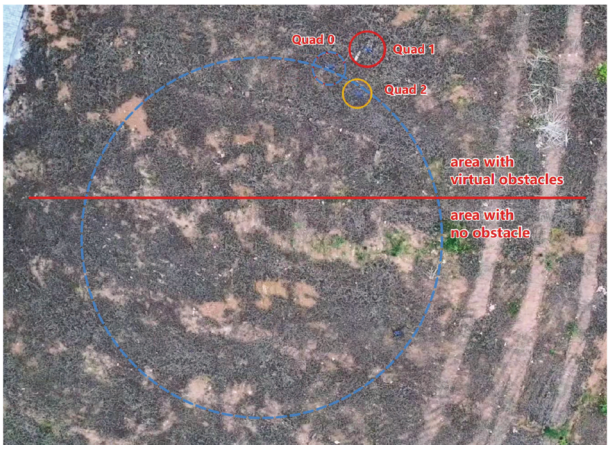


Fig. 12 Outdoor experimental environment.

experiment platform is post-processed in MATLAB from the flight logs, as shown in Figs. 13−15. In Fig. 13, trajectories of the virtual leader and the 2 followers present the overall spatial motion of the swarm in the outdoor experiment, together with the predefined obstacle environment. Here, the blue curve with Quad 0 tag is trajectory of the virtual leader. It is a circle on $x(t) = 5\left(1 - \cos\left(2\pi \cdot \frac{t}{60} + \phi\right)\right)$ m, $y(t) = 5\sin\left(2\pi \cdot \frac{t}{60} + \phi\right)$ m, $z(t) = 6$ m, where t is the flight time, ϕ is used to adjust the initial positions.

The other curves with Quad 1 and 2 tags are trajectories of the 2 quadrotors, which are guided by the proposed planning method. Under the guidance of the virtual leader, the 2 followers complete cooperative flight through the obstacle area while maintaining the intended swarm motion. These trajectories indicate that, in the outdoor platform and under the tested conditions, the two followers track the virtual leader while keeping their clearances above the thresholds reported below.

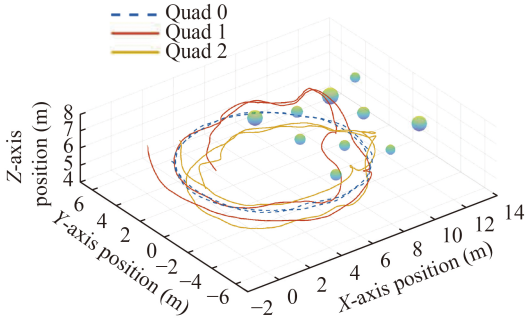


Fig. 13 Flight trajectories on 3D view in outdoor experiment. The curves are expressed in the ENU frame centered on the navigation start pose of the virtual leader, and the leader trajectory is drawn 2 m below its logged altitude so that it lies in the obstacle plane, which is a plotting choice.

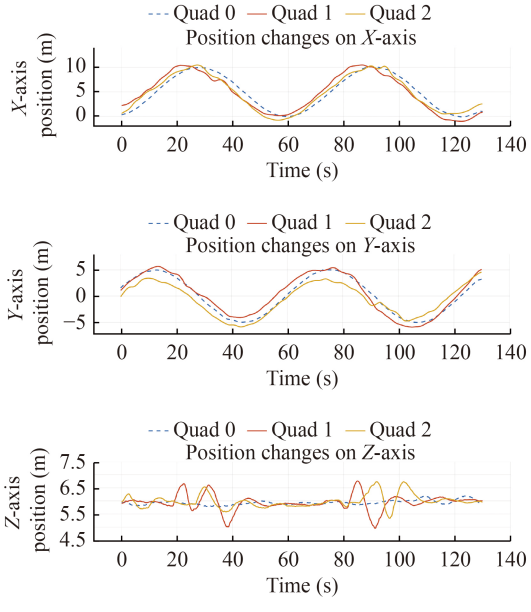


Fig. 14 Positions of quadrotors on 3 axes during the flight in outdoor experiment. The time axis covers the analysed window of 130 s, rebased to zero.

Fig. 14 further shows the position variations of the virtual leader and the 2 followers along the 3 axes during the flight. It can be observed that the followers track the virtual leader continuously. Their positions deviate adaptively in the obstacle region and return to the neighborhood of the virtual leader in the unobstructed region.

Fig. 15 shows the clearances between the 2 followers, together with the clearances from each follower to the obstacles. The inter-vehicle curves are center-to-center distances, for which the collision distance is $2r_{\text{safe}} = 0.9$ m, whereas the obstacle curves are surface clearances $\|r_i - c_o\| - r_o$, for which the collision distance is $r_{\text{safe}} = 0.45$ m; the minimum distance and the collision distance of every subfigure are emphasized in bold lines and labeled. The logged clearances remain above the corresponding thresholds over the analysed window, which is the separation evidence obtained in this experiment. Since the two logs are not sampled at the same instants, the trajectory of one follower is interpolated linearly onto the time base of the other for this comparison.

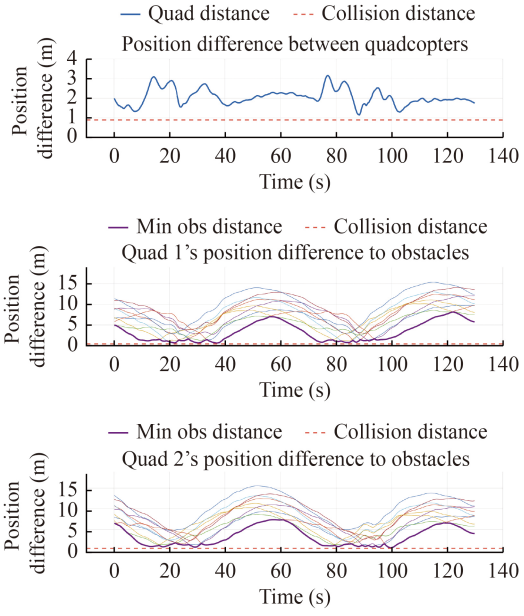


Fig. 15 Position differences between quadrotors and obstacles during the flight in outdoor experiment. Center-to-center distances are used between the quadrotors and surface clearances to the obstacles, and the trajectory of one follower is interpolated linearly onto the common time base.

4.4 Log processing and metric definitions

The outdoor results are obtained by post-processing the flight logs, and the protocol is summarized below so that the reported figures can be reproduced.

(1) Logged signals: the flight-controller logs of the three quadrotors are used, namely the logged local position of each vehicle and its GNSS time reference; the velocity commands and the planned references are not used for the figures.

(2) Time alignment: an absolute UTC axis is reconstructed for every record from its GNSS timestamps, and the three records are aligned to the earliest UTC instant of the three logs.

(3) Coordinate conversion: the logged local positions are converted into a common ENU frame and expressed relative to the navigation start pose of the virtual leader.

(4) Cropping: the analysed interval is 150–280 s after the common origin, and it is rebased to 0–130 s in the figures.

(5) Interpolation: for the pairwise curves, the trajectory of one follower is interpolated linearly onto the time base of the other one, since the two logs are not sampled at the same instants; this operation is used only for the distance computation.

(6) Filtering and discarded samples: no smoothing, no outlier removal and no gap filling is applied, and the only samples absent from the figures are those outside the cropping interval.

(7) Plotting choices rather than controller operations: the rebasing of the time axis, the ENU conversion together with the choice of the origin, the 2 m downward shift used to draw the leader trajectory in the obstacle plane, the interpolation onto a common time base, the collision-distance reference lines and the axis limits of the figures.

The distances shown in Fig. 15 are the clearances defined in Subsection 2.1, namely the center-to-center distance between the quadrotors and the surface clearance to the obstacles.

4.5 Scope and limitations

The assumptions that support the braking curve are listed in Subsection 3.2, where they are marked as satisfied or unproved. Two further limitations of the reported evaluation should be noted. First, the outdoor experiment involves three quadrotors, namely one physical leader and two followers, and the obstacle geometry is a prior input, so that the results support the distributed execution and the trajectory tracking of the planner under known obstacle conditions; they do not establish a formal safety guarantee. Second, the delay, the packet loss and the jitter of the communication link were not measured, so the corresponding assumptions remain unproved, and the safety margin is supported by the logged clearances rather than by an invariant-set argument over the discrete-time system.

5 Conclusions

In this paper, a distributed motion planning method is proposed for the cooperative flight of quadrotor swarms in obstacle environments, which is based on the flocking behavior framework. A separation algorithm via the safe valley attention field is presented as part of the method, to achieve proactive and smooth separation in the swarm flight. The reported evidence can be summarized at three levels. The simulation supports the planning and separation behavior in the Gazebo scenario. The outdoor experiment supports the distributed execution and the trajectory tracking of three quadrotors with ten predefined virtual spherical obstacles, for which the logged clearances remain above the collision thresholds. The braking curve provides a local, velocity-dependent interpretation of the separation geometry, rather than a formal guarantee of collision avoidance. The assumptions and the limitations of this interpretation are given in Subsections 3.2 and 4.5.

Data availability The datasets used and/or analyzed during the current study are available from the corresponding author upon reasonable request.

Acknowledgments This research received no external funding. The authors used DeepSeek-Flash (online version) as a language-editing tool to improve grammar, clarity, and readability. The AI tool was not used for data analysis, data generation, image processing, or scientific content creation. All outputs were carefully reviewed and verified by the authors, who take full responsibility for the final content.

Competing interests The authors declare no conflicts of interest.

References

1. Hossain N U I, Sakib N, Govindan K. Assessing the performance of unmanned aerial vehicle for logistics and transportation leveraging the Bayesian network approach. Expert Systems with Applications, 2022, 209: 118301
2. Lee H W, Lee C S. Research on logistics of intelligent unmanned aerial vehicle integration system. Journal of Industrial Information Integration, 2023, 36: 100534
3. Liang J C, Zhong H, Wang Y N, et al. Reference optimization-based compliant control for aerial pipeline inspection using a hexacopter with a robotic contact device. IEEE/ASME Transactions on Mechatronics, 2024, 29(6): 4063−4074
4. Zhong H, Liang J C, Chen Y J, et al. Prototype, modeling, and control of aerial robots with physical interaction: a review. IEEE

Transactions on Automation Science and Engineering, 2025, 22: 3528−3542
5. Mohite J, Sawant S, Agarwal R, et al. Detection of crop water stress in maize using drone based hyperspectral imaging. In: IGARSS 2022−2022 IEEE International Geoscience and Remote Sensing Symposium. Kuala Lumpur, Malaysia, 2022, 5957–5960
6. Istiak M A, Syeed M M M, Hossain M S, et al. Adoption of Unmanned Aerial Vehicle (UAV) imagery in agricultural management: a systematic literature review. Ecological Informatics, 2023, 78: 102305
7. Cao P, Lei L, Cai S S, et al. Computational intelligence algorithms for UAV swarm networking and collaboration: a comprehensive survey and future directions. IEEE Communications Surveys & Tutorials, 2024, 26(4): 2684−2728
8. Javed S, Hassan A, Ahmad R, et al. State-of-the-art and future research challenges in UAV swarms. IEEE Internet of Things Journal, 2024, 11(11): 19023−19045
9. Yu D X, Li J C, Wang Z, et al. An overview of swarm coordinated control. IEEE Transactions on Artificial Intelligence, 2024, 5(5): 1918−1938
10. Amirkhani A, Barshooi A H. Consensus in multi-agent systems: a review. Artificial Intelligence Review, 2022, 55(5): 3897−3935
11. Li R C, Yang Q K, Zhao W P, et al. Collision-free trajectory generation for multiple UAVs with sensing constraints. In: 2021 40th Chinese Control Conference (CCC). Shanghai, China, 2021, 5592–5597
12. Zhou X, Zhu J C, Zhou H Y, et al. EGO-Swarm: a fully autonomous and decentralized quadrotor swarm system in cluttered environments. In: 2021 IEEE International Conference on Robotics and Automation (ICRA). Xi'an, China, 2021, 4101–4107
13. Arul S H, Manocha D. SwarmCCO: probabilistic reactive collision avoidance for quadrotor swarms under uncertainty. IEEE Robotics and Automation Letters, 2021, 6(2): 2437−2444
14. Ourari R, Cui K, Elshamanhory A, et al. Nearest-neighbor-based collision avoidance for quadrotors via reinforcement learning. In: 2022 International Conference on Robotics and Automation (ICRA). Philadelphia, USA, 2022, 293–300.
15. Quan L, Yin L J, Zhang T R, et al. Robust and efficient trajectory planning for formation flight in dense environments. IEEE Transactions on Robotics, 2023, 39(6): 4785−4804
16. Plou C, Pueyo P, Martinez-Cantin R, et al. Gen-swarms: adapting deep generative models to swarms of drones. In: 2024 International Conference on Computer Vision. Milan, Italy, 2025, 85–101
17. Peng P, Dong W, Chen G, et al. Obstacle avoidance of resilient UAV swarm formation with active sensing system in the dense environment. In: 2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). Kyoto, Japan, 2022, 10529–10535
18. Guo J J, Qi J T, Wang M M, et al. Distributed cooperative obstacle avoidance and formation reconfiguration for multiple quadrotors: theory and experiment. Aerospace Science and Technology, 2023, 136: 108218
19. Mao P D, Fu R, Quan Q. Optimal virtual tube planning and control for swarm robotics. The International Journal of Robotics Research, 2024, 43(5): 602−627
20. Luo L B, Wang X Y, Ma J F, et al. GrpAvoid: multigroup collision-avoidance control and optimization for UAV swarm. IEEE Transactions on Cybernetics, 2023, 53(3): 1776−1789
21. Sun G B, Zhou R, Ma Z, et al. Mean-shift exploration in shape assembly of robot swarms. Nature Communications, 2023, 14(1): 3476
22. Reynolds C W. Flocks, herds and schools: a distributed behavioral model. ACM SIGGRAPH Computer Graphics, 1987, 21(4): 25−34



Hongtao Liang received the B.S. degree in Automation Science from Hunan University, Changsha, China, in 2023. He is currently with the School of Artificial Intelligence and Robotics, Hunan University, Changsha, China, and with the National Engineering Research Center for Robot Visual Perception and Control Technology, Hunan University, Changsha, China. His research interests include cooperative behavior of unmanned aerial vehicle swarms and intelligent control in multi-robot systems.



Yaonan Wang received the Ph.D. degree in Electrical Engineering from Hunan University, Changsha, China, in 1994. Since 1995, he has been a Professor with the College of Electrical and Information Engineering, Hunan University. From 1994 to 1995, he was a Post-Doctoral Research Fellow with the National University of Defense Technology, Changsha. From 1998 to 2000, he was a Senior Humboldt Fellow with the University of Bremen, Bremen, Germany. From 2001 to 2004, he was a Visiting Professor with the University of Bremen. His research interests include robotics and image processing. He is a member of Chinese Academy of Engineering.



Hang Zhong received the B.S. degree in Electrical Engineering and Automation and the Ph.D. degree in Control Science and Engineering from Hunan University, Changsha, China, in 2013 and 2020, respectively, through a successive master-doctoral program under the supervision of Prof. Yaonan Wang, Academician of the Chinese Academy of Engineering. From 2020 to 2022, he was a Postdoctoral Fellow with Hunan University, where he is currently an Associate Professor with the School of Artificial Intelligence and Robotics. His research interests include aerial robotics, multirobot systems, visual servoing, visual navigation, and nonlinear control. Dr. Zhong was the recipient of the Outstanding Doctoral Dissertation Award of Hunan Province in 2022.



VanCuong Pham received the B.S. and M.S. degrees from the Department of Automatic Control, Military Technical Academy, Vietnam, in 2007 and 2010, respectively, and Ph.D. degree in Control Science and Engineering from Hunan University, Changsha, China, in 2015. He joined the Faculty of Electrical Engineering Technology as a Lecturer in Hanoi University of Industry, Hanoi, Viet Nam since 2003. His research interests include intelligent control theory, adaptive control, robust control, applications and robot manipulators.



Hao Chang received the B.S. degree in Automation Science from Hunan University, Changsha, China, in 2024. He is currently with the School of Artificial Intelligence and Robotics, Hunan University, Changsha, China, and with the National Engineering Research Center for Robot Visual Perception and Control Technology, Hunan University, Changsha, China. His research interests include intelligent physical interaction control of aerial robots.



Xin Chen received the B.S. degree in Artificial Intelligence from China University of Mining and Technology, Xuzhou, China, in 2024. She is currently with the School of Artificial Intelligence and Robotics, Hunan University, Changsha, China, and with the National Engineering Research Center for Robot Visual Perception and Control Technology, Hunan University, Changsha, China. Her research interests include motion planning for unmanned aerial vehicles.



Hui Zhang received his B.S., M.S., and Ph.D. degrees in Pattern Recognition and Intelligent Systems from Hunan University, Changsha, China, in 2004, 2007, and 2012, respectively. From 2017 to 2018, he held a position as a Visiting Scholar at the University of Windsor, Windsor, Canada. He currently serves as Dean of the School of Artificial Intelligence and Robotics, Dean of the College of Future Technology at Hunan University, and Deputy Director of the National Engineering Research Center for Robot Visual Perception and Control Technology. His primary research interests lie in robotic visual inspection, deep learning-based image recognition, and intelligent manufacturing robot technology.



Hean Hua received the B.S. degree in Measurement and Control Technology and Instruments from Harbin Engineering University, Harbin, China, in 2017, and the Ph.D. degree in control science and engineering from Nankai University, Tianjin, China, in 2022. From 2022 to 2025, he was a Postdoc Fellow with the Department of Electrical and Information Engineering, Hunan University, Changsha, China, where he is currently an Associate Professor with the School of Artificial Intelligence and Robotics. His current research interests include motion planning, intelligent control, and machine learning for autonomous UAVs.