IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received January 10, 2019, accepted January 23, 2019, date of publication January 31, 2019, date of current version February 14, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2895643

Flocking Control of Fixed-Wing UAVs With Cooperative Obstacle Avoidance Capability

WEIWEI ZHAO[1,2], HAIRONG CHU[1], MINGYUE ZHANG[1], TINGTING SUN[1], AND LIHONG GUO[1]

[1]Changchun Institute of Optics, Fine Mechanics and Physics, Chinese Academy of Sciences, Changchun 130033, China
[2]College of Materials Science and Opto-Electronic Technology, University of Chinese Academy of Sciences, Beijing 10049, China

Corresponding author: Tingting Sun (zhaowei5512@foxmail.com)

This work was supported by the Research on Estimation Algorithm and Test Technology for Guidance Information of Strapdown Imaging Seeker, under Grant 20180520018JH.

ABSTRACT In recent years, with the development of the unmanned aerial vehicle (UAV) and battlefield environments, the UAV swarm has attracted significant research attention. To solve problems regarding poor state consensus among swarm individuals due to a small number of individuals easily falling into local minima upon encountering an obstacle, this paper proposes a flocking obstacle avoidance algorithm with local interaction of obstacle information. To make the UAV swarm follow the desired trajectory with better state consensus, we improved the flocking control algorithm of agents according to the characteristics and requirements of the UAV swarm. The obstacle avoidance algorithm for the UAV swarm is based on Olfati-Saber's multi-agent obstacle avoidance algorithm. The proposed method has individuals in the swarm communicate obstacle information with their neighbors, and we present a simple analysis of this method. The method improves the cooperative obstacle avoidance capability of the flocking control algorithm. The simulation results showed that the proposed flocking control algorithm provides a better tracking effect and consensus for the UAV swarm when avoiding obstacles.

INDEX TERMS Fixed-wing UAV swarm, multi-agent system, flocking control, cooperative obstacle avoidance, consensus, local information communication.

I. INTRODUCTION

Flocking is a common phenomenon in nature that has gained significant attention in various research fields [1]–[7]. In 1986, Reynolds [8] introduced three heuristic rules of flocking control: cohesion, separation, and alignment. Ever since, this classical model has often been applied to flocking control of multi-agents.

In recent years, the unmanned aerial vehicle (UAV), as the embodiment of the agent, has attracted much attention due to increasingly high autonomy levels and the application value of the UAV in military and civilian fields [9]–[15]. According to Lanchester's laws, the number of combat units is a decisive factor for victory or defeat in warfare and is more important than the unit's capability. Thus, the application of UAV swarms in warfare situations may affect battlefield conditions. The cooperative UAV swarm can accomplish highly difficult and dangerous tasks, which has roused the interest of many researchers. Towards these advantages, the issue of flocking obstacle avoidance is a major focus in flocking control.

Qiu and Duan [16] presented a UAV distributed flocking control algorithm for obstacle environments based on a pigeon flocking model, which used pigeon behavior where flocks switch between hierarchical and egalitarian interaction modes at different flight phases. Olfati-Saber [5] proposed an additional feedback term to track a virtual leader, which is necessary to avoid fragmentation. The author also presented a flocking algorithm with obstacle avoidance capability by creating a virtual agent on the boundary of each nearby obstacle. Su *et al.* [6] considered a case where only a small fraction of agents has information about the virtual leader with a constant velocity and a varying velocity. Kownacki and Ołdziej [17] presented a novel approach to swarm control of small fixed-wing UAVs through cohesion and repulsion behaviors combined with leadership. Zhang and Duan [18] presented a 3D flocking control algorithm for tracking a desired trajectory with obstacle avoidance capability using the improved artificial potential field method. Li *et al.* [19] studied the multi-agent

The associate editor coordinating the review of this manuscript and approving it for publication was Heng Zhang.

system coordination obstacle avoidance algorithm using a variable structure method, wherein only some agents have dynamic information on obstacles and each agent has a local interaction. Sakai *et al.* [20] proposed a flocking algorithm that does not distinguish between a robot and an obstacle, which was constructed by modifying Olfati-Saber's control law [5]. Iovino *et al.* [21] presented a real quad-rotor UAV experiment using a distributed flocking algorithm with obstacle avoidance capability, where the obstacle avoidance algorithm was derived from [5]. Wang *et al.* [22] used the artificial potential function combined with the stream function to asymptotically reach the ideal stable flocking motion, which not only keeps the dynamic multi-agent system constantly network-connected, but also enables all agents to avoid obstacles without being caught in local minima. Luo and Duan [23] presented a distributed control framework based on homing pigeon hierarchy strategies to solve the problem of flocking. The algorithms were generally implemented to achieve stable performance by controlling the local position and velocity of each UAV. Vries and Subbarao [24] used a potential function to generate steering commands to control a swarm of quad-rotors. A flocking obstacle avoidance algorithm has been proposed using a combination of velocity consensus and local artificial potential field [25].

The flocking obstacle avoidance algorithms used in the above papers did not consider the interaction of obstacle information within the swarm and thus could not use that information to avoid obstacles to the maximum extent. This results in a poor consensus within the swarm when avoiding obstacles. Therefore, to solve this problem, we propose a shared obstacle information algorithm. The UAV flocking control algorithm with obstacle avoidance capability proposed in this paper is based on Saber's agent flocking control algorithm and considers some of the dynamic characteristics of the UAV and altitude consistency for UAVs. Some of the disadvantages of the agent flocking obstacle avoidance algorithm described in Saber's paper have been discussed in [21]. The method tends to result in oscillations and individuals falling into local minima when UAVs move near obstacles. In this paper, we propose an algorithm for the interaction and sharing of obstacle information between individuals in the UAV swarm so that the swarm can pass through obstacles steadily. A simple analysis of the algorithm is also presented.

The rest of this paper is organized as follows. Section II presents a flocking control algorithm of agents with obstacle avoidance capability and describes a simplified UAV model with a constraint condition and the UAV flocking control algorithm based on flocking control of agents. The proposed obstacle avoidance algorithm with local communication of obstacle information and algorithm analyses are detailed in Section III. A comparison of simulations is conducted in Section IV, and our concluding remarks are drawn in Section V.

II. FLOCKING CONTROL ALGORITHM

The dynamics of agents are modeled as second-order integrators in a three-dimensional Euclidean space as follows:

$$\begin{cases} \dot{q}_i = p_i \\ \dot{p}_i = u_i, \end{cases} \quad i = 1, 2, ...N, \tag{1}$$

where q_i, p_i, and $u_i \in R^n$ denote the position, velocity, and control input vectors of the i-th agent, respectively. Each agent can only communicate with its neighbors within its communications region, and the neighboring set at time t is denoted as follows:

$$N_i^{\alpha}(t) = \{j : ||q_i - q_j|| r, \quad j = 1, 2, ..., N, \ j \neq i, \tag{2}$$

where $\|\cdot\|$ is the Euclidean distance and r is the maximum interaction radius or maximum critical distance. The desired geometric model of the swarm requires that each agent be equally distanced from all of its neighbors and satisfy the following constraints:

$$\left\| q_i - q_j \right\| = d, \quad \forall i, j \in N_i\,(t), \tag{3}$$

where d is a positive constant indicating the minimum allowable distance or minimum critical distance between every pair of neighboring agents, and $d \leq r$.

In a multi-obstacle environment, the input of each agent in the multi-agent control algorithm consists of the following three components [5]:

$$u_i = u_i^{\alpha} + u_i^{\beta} + u_i^{\gamma}, \tag{4}$$

where α, β, and γ denote three kinds of agents used directly from the Olfati-Saber thesis. The α-agent denotes an arbitrary agent in the swarm, and the β-agent is generated from the projection of neighboring α-agents on the surface of the obstacle to represent the physical obstacle avoided. The γ-agent is used to construct the navigational feedback and represents the target to be tracked. u_i^{α} denotes the (α, α) interaction terms, u_i^{β} denotes the (α, β) interaction terms, and u_i^{γ} is a distributed navigational feedback. The definitions of u_i^{α}, u_i^{β}, and u_i^{γ} are as follows:

$$u_i^{\alpha} = -c_q^{\alpha} \sum_{j \in N_i^{\alpha}} \rho_H\,(q_i)\,\phi_{\alpha}\,(q_i) - c_p^{\alpha} \sum_{j \in N_i^{\alpha}} a_{ij}\,(q_i)\,\left(p_i - p_j\right), \tag{5}$$

$$u_i^{\beta} = -c_q^{\beta} \sum_{k \in N_i^{\beta}} b_{i,k}\,(q_i)\,\phi_{\beta}\,(q_i) - c_p^{\beta} \sum_{k \in N_i^{\beta}} b_{i,k}\,(q_i)\,\left(p_i - \hat{p}_{i,k}\right), \tag{6}$$

$$u_i^{\gamma} = -c_q^{\gamma} \sigma_1\left(q_i - q_{\gamma}\right) - c_p^{\gamma}\left(p_i - p_{\gamma}\right) - c_h^{\gamma} Q_h, \tag{7}$$

u_i^{α} causes the agents to gather together and consists of two components. The first component sets the distance between agents to the desired distance. The second component sets the velocity to be consistent with its neighbors. The specific expression for the first component is as follows:

$$\phi_{\alpha}\,(q_i) = \frac{D_{ij}}{\sqrt{1 + \epsilon_{\alpha} \left\| D_{ij} \right\|^2}}, \tag{8}$$

$$D_{ij} = (q_i - q_j) - \frac{q_i - q_j}{\|q_i - q_j\|} * d, \tag{9}$$

$$\rho_H(q_i) = \frac{(\|q_i - q_j\| - d)^2}{H} + 1, \tag{10}$$

where H, ϵ_α, c_q^α, and c_p^α are normal constants. The value of H is typically greater than d. It is important to note that fragmentation is a pitfall of flocking by the Olfati-Saber algorithm. The purpose of using $\rho_H(q_i)$ is to prevent fragmentation. The value of $\rho_H(q_i)$ increases rapidly when the distance between agents grows larger.

The second component of u_i^α is $a_{ij}(q_i) = \rho_h\left(\frac{\|q_i - q_j\|}{r}, h_\alpha\right) \in [0, 1], j \neq i. \rho_h(z)$ is a bump function defined as follows [1]:

$$\rho_h(z) = \begin{cases} 1, & z \in [0, h) \\ \frac{1}{2}\left[1 + \cos\left(\pi \frac{(\text{z-h})}{(1-h)}\right)\right], & z \in [h, 1] \\ 0, & otherwise \end{cases} \tag{11}$$

The purpose of u_i^γ is to allow the agents to track the virtual leader or the desired trajectory. It consists of three components. c_q^γ, c_p^γ, and c_h^γ are normal constants. q_γ and p_γ denote the position and velocity of the virtual leader, respectively. The first and third components are denoted as follows [5], [16]:

$$\sigma_1(q_i - q_\gamma) = \frac{q_i - q_\gamma}{\sqrt{1 + \epsilon_\gamma \|q_i - q_\gamma\|^2}}, \tag{12}$$

$$Q_h = \begin{bmatrix} 0 \\ 0 \\ q_i^h \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ q_\gamma^h \end{bmatrix}. \tag{13}$$

The purpose of Q_h is to minimize the altitude differences between agents and to set the altitude as close as possible to that of the virtual leader. q_i^h and q_γ^h denote the altitudes of agents and virtual leader, respectively.

The purpose of u_i^β is to avoid obstacles. Its principle is as follows: we construct a virtual agent with position and velocity on the surface of the obstacle within the detectable range of the agent in the swarm, and denote the virtual agent as the β-agent. The method of construction is as follows [5]:

i) For an obstacle with a hyper-plane boundary that has a unit normal $\mathbf{a}_k$ and passes through the point y_k, the position and velocity of the β-agent are determined by

$$\hat{q}_{i,k} = Pq_i + (\text{I-P})\, y_k, \quad \hat{p}_{i,k} = Pp_i, \tag{14}$$

where $P = I - a_k a_k^T$ is a projection matrix.

ii) For a spherical obstacle with radius R_k centered at y_k, the position and velocity of the β-agent are given by

$$\hat{q}_{i,k} = \mu q_i + (I - \mu)\, y_k, \quad \hat{p}_{i,k} = \mu P p_{i,} \tag{15}$$

where $\mu = R_k / \|q_i - y_k\|$, $a_k = (q_i - y_k)/\|q_i - y_k\|$, and $P = I - a_k a_k^T$.

According to the above method, a virtual β-agent with corresponding velocity and position is constructed, as shown in Figure 1. The ultimate goal is to keep individuals in the swarm consistent with the virtual β-agent while maintaining a certain distance.

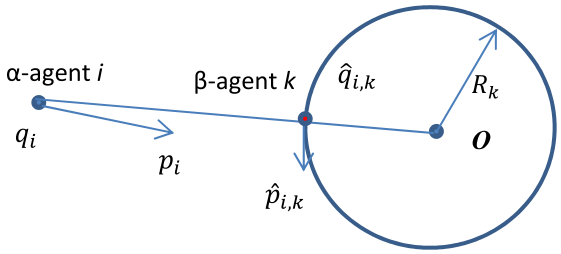


FIGURE 1. Position and velocity of the β-agent.

The role of u_i^β is to make agents pass around obstacles. c_q^β and c_p^β are positive constants, and u_i^β is defined as follows:

$$\phi_\beta(q_i) = \frac{q_i - \hat{q}_{i,k}}{\sqrt{1 + \epsilon_\beta \|q_i - \hat{q}_{i,k}\|^2}} - 1, \tag{16}$$

$$b_{i,k}(q_i) = \rho_h\left(\frac{\|q_i - \hat{q}_{i,k}\|}{r_O}, h_\beta\right), \tag{17}$$

where ϵ_β is a positive constant and r_O is the UAV's maximum detection distance to the obstacle.

A. UAV FLOCKING CONTROL BASED ON FLOCKING ALGORITHM OF AGENTS

Usually, we consider the multi-UAV system to be a multi-agent system. However, we must acknowledge that the motion of the UAV is much more complicated. The simplified kinematic model of a fixed-wing UAV is as follows [27]:

$$\begin{aligned} \dot{x}_i &= V_i cos(\psi_i), \\ \dot{y}_i &= V_i sin(\psi_i), \\ \dot{V}_i &= \frac{1}{\tau_v}(V_i^c - V_i), \\ \ddot{\psi}_i &= -\frac{1}{\tau_{\dot{\psi}}}\dot{\psi}_i + \frac{1}{\tau_\psi}(\psi_i^c - \psi_i), \\ \ddot{h}_i &= -\frac{1}{\tau_{\dot{h}}}\dot{h}_i + \frac{1}{\tau_h}\left(h_i^C - h_i\right), \end{aligned} \tag{18}$$

where $[x_i, y_i, h_i]$, V_i, ψ_i, and $\dot{h}_i$ denote the inertial position, forward velocity, heading angle, and speed of altitude of the i-th UAV, respectively. V_i^c, ψ_i^c, and h_i^C are the command inputs for velocity, heading angle, and altitude to the corresponding autopilots, respectively. τ_ψ, τ_v, and $(\tau_{\dot{h}}, \tau_h)$ are the positive time constants for the heading angle, velocity, and altitude response with respect to the corresponding command inputs, respectively. These four parameters are positive constants that depend on the implementation of the autopilot and the status prediction configuration. Referring to [27], the specific values will be introduced later.

Considering the constraints of a real UAV model, based on the above model, velocity limits, acceleration limits, heading

angle limits, climbing velocity limits, and climbing acceleration limits were introduced. The limiting model is as follows [26]:

$$\begin{aligned} v_{min} &\le V_i \le v_{max}, \\ a_{min} &\le \dot{V}_i = a \le a_{max}, \\ \omega_{min} &\le \dot{\psi}_i = \omega \le \omega_{max}, \\ \alpha_{min} &\le \ddot{\psi}_i \le \alpha_{max}, \\ \lambda_{glide} &\le \dot{h}_i \le \lambda_{climb}, \\ a^h_{min} &\le \ddot{h}_i \le a^h_{max}. \end{aligned} \tag{19}$$

It should be noted that, if the heading angle equation in the simplified kinematics model of the UAV is regarded as a first-order equation, the roll angle cannot be initialized. This is because, according to the coordinated turning condition, the roll angle is calculated from the velocity and heading angular velocity. The pitch angle is calculated with the climbing rate and velocity, as shown below [27].

$$\phi_i = \text{atan}\left(V_i * \frac{\dot{\psi}_i}{g}\right), \tag{20}$$

$$\theta_i = asin(\dot{h}_i / V_i). \tag{21}$$

To apply the agent swarm algorithm to the UAV swarm, we must establish the relationship between the UAV motion model and the agent motion model, referring to previous work by Zhang and Duan [18]. Regarding the UAV swarm system movement in three-dimensional space, the position, velocity, and control vectors are given by $q_i = [x_i, y_i, h_i]^T$, $p_i = [V_i cos\psi_i, V_i sin\psi_i, \dot{h}_i]^T$, and $u_i = [u_{x,i}, u_{y,i}, u_{h,i}]^T$, respectively. Using $\dot{p}_i = u_i$, we obtain:

$$\dot{V}_i = u_{x,i} cos\psi_i + u_{y,i} sin\psi_i, \tag{22}$$

$$\dot{\psi}_i = \frac{u_{y,i} cos\psi_i - u_{x,i} sin\psi_i}{V_i}. \tag{23}$$

Substituting equations (22) and (23) into the UAV simplified model (19) yields:

$$V_i^c = \tau_v \left(u_{x,i} cos\psi_i + u_{y,i} sin\psi_i\right) + V_i, \tag{24}$$

$$\psi_i^c = \frac{\tau_\psi}{\tau_{\dot{\psi}} V_i} \left(u_{y,i} cos\psi_i - u_{x,i} sin\psi_i\right) + \tau_\psi \ddot{\psi}_i + \psi_i, \tag{25}$$

$$h_i^c = h_i + \frac{\tau_h}{\tau_{\dot{h}}} \dot{h}_i + \tau_h u_{h,i}. \tag{26}$$

III. FLOCKING OBSTACLE AVOIDANCE CONTROL ALGORITHM WITH SHARED OBSTACLE INFORMATION CAPABILITY

The obstacle avoidance algorithm was introduced in Section II, but note that, as stated in the paper by Iovino *et al.* [21], multiple UAVs may fall into local optima or decision-making dilemmas when using this obstacle avoidance algorithm, as described below.

As depicted in Figure 2, three UAVs with the same velocity encounter an obstacle. The projective value of velocity of UAV i on the obstacle surface is 0, and the projection of the velocity of UAV j and m on the surface of the obstacle have

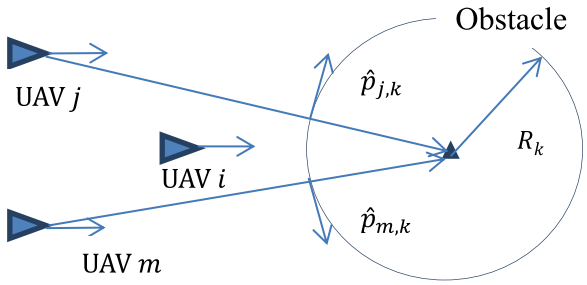


FIGURE 2. Multiple UAVs encountering an obstacle.

the same value with different directions. According to the algorithm in Section II for calculating u_i, especially μ_i^β, it can be found that the velocity value of UAV i is affected by obstacles, which gradually decreases, but the direction of UAV i's velocity is not affected due to $\hat{p}_{i,k} = 0$. The velocity direction and values of UAV j and p are affected to the same degree. This ultimately leads to UAV i potentially colliding with the obstacle and UAV j and p bypassing the obstacle separately. Of course, in practice, the projection of the speed of UAV i on the obstacle will not always be 0 during approach due to various disturbances, which means that UAV i can bypass the obstacle. However, due to the limitations of the algorithm, in the initial stage of the obstacle entering the UAV i detection range, the speed will decrease rapidly, while the direction cannot be changed rapidly, which may eventually cause the distance between UAV i and the obstacle to exceed the minimum allowable distance. To summarize, UAV i cannot jump out of the local optimum quickly, which causes the swarm to have poor consensus. Therefore, to solve this problem, we propose a shared obstacle information algorithm.

When the distance between UAV i and the obstacle is less than the maximum detection distance r_O, UAV i not only needs to share its own position and speed information with the surrounding UAVs, but it also needs to share the obstacle information, $(\hat{q}_{i,k}, \hat{p}_{i,k})$, that it has detected. Of course, when UAV i receives the location and speed information of the surrounding UAVs, it also receives the obstacle information shared by other UAVs. When UAV i receives the obstacle information shared by multi-UAVs, only one pair of obstacle information $(\hat{q}_{\tau,k}, \hat{p}_{\tau,k})$ is selected. The pair is selected based on the maximum speed value, $\max(\|\hat{p}_{\tau,k}\|)$. When the speed values of multiple pairs of obstacle information are equal, the pair for which $\hat{q}_{\tau,k}$ and q_i have the minimum distance, $\min(\|q_i - \hat{q}_{\tau,k}\|)$, is selected. When the velocity and distance values of multiple pairs of obstacle information are equal, a pair is selected randomly. How does UAV i use the selected obstacles information? There is an intuitive awareness that UAV i is farther away from $\hat{q}_{\tau,k}$ in position and closer to $\hat{p}_{\tau,k}$ in speed. Thus, we obtain the following equation:

$$u_i^{\beta,\tau} = -c_q^{\beta,\tau} \left(\frac{q_i - \hat{q}_{\tau,k}}{\sqrt{1 + \epsilon_{\beta,\tau} \left\| q_i - \hat{q}_{\tau,k} \right\|^2}} - 1 \right) - c_p^{\beta,\tau} \left(p_i - \hat{p}_{\tau,k}\right). \tag{27}$$

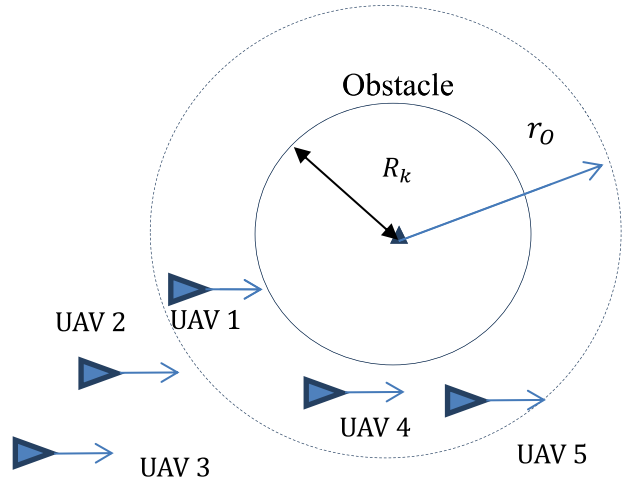


FIGURE 3. Multiple UAVs encountering an obstacle.

When the swarm encounters an obstacle, the input of the UAVs can be one of four cases:

$$u_i = c_\alpha u_i^\alpha + c_\gamma u_i^\gamma, \tag{28}$$
$$u_i = c_\alpha u_i^\alpha + c_\beta u_i^\beta + c_\gamma u_i^\gamma, \tag{29}$$
$$u_i = c_\alpha u_i^\alpha + c_\gamma u_i^\gamma + c_\tau u_i^{\beta,\tau}, \tag{30}$$
$$u_i = c_\alpha u_i^\alpha + c_\beta u_i^\beta + c_\gamma u_i^\gamma + c_\tau u_i^{\beta,\tau}. \tag{31}$$

We use Figure 3 to illustrate equations (28)–(31).

The distance relationship between UAVs in Figure 3 is as follows:

$$d_{1O} < r_O, \quad d_{4O} < r_O, \quad d_{5O} < r_O, \quad d_{2O} > r_O, \quad d_{3O} > r_O$$
$$d_{45} < r, \quad d_{14} > r, \quad d_{12} < r, \quad d_{23} < r, \quad d_{13} > r,$$

where d_{1O} denotes the distance between UAV 1 and the obstacle, d_{45} denotes the distance between UAV 4 and UAV 5, and so on. Thus, the input for UAV 4 and UAV 5 is equation (31), the input for UAV 1 is equation (29), the input for UAV 2 is equation (30), and the input for UAV 3 is equation (28).

A. ALGORITHMS ANALYSIS

In the process of UAV swarm flight, the normal algorithm also requires interactive data, whereas the proposed algorithm only adds obstacle information to the data for obstacles that require interaction. The proposed algorithm also needs to filter the received obstacle information and compute $u_i^{\beta,\tau}$. Therefore, the complexity of the proposed algorithm is not greatly increased compared to the normal algorithm.

We use some visualized expressions to illustrate the superiority of the proposed algorithm. Figure 4-a(1) is the case of the normal algorithm and 4-a(2) is the case of the proposed algorithm. In Figure 4-a(1), the distance between UAVs and obstacles is less than r_O, the distance between UAV i and j is d, and their speed is the same. Thus, $u_i^\alpha = 0$ and $u_j^\alpha = 0$. The velocity direction of UAV i is perpendicular to the obstacle, so $\hat{p}_{i,k} = 0$. From the above assumptions, the input of UAV i and j are obtained as follows:

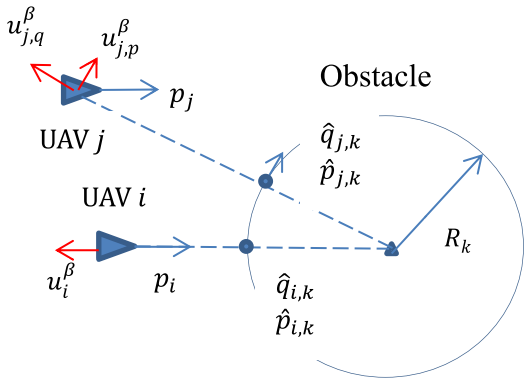
$$u_i = c_\beta u_i^\beta + c_\gamma u_i^\gamma, \tag{32}$$
$$u_j = c_\beta u_j^\beta + c_\gamma u_j^\gamma. \tag{33}$$

In Figure 4-a(2), UAVs and obstacles have the same assumptions as in Figure 4-a(1), but because UAVs share obstacle information with each other, the input of UAV m and UAV n are obtained as follows:

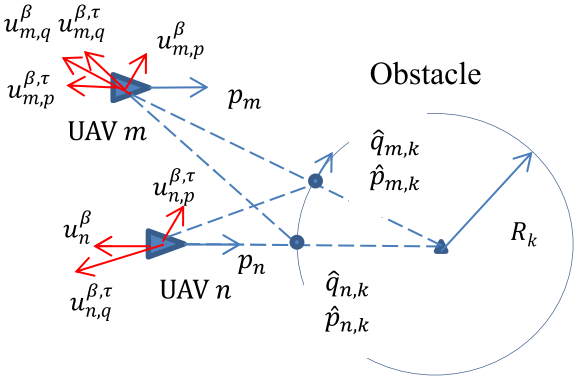
$$u_m = c_\beta u_m^\beta + c_\gamma u_m^\gamma + c_\tau u_m^{\beta,\tau}, \tag{34}$$
$$u_n = c_\beta u_n^\beta + c_\gamma u_n^\gamma + c_\tau u_n^{\beta,\tau}. \tag{35}$$

The obstacle avoidance terms in equations (32)–(35) are composed of two parts, which can be expressed as $u_i^\beta = u_{i,q}^\beta + u_{i,p}^\beta$. $u_{i,q}^\beta$ keeps the UAV away from $\hat{q}_{.,k}$, and $u_{i,p}^\beta$ causes the UAV to approach $\hat{p}_{.,k}$. Therefore, equations (32)–(35) can be expressed as follows:

$$u_i = c_\beta \left(u_{i,q}^\beta + u_{i,p}^\beta\right) + c_\gamma u_i^\gamma, \tag{36}$$
$$u_j = c_\beta \left(u_{j,q}^\beta + u_{j,p}^\beta\right) + c_\gamma u_j^\gamma, \tag{37}$$
$$u_m = c_\beta \left(u_{m,q}^\beta + u_{m,p}^\beta\right) + c_\tau \left(u_{m,q}^{\beta,\tau} + u_{m,p}^{\beta,\tau}\right) + c_\gamma u_m^\gamma, \tag{38}$$



a(1) The normal algorithm.



a(2) The proposed algorithm.

FIGURE 4. Multiple UAVs encountering an obstacle using a(1) the normal algorithm and a(2) the proposed algorithm.

TABLE 1. Parameters of the simplified kinematic model of the UAV.

Parameter	Notation and Value
Minimum and maximum velocity (m/s)	$v_{min} = 15, v_{max} = 40$
Minimum and maximum acceleration (m/s^2)	$a_{min} = -5, a_{max} = 5$
Minimum and maximum velocity of heading angle (rad)	$\omega_{min} = -3.3, \omega_{max} = 3.3$
Minimum and maximum acceleration of heading angle (rad/s^2)	$\alpha_{min} = -1, \alpha_{max} = 1$
Maximum gliding velocity and climbing velocity (m/s)	$\lambda_{glide} = -5, \lambda_{climb} = 5$
Maximum gliding acceleration and climbing acceleration (m/s^2)	$a^h_{min} = -6, a^h_{max} = 6$

$$u_n = c_\beta \left(u^\beta_{n,q} + u^\beta_{n,p} \right) + c_\tau \left(u^{\beta,\tau}_{n,q} + u^{\beta,\tau}_{n,p} \right) + c_\gamma u^\gamma_n. \quad (39)$$

These obstacle avoidance terms are shown as solid red arrows in Figure 4, but u^γ_i is not shown because the effect of u^γ_i for the UAV is the same in both cases. Obstacles have a greater impact on UAVs m and n than on UAVs i and j. UAVs m and n obtain more information about obstacles. The special case where the velocity directions of UAVs i and n are perpendicular to the surface of obstacles provides a clearer understanding. At this special moment, obstacles cause the velocity value of UAV i to decrease, but the direction of UAV i is not affected. However, for UAV n, the obstacle affects both its value and direction. As demonstrated by the above analysis and explanation, the proposed algorithm has a better obstacles avoidance effect than the normal algorithm. The obstacle information detected by each UAV will not only act on itself but also help the adjacent UAVs acquire more comprehensive information about obstacles and avoid obstacles more effectively.

IV. NUMERICAL SIMULATIONS

In this section, we detail the simulations that were performed to verify the validity of the algorithm. All algorithms were implemented using a 3.2 GHz CPU and 8 GB memory personal computer running Windows 10 and Matlab R2016b. A UAV swarm of 7 UAVs ($N = 7$) was considered, assuming the same UAV model for all units, which adheres to the same UAV kinematic simplification model as shown in Table 1. According to the UAV model in [27], we used the following parameter values in our UAV model: $\tau_v = 0.2$, $\tau_{\dot{\psi}} = 0.6250$, $\tau_\psi = 0.0156$, $\tau_{\dot{h}} = 0.7072$, and $\tau_h = 1$ [27]. In addition, according to the actual flight characteristics of the UAV, velocity, acceleration, etc. need to be restricted. The specific data are given in Table 1.

We provided seven initial parameter values for each of the 7 UAVs. The seven parameter values are position, velocity, heading angle, heading angular velocity and speed of altitude, namely $[x_i, y_i, h_i, V_i, \psi_i, \dot{\psi}_i, \dot{h}_i]$, where the initial roll angle is determined according to the velocity and the heading angular velocity, and the velocity and initial velocity determine the initial pitch angle. Therefore, the seven initial parameters determine the state of the UAV in space described by position, velocity, angle, and angular velocity. The seven initial states, once selected, are applied to all the simulations. The values are selected such that the horizontal position is uniformly distributed in the interval $[-50, 50] \times [-70, 70]$, the altitude is uniformly distributed in the interval [80, 120], the velocity is uniformly distributed in the interval [15, 30], the initial value of the heading angle is 0, the heading angular velocity is uniformly distributed in the interval $[-\pi/2, \pi/2]$, and the climbing velocity is in the interval $[-3, 3]$. The step size in all simulations is $\Delta t = 0.1$ s.

TABLE 2. Parameters of the flocking algorithms.

Parameter	Notation and Value
Expected safe distance between UAVs (m)	$d = 60$
Detection range of the UAV (m)	$r = 110$
Detection range of the UAV to obstacles (m)	$r_O = 100$
Parameters of $\phi_\alpha(q_i), \phi_\beta(q_i)$, $\sigma_1(q_i - q_\gamma)$ and $u^{\beta,\tau}_i$	$\epsilon_\alpha = 1, \epsilon_\beta = 1$ $\epsilon_\gamma = 1, \epsilon_{\beta,\tau} = 1$
Parameter of $\rho_H(q_i)$	$H = 200$
Parameter of ρ_h in u^α_i	$h_\alpha = 0.7$
Parameter of ρ_h in u^β_i	$h_\beta = 0.9$
Control gains of the algorithm (31)	$c_\alpha = 1, c_\beta = 1$ $c_\gamma = 1, c_\tau = 1$
Control gains of the algorithm u^α_i	$c^\alpha_q = 1, c^\alpha_p = 1$
Control gains of the algorithm u^γ_i	$c^\gamma_q = 5, c^\gamma_p = 1, c^\gamma_h = 1$
Control gains of the algorithm $u^{\beta,\tau}_i$	$c^{\beta,\tau}_q = 5, c^{\beta,\tau}_p = 5$

The above parameters and initial state values are for a single UAV. We must also set parameters and states for the UAV swarm, as shown in Table 2. Some of the data in Table 1 and Table 2 are from [5], [18], [26], and [27], and some are from repetitive simulation experiments.

A. FLOCKING AVOIDING OBSTACLES ALGORITHM SIMULATION

In this section, we mainly simulate comparisons of the obstacle avoidance algorithms described in Sections II and III, i.e., the normal method and the proposed method. The obstacle avoidance algorithm in Section II is based on Saber's thesis, and Section III is an improvement on Saber's obstacle avoidance algorithm. The simulation environment has the UAV swarm follow a desired path from [0, 0, 100] to [0, 1500, 100]. The velocity of the swarm is 25 m/s, and the position of the spherical obstacle is [0, 600, 100], whose radius is 50 m.

Figure 5 shows the simulation results for the two algorithms. Series b corresponds to the obstacle avoidance flocking algorithm, in which UAVs do not share obstacle information, with set parameters $c^\beta_q = 17.5$ and $c^\beta_p = 17.5$. Series c corresponds to the obstacle avoidance algorithm with shared obstacle information, which was described in Section III, with set parameters $c^\beta_q = 13$ and $c^\beta_p = 13$. These parameters were obtained through repetitive simulation experiments. The main purpose is to make the

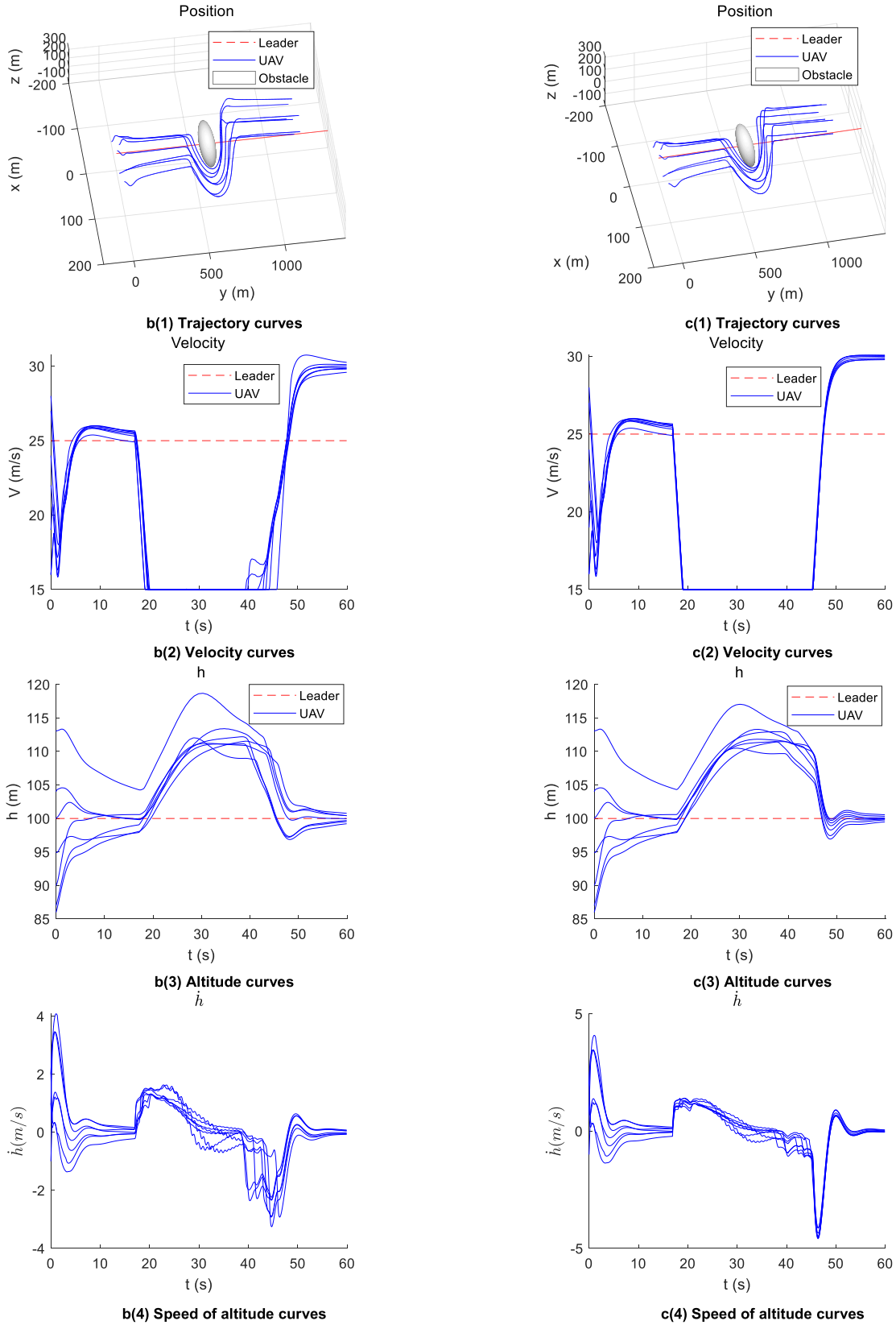
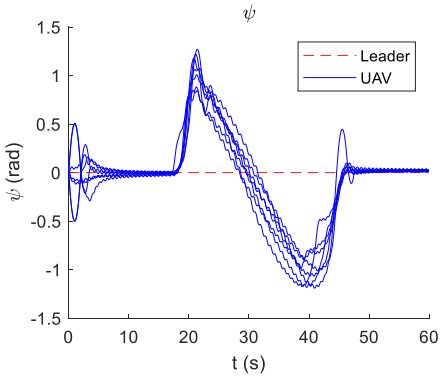
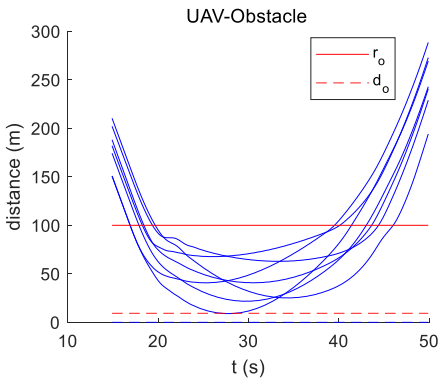


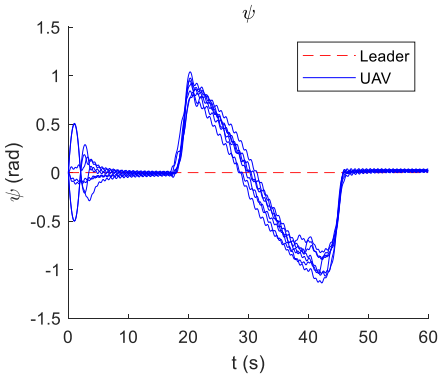
FIGURE 5. Comparison of two kinds of flocking algorithms with obstacle avoidance capability. These figures show the simulation curves of trajectory, velocity, altitude, speed of altitude, heading angle, and distance between each UAV and the obstacle from $t = 15$ s. b(1-6) are the results using the normal flocking algorithm with obstacle avoidance. c(1-6) are the results using the proposed flocking algorithm with shared obstacle information.



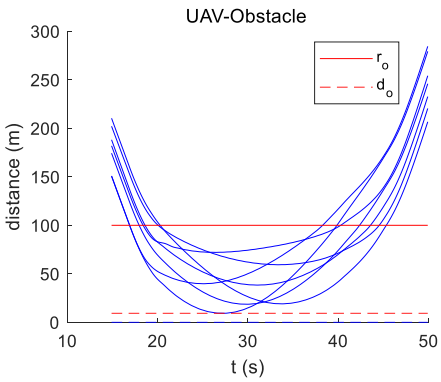
b(5) Heading angle curves



b(6) Distance curves between each UAV and the obstacle beginning from t = 15 s



c(5) Heading angle curves



c(6) Distance curves between each UAV and the obstacle beginning from t = 15 s

FIGURE 5. ***(Continued.)*** **Comparison of two kinds of flocking algorithms with obstacle avoidance capability. These figures show the simulation curves of trajectory, velocity, altitude, speed of altitude, heading angle, and distance between each UAV and the obstacle from $t = 15$ s. b(1-6) are the results using the normal flocking algorithm with obstacle avoidance. c(1-6) are the results using the proposed flocking algorithm with shared obstacle information.**

results of the two algorithms roughly the same, especially the nearest distance between the UAV and the obstacle, as shown in Figures 5-b(6) and 5-c(6). Comparing curves in Figure 5 intuitively, the proposed algorithm has better consistency, especially in speed, speed of altitude, and heading angle. This shows that the proposed algorithm enables UAVs in the swarm to rapidly escape from local minima after encountering obstacles and to maintain consensus with the surrounding UAVs.

Comparing Figures 5-b(1) and 5-c(1), it is clear that the proposed algorithm has a smoother flight path when avoiding the obstacle, especially when flying away from the obstacle. A comparison of velocity curves in Figures 5-b(2) and 5-c(2) shows the difference in UAV velocities for the normal algorithm and the proposed algorithm. For a clearer comparison of the speed differences between UAVs in the swarm, we plotted the maximum and minimum difference curves of UAVs in the two algorithms, as shown in Figures 6-d(1) and 6-e(1), which show the differences between the maximum and the minimum velocities of UAVs from $t = 15$ s to $t = 50$ s. From Figure 6, we see that the peak value of the velocity difference in the normal algorithm reaches approximately 6 m/s after encountering the obstacle, whereas the peak value of velocity difference in the proposed algorithm is approximately 1 m/s, and thus the effect of improvement is obvious. In Figures 6-d(1) and 6-e(1), the speed difference curves have a minimum value of 0 after approximately $t = 20$ s because the UAVs fly around the obstacles at a minimum speed, as shown in Figures 5-b(2) and 5-c(2).

Figures 5-b(4) vs. 5-c(4) and 5-b(5) vs. 5c(5) show that the proposed algorithm has a better consensus on the speed of altitude and heading angle, especially when flying away from the obstacle. To specifically evaluate the performance of the speed of altitude and heading angle, we calculate the first-order absolute center moment and its sum separately as follows.

For $\dot{h}$:

$$\begin{aligned} \bar{e}_{\dot{h}}(\mathrm{t}) &= \frac{1}{N}\sum_{i=1}^{N}\left|\dot{h}_i(t) - \overline{\dot{h}(t)}\right|, \\ \overline{\dot{h}(t)} &= \frac{1}{N}\sum_{i=1}^{N}\dot{h}_i(t), \\ E_{\dot{h}} &= \sum_{t=0}^{T}\bar{e}_{\dot{h}}(t). \end{aligned} \tag{40}$$

For ψ:

$$\begin{aligned} \bar{e}_{\psi}(\mathrm{t}) &= \frac{1}{N}\sum_{i=1}^{N}\left|\psi_i(t) - \overline{\psi(t)}\right|, \\ \overline{\psi(t)} &= \frac{1}{N}\sum_{i=1}^{N}\psi_i(t), \\ E_{\psi} &= \sum_{t=0}^{T}\bar{e}_{\psi}(t). \end{aligned} \tag{41}$$

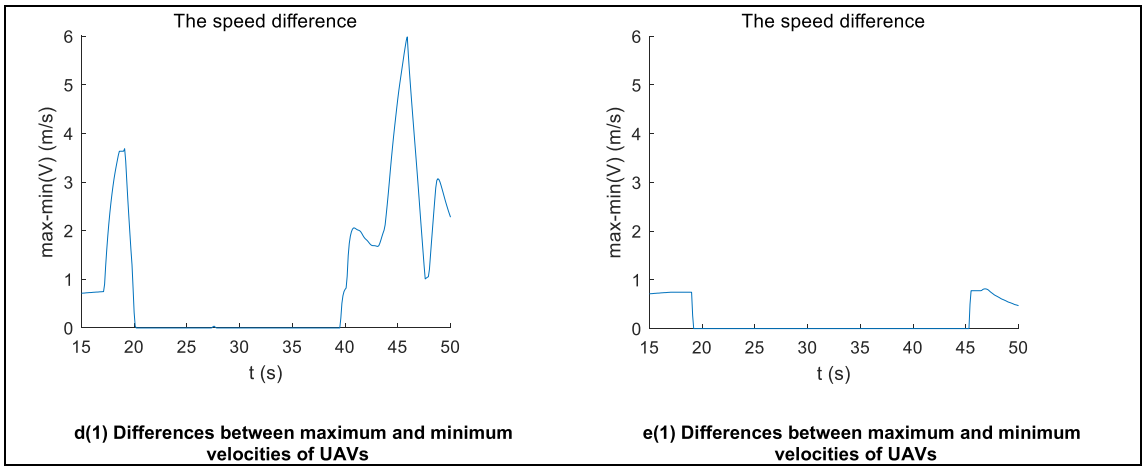


FIGURE 6. Comparison of differences between maximum and minimum velocities of UAVs in the swarm based on two algorithms controlling the UAV swarm while meeting the obstacle. d(1) is the results using the normal flocking algorithm with obstacle avoidance. e(1) is the results using the proposed flocking algorithm with obstacle avoidance.

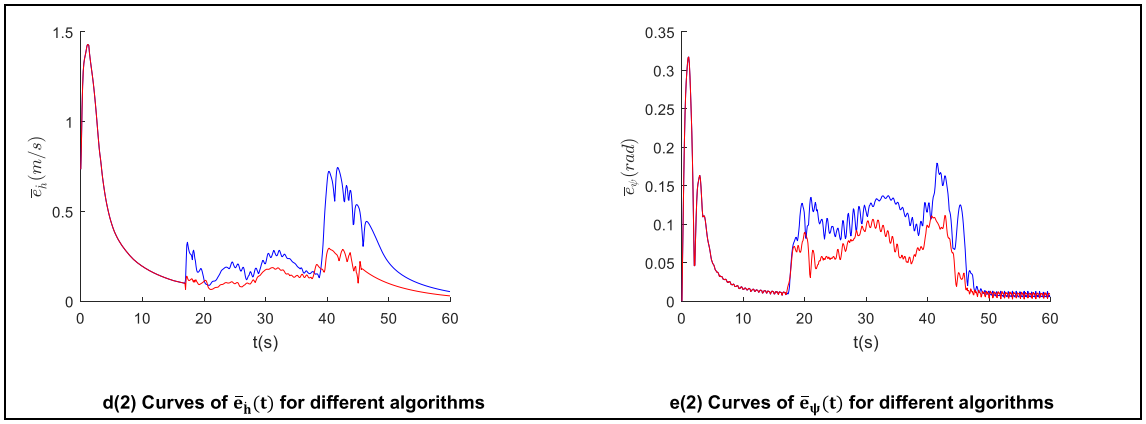


FIGURE 7. Comparison of the first-order absolute center moment of speed of altitude (d(2)) and heading angle (e(2)) of the UAV swarms controlled by two algorithms. Red curves are the results using the proposed flocking algorithm, and blue curves are the results using the normal flocking algorithm.

Figure 7 and Table 3 show the values of $\bar{e}$ and E acquired using the two algorithms. The proposed algorithm has smaller values of $\bar{e}_{\dot{h}}(t)$ and $\bar{e}_\psi(t)$. $E_{\dot{h}}$ of the proposed algorithm improved by 28.60% ((178.16–127.21)/178.16 • 100%) compared with the normal algorithm. E_ψ of the proposed algorithm improved by 26.89% compared with the normal algorithm. This shows that the proposed algorithm allows the swarm a more gentle speed of altitude and heading angle.

The speed of altitude has a direct effect on altitude. The proposed algorithm, with a more stable altitude speed performance, grants a better tracking performance for the swarm's altitude when avoiding the obstacle, as shown in Figures 5-b(3) and 5-c(3).

The speed, speed of altitude, and heading angle were also analyzed. A comparison of Figure 6 and Figure 7 shows that, when the UAV swarm enters the obstacle range and leaves the obstacle range, the UAV swarm controlled by the normal algorithm makes large maneuvering changes, resulting in large speed differences, $\bar{e}_{\dot{h}}(t)$ and $\bar{e}_\psi(t)$. In the UAV swarm controlled by the proposed algorithm, UAVs enter and leave the obstacle range with smoother changes in speed, speed of altitude, and heading angle. The reasons for this difference were explained in Section III. When the UAV swarm controlled by the proposed algorithm has just entered the obstacle range, some UAVs can detect obstacles and some cannot. UAVs that can detect obstacles begin to make obstacle avoidance maneuvers, and the UAVs that cannot detect obstacles are prepared for obstacle avoidance under the function of $u_i^{\beta,\tau}$, such as deceleration and attitude change, so that

TABLE 3. The sum of the first-order absolute center moment.

	Normal method	Proposed method	Percentage
$E_{\dot{h}}$	178.16	127.21	28.60%
E_ψ	43.51	31.81	26.89%

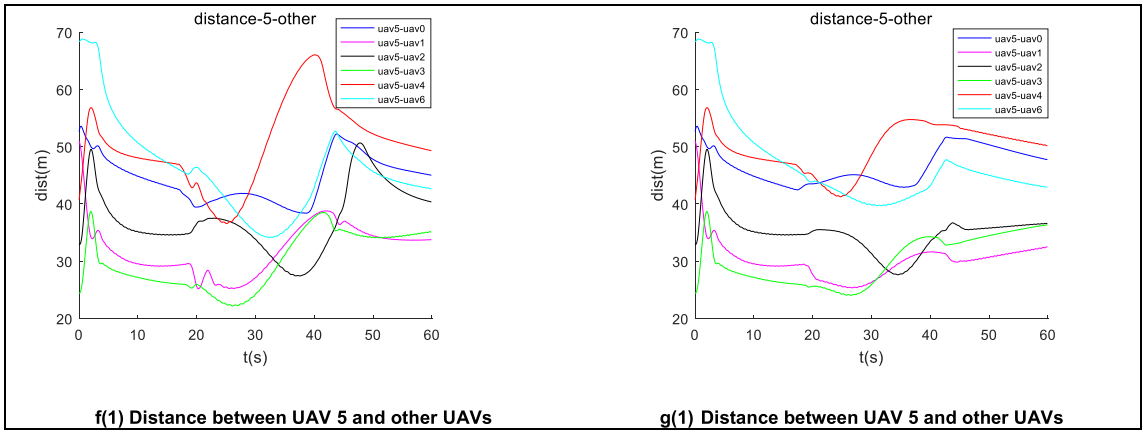


FIGURE 8. Comparison of distances between UAV 5 and other UAVs. f(1) is the results using the normal flocking algorithm with obstacle avoidance. g(1) is the results using the proposed flocking algorithm with obstacle avoidance.

the swarm has better consistency. When the UAV swarm controlled by the normal algorithm enters the obstacle range, UAVs that can detect the obstacle make obstacle avoidance maneuvers quickly, and UAVs that cannot detect the obstacle do not prepare for obstacle avoidance, and thus the swarm does not have good consistency. When the UAV swarm leaves the obstacle, they show the same behaviors. UAVs that have left the obstacle range in the UAV swarm controlled by the proposed algorithm cannot accelerate rapidly under the function of $u_i^{\beta,\tau}$, thus maintaining a good consistency with UAVs within the obstacle range. Therefore, the proposed algorithm allows the UAV swarm to complete the cooperative obstacle avoidance with good consistency.

We compared the distance between UAVs in the two algorithms during the obstacle avoidance process. In Figure 8, we plotted the distance curves between UAV 5 and other UAVs for the two algorithms. The proposed algorithm's distance curves are smoother. To illustrate this point more clearly, we performed simple statistics on the rate of change of distance using equation (42).

$$\overline{dist} = \frac{1}{N-1}\sum_{i=1}^{N-1}\sum_{t=\tau}^{end}\left|\frac{(dist_i(t+\Delta t)-dist_i(t))}{\Delta t}\right|. \tag{42}$$

From Figures 5-b(6) and 5-c(6), it can be seen that the UAV swarm begins to detect the obstacle around $t = 15$ s. Prior to this, the obstacle avoidance algorithm did not start its role. Therefore, we start from $t = 15$ s to calculate the average changes in the distance between UAV 5 and other UAVs. The statistical results are shown in Table 4.

From Table 4, it can be seen that, after the swarm encounters an obstacle, the swarm controlled by the proposed algorithm has a smaller rate of change for distance during obstacle avoidance. $\overline{dist}$ of the proposed algorithm improves by 40.06% compared with the normal algorithm.

Through the above analysis, it can be found that the proposed algorithm has better stability and consistency when avoiding obstacles.

TABLE 4. The average change in distance between UAV 5 and other UAVs.

	Normal method	Proposed method	Percentage
$\overline{\text{dist}}$ (m/s)	373.27	223.75	40.06%

V. CONCLUSION

In this paper, we studied UAV swarm obstacle avoidance-related issues. The proposed multi-agent flocking control algorithm was applied to a UAV swarm, for which the UAV kinematics model and UAV dynamic constraints were considered. We considered the following problems caused by obstacle avoidance for UAV swarms: (1) when the UAV swarm encounters an obstacle, the individual states of the UAVs cannot be stable due to fluctuations in the distance between UAVs and between UAVs and the obstacle; (2) individuals in the swarm become trapped in local minima, resulting in the swarm failing to pass obstacles or having poor consensus. To solve these problems, we proposed a flocking obstacle avoidance algorithm with shared obstacle information. The overarching goal of the algorithm is to improve the normal isolated and local obstacle avoidance algorithm to a global obstacle avoidance algorithm. We improved the shortcomings of the normal algorithm, in which it is easy to fall into local minima and results in poor consensus, and our algorithm ultimately improves the consensus of the UAV swarm while avoiding obstacles. Two simulation scenarios were tested, and the simulation results showed that our algorithm can achieve better consensus when bypassing the obstacle.

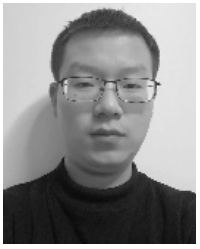
REFERENCES

[1] X. Guo, J. Q. Lu, A. Alsaedi, and F. E. Alsaadi, "Bipartite consensus for multi-agent systems with antagonistic interactions and communication delays," *Phys. A, Stat. Mech. Appl.*, vol. 495, pp. 488–497, Apr. 2018.

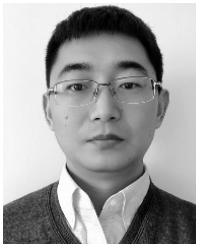
[2] W. Lee and D. Kim, "Autonomous shepherding behaviors of multiple target steering robots," *Sensors*, vol. 17, p. 21, Nov. 2017.

[3] A. Forestiero, "Bio-inspired algorithm for outliers detection," *Multimed. Tools Appl.*, vol. 76, no. 24, pp. 25659–25677, 2017.

[4] J. Wang, I. Ahn, Y. Lu, T. Yang, and G. Staskevich, "A distributed estimation algorithm for collective behaviors in multiagent systems with applications to unicycle agents," *Int. J. Control Autom. Syst.*, vol. 15, pp. 2829–2839, 2017.
[5] R. Olfati-Saber, "Flocking for multi-agent dynamic systems: Algorithms and theory," *IEEE Trans. Autom. Control*, vol. 51, no. 3, pp. 401–420, Mar. 2006.
[6] H. Su, X. Wang, and Z. Lin, "Flocking of multi-agents with a virtual leader," *IEEE Trans. Autom. Control*, vol. 54, no. 2, pp. 293–307, Feb. 2009.
[7] F. Sun and K. Turkoglu, "Distributed real-time non-linear receding horizon control methodology for multi-agent consensus problems," *Aerosp. Sci. Technol.*, vol. 63, pp. 82–90, Apr. 2017.
[8] C. W. Reynolds, "Flocks, herds and schools: A distributed behavioral model," in *Proc. Conf. Comput. Graph. Interact. Techn.*, 1987, pp. 25–34.
[9] S. A. P. Quintero, G. E. Collins, and J. P. Hespanha, "Flocking with fixed-wing UAVs for distributed sensing: A stochastic optimal control approach," in *Proc. IEEE Amer. Control Conf.*, New York, NY, USA, Jun. 2013, pp. 2025–2031.
[10] L. He, P. Bai, X. Liang, J. Zhang, and W. Wang, "Feedback formation control of UAV swarm with multiple implicit leaders," *Aerosp. Sci. Technol.*, vol. 72, pp. 327–334, Jan. 2018.
[11] X. Mao, H. Zhang, and Y. Wang, "Flocking of quad-rotor UAVs with fuzzy control," *ISA Trans.*, vol. 74, pp. 185–193, Mar. 2018, doi: 10.1016/j.isatra.2018.01.024.
[12] C. S. Do, "Multi-agent based design of autonomous UAVs for both flocking and formation flight," *J. Korea Navigat. Inst.*, vol. 21, pp. 521–528, 2017.
[13] B. Di, R. Zhou, and H. B. Duan, "Potential field based receding horizon motion planning for centrality-aware multiple UAV cooperative surveillance," *Aerosp. Sci. Technol.*, vol. 46, pp. 386–397, Oct. 2015.
[14] H. X. Qiu and H. B. Duan, "Multiple UAV distributed close formation control based on in-flight leadership hierarchies of pigeon flocks," *Aerosp. Sci. Technol.*, vol. 70, pp. 471–486, Nov. 2017.
[15] C. Kownacki and L. Ambroziak, "Local and asymmetrical potential field approach to leader tracking problem in rigid formations of fixed-wing UAVs," *Aerosp. Sci. Technol.*, vol. 68, pp. 465–474, Sep. 2017.
[16] H. Qiu and H. Duan, "Pigeon interaction mode switch-based UAV distributed flocking control under obstacle environments," *ISA Trans.*, vol. 71, pp. 93–102, Nov. 2017.
[17] C. Kownacki and D. Ołdziej, "Fixed-wing UAVs flock control through cohesion and repulsion behaviours combined with a leadership," *Int. J. Adv. Robot. Syst.*, vol. 13, no. 1, p. 36, 2016.
[18] X. Zhang and H. Duan, "Altitude consensus based 3D flocking control for fixed-wing unmanned aerial vehicle swarm trajectory tracking," *Proc. Inst. Mech. Eng., G, J. Aerosp. Eng.*, vol. 230, no. 14, pp. 2628–2638, 2016.
[19] J. Li, W. Zhang, H. Su, Y. Yang, and H. Zhou, "Coordinated obstacle avoidance with reduced interaction," *Neurocomputing*, vol. 139, pp. 233–245, Sep. 2014.
[20] D. Sakai, H. Fukushima, and F. Matsuno, "Flocking for multirobots without distinguishing robots and obstacles," *IEEE Trans. Control Syst. Technol.*, vol. 25, no. 3, pp. 1019–1027, May 2017.
[21] S. Iovino, A. R. Vetrella, G. Fasano, D. Accardo, and A. Savvaris, "Implementation of a distributed flocking algorithm with obstacle avoidance capability for UAV swarming," in *Proc. AIAA Inf. Syst.-AIAA Infotech Aerosp., AIAA SciTech Forum, (AIAA)*, 2017. doi: 10.2514/6.2017-0878.
[22] Q. Wang, H. Fang, J. Chen, Y. Mao, and L. Dou, "Flocking with obstacle avoidance and connectivity maintenance in multi-agent systems," in *Proc. CDC*, vol. 8350, 2012, pp. 4009–4014.
[23] Q. Luo and H. Duan, "Distributed UAV flocking control based on homing pigeon hierarchical strategies," *Aerosp. Sci., Technol.*, vol. 70, pp. 257–264, Nov. 2017.
[24] E. D. Vries and K. Subbarao, "Cooperative control of swarms of unmanned aerial vehicles," in *Proc. AIAA Aerosp. Sci. Meeting Including New Horizons Forum Aerosp. Expo.*, 2013, pp. 89–94.
[25] J. Sun, J. Tang, and S. Lao, "Collision avoidance for cooperative UAVs with optimized artificial potential field algorithm," *IEEE Access*, vol. 5, pp. 18382–18390, 2017.
[26] M. Zhang and C. Wei, "Multi-UAV formation control method based on edge Laplacian consensus," *Scientia SinicaTechnologica*, vol. 47, no. 3, pp. 259–265, 2017.
[27] R. W. Beard and T. W. Mclain, *Small Unmanned Aircraft: Theory and Practice*. Princeton, NJ, USA: Princeton Univ. Press, 2012.



WEIWEI ZHAO received the B.E. degree from the North China University of Water Resources and Electric Power, Zhengzhou, China, in 2015. He is currently pursuing the Ph.D. degree with the University of Chinese Academy of Sciences, and the Changchun Institute of Optics, Fine Mechanics and Physics, Chinese Academy of Sciences. His current research interests include reinforcement learning, multi-agent control systems, and cooperative unmanned aerial vehicles' flight control.



HAIRONG CHU received the B.E. degree from Jilin University, in 2005, and the Ph.D. degree from the University of Chinese Academy of Sciences and the Institute of Optics, Fine Mechanics and Physics, in 2010, where he is currently a Research Fellow. His research interests include strapdown inertial navigation, guidance, and control systems.

MINGYUE ZHANG received the M.E. degree from the Huazhong University of Science and Technology, in 2011, and the Ph.D. degree from the University of Chinese Academy of Sciences and the Changchun Institute of Optics, Fine Mechanics and Physics, Chinese Academy of Sciences, where she is currently an Assistant Research Fellow. Her research interests include intelligent control of nonlinear systems and its application of advanced control and optimization techniques to aerospace.

TINGTING SUN received the B.E. degree from Jilin University, in 2011, and the Ph.D. degree from the University of Chinese Academy of Sciences and the Changchun Institute of Optics, Fine Mechanics and Physics, in 2016. Her current research interests include strapdown guidance systems and inertial navigation systems.

LIHONG GUO received the B.E. degree from the Changchun University of Science and Technology, in 1986, and the M.Sc. and Ph.D. degrees from the Changchun Institute of Optics, Fine Mechanics and Physics, in 1998 and 2003, respectively, where she is currently a Professor. Her current research interests include information fusion and computer science.