Hindawi
International Journal of Intelligent Systems
Volume 2023, Article ID 1493299, 13 pages
https://doi.org/10.1155/2023/1493299

Research Article

Obstacle Boundary Point and Expected Velocity-Based Flocking of Multiagents with Obstacle Avoidance

Jianhui Wu,[1,2,3] **Yuanfa Ji**,[1,3] **Xiyan Sun**,[1,3] **and Weibin Liang**[1,3]

[1] *School of Information and Communication, Guilin University of Electronic Technology, Guilin, China*
[2] *School of Information Science and Technology, Hunan Institute of Science and Technology, Yueyang, China*
[3] *Guangxi Key Laboratory of Precision Navigation Technology and Application, Guilin University of Electronic Technology, Guilin, China*

Correspondence should be addressed to Yuanfa Ji; jiyuanfa@163.com

Received 23 October 2022; Revised 5 January 2023; Accepted 21 January 2023; Published 20 February 2023

Academic Editor: Mohammad R. Khosravi

Obstacle avoidance is a key technology of multiagents flocking control. However, most existing research studies assume that the agent can obtain global information about obstacles and have specific constraints on the shape and boundary of obstacles, which easily limit their practical applications. To relax these constraints, we assume that the agent can only perceive the position of obstacle boundary points within its sensing radius and propose an obstacle boundary point and an expected velocity-based flocking algorithm of multiagents with obstacle avoidance. In this algorithm, the attraction/repulsion potential is designed to avoid collisions between agents and between the agent and obstacle boundary points, the expected velocity is designed to tow the agent to move along the boundary of obstacles, and the virtual leader is designed to lead all agents to realize the group objective. Finally, the sufficient conditions that the agent does not collide are demonstrated, and the performance of the proposed algorithm is further verified through simulations.

1. Introduction

Flocking is an orderly motion phenomenon generated by relatively simple local interaction among multiagents, which is very common in nature, such as the migration of birds [1, 2], foraging of fish schools [3, 4], avoiding predators of deers [5], and swimming of bacteria [6, 7]. Multiagents often encounter some obstacles in the process of flocking motion [8–10]. Therefore, obstacle avoidance is a key technology problem to be solved in the research of flocking control.

At present, the research on flocking with obstacle avoidance mainly concentrate on the collision cone approach, the velocity obstacle approach, the numerical solution approach, and the potential field approach. The main characteristics of the above four approaches are summarized in Table 1. The collision cone approach detects the collision risk through the collision cone and converts the relative velocity vector into the obstacle-avoidance vector to solve the obstacle avoidance trajectory when the collision risk is detected. The velocity obstacle approach defines the space obtained by translating the collision cone along the obstacle velocity vector as the velocity obstacle space and realizes obstacle avoidance by bypassing the velocity obstacle space along the shortest path. Obviously, both the collision cone approach and the velocity obstacle approach solve the obstacle avoidance problem based on the relative geometric relationship between the agent and obstacles, but the difficulty in constructing and solving the relative geometric relationship will increase significantly with increasing the scale of agents or obstacles. In the numerical solution approach, the collision process between the agent and obstacles is constructed as a mathematical model under the motion constraints, and the optimal obstacle-avoidance trajectory at the current state is obtained by the intelligent algorithm [19, 20].

The potential field approach generally maintains the distance between the agent and obstacles through the position-based virtual potential field and guides the agent to bypass obstacles through the velocity-based virtual potential field. Olfati-Saber assumed that all agents can completely obtain the smooth shape, position, and motion information of a spherical obstacle or infinite wall and presented a flocking algorithm with obstacle avoidance [25]. The algorithm assumes that the velocity of the obstacle is the projection velocity of the agent on the obstacle, which enables the agent to bypass the obstacle along the tangent direction of the obstacle but also restricts the obstacle to a hyperplane or convex obstacle with a smooth boundary. Li et al. proposed a filling scheme to change the nonconvex obstacle into the convex obstacle to solve the problem of obstacle avoidance under nonconvex obstacles [26]. The scheme can make the agent bypass the concave obstacle, but it still needs to obtain global information about obstacles. This constraint limits its practical applications. Therefore, relaxing the constraints of obstacle information, shape, and boundary is very challenging for the research of the flocking algorithm of multiagents with obstacle avoidance.

The Bug algorithm is a kind of obstacle avoidance algorithm for an agent without knowing the global information of a map and obstacle shape, such as the Bug2 algorithm [27, 28], the IBug algorithm [29], the FuzzyBug algorithm [30], and the RandomBug algorithm [31, 32]. In this kind of algorithm, the agent firstly moves along the boundary of obstacles when encountering some, then leaves these obstacles based on certain judgment criteria, and then continues to move towards the target in a straight line. Moving along the boundary of obstacles is an efficient scheme to bypass complex obstacles [33]. Based on this, we attempt to convert the expected position of the agent at the next time into the expected velocity of the agent for all obstacle boundary points within its sensing radius and make the agent that perceives the obstacles preventing it following bypass obstacles follow along the boundary of obstacles using the expected velocity. Meanwhile, we regard the agent, obstacle boundary point, and virtual leader as α-agent, β-agent, and γ-agent, respectively, attempt to ensure that there is no collision between α-agents and between α-agent and β-agent at any time through the attractive/repulsive potential, and lead all α-agents to realize the group objective following through the γ-agent. In addition, to relax the constraint of obstacle information, we assume that α-agent can only perceive the position of β-agents within its sensing radius. Unlike the Bug algorithm, this study aims to realize collision avoidance and group objective following through the attractive/repulsive potential, expected velocity, and γ -agent, which is to relax the constraints of the flocking algorithm with obstacle avoidance based on obstacle information, shape, and boundary.

The main contributions of this paper can be summarized as follows: first, a novel method is established to calculate the expected velocity of α-agent for all β-agents within its sensing radius. Second, a novel obstacle boundary point and expected velocity-based flocking algorithm of multiagents with obstacle avoidance is designed to relax the constraints of obstacle information, shape, and boundary. Third, the sufficient conditions that α-agent does not collide in the proposed algorithm are obtained.

The rest of this paper is organized as follows: In Section 2, we formulate the multiagent motion model with velocity limit and introduce some preliminaries. Section 3 describes the design of the obstacle boundary point and the expected velocity-based flocking algorithm of multiagents with obstacle avoidance. The property analysis of the proposed algorithm is presented in Section 4. The simulations are provided to verify the theoretical results in Section 5. Finally, the main conclusions are drawn in Section 6.

2. Preliminaries

2.1. Problem Formulation. Consider N α-agents moving in the m-dimensional $(e.g., m = 2, 3)$ Euclidean space. Let $q_i \in \mathbb{R}^m$ denote the position vector of α-agent i at the current time, $p_i \in \mathbb{R}^m$ is the velocity vector of α-agent i at the current time, and $u_i \in \mathbb{R}^m$ is the control input vector of α-agent i at the current time. The movement of α-agent i is described as [26]

$$\dot{q}_i = p_i, \dot{p}_i = u_i, \forall i \in \{1, \cdots, N\}. \tag{1}$$

In the practical implementation, the velocity of α-agent i is limited. Therefore, the movement of α-agent i can be rewritten as

$$\dot{q}_i = sat\left(p_i, p_i^{\max}\right), \dot{p}_i = u_i, \forall i \in \{1, \cdots, N\}, \tag{2}$$

where $p_i^{\max}$ denotes the maximum velocity of α-agent i, $\mathrm{sat}(p_i, p_i^{\max}) = \delta p_i + (1 - \delta) p_i p_i^{\max}/\|p_i\|$, specifically $\delta = 1$ if $\|p_i\| \leq p_i^{\max}$ and $\delta = 0$ otherwise, and $\|\cdot\|$ is the Euclidean norm.

To relax the constraint of obstacle information, we assume that the α-agent can perceive the position of obstacle boundary points within the sensing radius and obtain the position and velocity information of other α -agents within the sensing radius by information interaction. If the objects within the sensing radius of α -agent i have other α-agents and obstacles, N_i^{α} is the set of α-agents within the sensing radius r of α-agent i at the current time, N_i^{β} is the set of obstacle boundary points within the sensing radius r of α-agent i at the current time, then the object set N_i within the sensing radius r of α -agent i can be defined as

$$\begin{cases} N_i = N_i^{\alpha} \cup N_i^{\beta}, \\ N_i^{\alpha} = \left\{ j \in \{1, \cdots, N\} \colon \left\| q_j - q_i \right\| \leq r, i \neq j \right\}, \\ N_i^{\beta} = \left\{ k \in O_k \colon \left\| q_k - q_i \right\| \leq r \right\}, \end{cases} \tag{3}$$

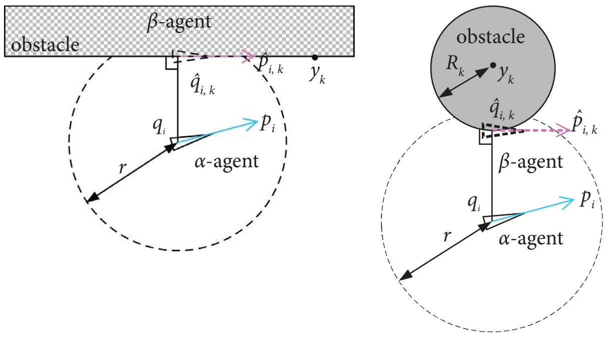
where q_k denotes the position vector of obstacle boundary point k, and O_k is the set of obstacle boundary points.

As in previous studies [25], we introduce a γ-agent to represent a group objective. The movement of γ-agent is described as

$$\dot{q}_{\gamma} = p_{\gamma}, \dot{p}_{\gamma} = u_{\gamma}, \tag{4}$$

Table 1: Summary of related research on flocking with obstacle avoidance.

Approach	Reference	Control input	Large-scale agent	Obstacle constraint	Experimental scenario
Collision cone approach	Boardman et al. [11] (2021)	Position and velocity	No	Yes	Static and dynamic
	Bhattacharjee et al. [12] (2021) and Douthwaite et al. [13] (2019)	Velocity	No	Yes	Dynamic
	Chakravarthy and Ghose [14] (2020)	Acceleration	No	Yes	Dynamic
	Xu et al. [15] (2020)	Velocity	No	Yes	Static and dynamic
Velocity obstacle approach	Chen et al. [16] (2020)	Velocity	No	Yes	Static and dynamic
	López et al. [17] (2020) and Fuad et al. [18] (2021)	Linear and angular velocities	No	Yes	Static and dynamic
Numerical solution approach	He et al. [19, 20] (2019 and 2022)	Surge, sway and yaw accelerations	No	Yes	Dynamic
Potential field approach	Sakai et al. [21] (2016), Nath et al. [22] (2022), Dang et al. [23] (2019), and Wang et al. [24] (2014)	Acceleration	Yes	Yes	Static and dynamic



(a) (b)

Figure 1: Position and velocity of β-agent (previous algorithm). (a) Infinite wall. (b) Spherical obstacle.

where $q_\gamma \in \mathbb{R}^m$ denotes the position vector of γ-agent at the current time, $p_\gamma \in \mathbb{R}^m$ is the velocity vector of γ-agent at the current time, and $u_\gamma \in \mathbb{R}^m$ is the control input vector of γ -agent at the current time.

2.2. Overview of Previous Flocking Algorithm with Obstacle Avoidance. The proposed algorithm is constructed by modifying the control input in [25], which considers relaxing the constraints of obstacle information, shape, and boundary. For the sake of clarity, this section gives a brief overview of the flocking algorithm with obstacle avoidance in [25].

In Figure 1, when an α-agent i perceives an obstacle k within the sensing radius r, it will generate a virtual β-agent k at the closest point on the obstacle. The position $\widehat{q}_{i,k}$ of β-agent k is the projection position of α-agent i on the obstacle k, and the velocity $\widehat{p}_{i,k}$ is the projection velocity of α-agent i on the obstacle k. In the case where the obstacle is an infinite wall, the position $\widehat{q}_{i,k}$ and velocity $\widehat{p}_{i,k}$ of β-agent k are given by [25]

$$\widehat{q}_{i,k} = Pq_i + (1 - P)y_k, \widehat{p}_{i,k} = Pp_i, \tag{5}$$

where y_k is the passing point of infinite wall boundary, $P = I - \mathrm{a}_k\mathrm{a}_k^T$ is the projection matrix, and $\mathrm{a}_k = q_i - y_k/\|q_i - y_k\|$.

In the case where the obstacle is a spherical obstacle, the position $\widehat{q}_{i,k}$ and velocity $\widehat{p}_{i,k}$ of β-agent k are determined by [25]

$$\widehat{q}_{i,k} = \mu q_i + (1 - \mu)y_k, \widehat{p}_{i,k} = \mu P p_i, \tag{6}$$

where y_k is the center of spherical obstacle, R_k is the radius of spherical obstacle, and $\mu = R_k/\|q_i - y_k\|$.

Besides, following the results in [25], the control input u_i is composed of (α, α) interaction term u_i^α, (α, β) interaction term u_i^β, and (α, γ) interaction term u_i^γ. The control input u_i is described as

$$u_i = u_i^\alpha + u_i^\beta + u_i^\gamma. \tag{7}$$

The (α, α) interaction term u_i^α in formula (7) is defined as

$$u_i^\alpha = \underbrace{-c_1^\alpha \sum_{j \in N_i^\alpha} \nabla_{q_i}\psi_\alpha\left(\left\|q_j - q_i\right\|_\sigma\right)}_{\text{gradi ent–based term}} + \underbrace{c_2^\alpha \sum_{j \in N_i^\alpha} a_{ij}(q)\left(p_j - p_i\right)}_{\text{consensus term}}, \tag{8}$$

where $c_1^\alpha > 0$ and $c_2^\alpha > 0$ are the feedback gains, $\nabla_{q_i}\psi_\alpha(\|q_j - q_i\|_\sigma)$ is the gradient of the pairwise attractive/repulsive potential $\psi_\alpha(\|q_j - q_i\|_\sigma)$ between α-agent i and α -agent j at position q_i, $\psi_\alpha(\|q_j - q_i\|_\sigma) = \int_{d_\alpha}^{\|q_j - q_i\|_\sigma} \phi_\alpha(s)\mathrm{d}s$, $\phi_\alpha(s) = \rho_h(s/r_\alpha)\phi(s - d_\alpha)$, $\phi(z) = 1/2[(a + b)\sigma_1(z + c) + (a - b)]$, $\sigma_1(z) = z/\sqrt{1 + \|z\|^2}$, $b \geq a > 0$, $c = |a - b|/\sqrt{4ab}$, $d_\alpha = \|d\|_\sigma$, $a_{ij}(q) = \rho_h(\|q_j - q_i\|_\sigma/r_\alpha)$, $r_\alpha = \|r\|_\sigma$, $\|z\|_\sigma = 1/\varepsilon(\sqrt{1 + \varepsilon\|z\|^2} - 1)$, $\varepsilon > 0$, and the bump function $\rho_h(z)$ for $h \in (0, 1)$ is defined as [21]

$$\rho_h(z) = \begin{cases} 1, & \text{if } z \in [0, h), \\ \dfrac{1}{2}\left[1 + \cos\left(\dfrac{(z - h)\pi}{1 - h}\right)\right], & \text{if } z \in [h, 1], \\ 0, & \text{otherwise.} \end{cases} \tag{9}$$

The (α, β) interaction term u_i^β in formula (7) is defined as

$$u_i^\beta = -c_1^\beta \sum_{k \in N_i^\beta} \nabla_{q_i}\psi_\beta\left(\|\widehat{q}_{i,k} - q_i\|_\sigma\right) + c_2^\beta \sum_{k \in N_i^\beta} b_{i,k}(q)\left(\widehat{p}_{i,k} - p_i\right), \tag{10}$$

where $c_1^\beta > 0$ and $c_2^\beta > 0$ are the feedback gains, $b_{i,k}(q) = \rho_h(\|\widehat{q}_{i,k} - q_i\|_\sigma/d_\beta)$, and $\sum_{k \in N_i^\beta} \nabla_{q_i}\psi_\beta(\|\widehat{q}_{i,k} - q_i\|_\sigma)$ is the gradient of the pairwise attractive/repulsive potential $\psi_\beta(\|\widehat{q}_{i,k} - q_i\|_\sigma)$ between α-agent i and β-agent k at position q_i, $\psi_\beta(\|\widehat{q}_{i,k} - q_i\|_\sigma) = \int_{d_\beta}^{\|\widehat{q}_{i,k} - q_i\|_\sigma} \phi_\beta(s)\mathrm{d}s$, $\phi_\beta(s) = \rho_h(s/d_\beta)(\sigma_1(z - d_\beta) - 1)$, $d_\beta = \|d'\|_\sigma$.

The (α, γ) interaction term u_i^γ in formula (7) is defined as

$$u_i^\gamma = -c_1^\gamma\sigma_1\left(q_i - q_\gamma\right) - c_2^\gamma\left(p_i - p_\gamma\right), \tag{11}$$

where $c_1^\gamma > 0$ and $c_2^\gamma > 0$ are the feedback gains.

3. Algorithm Design

In this section, we design an obstacle boundary point and an expected velocity-based flocking algorithm of multiagents with obstacle avoidance, which seeks to modify the gradient-based terms in formulas (8) and (10) and the consensus term in formula (10) to relax the constraints of obstacle shape and

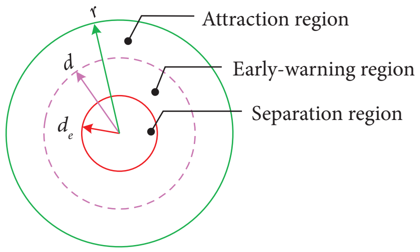


Figure 2: The sensing radius of α-agent.

boundary in the previous flocking algorithm with obstacle avoidance.

3.1. Gradient-Based Term. A distinguishing feature of the proposed algorithm as compared with the previous flocking algorithm with obstacle avoidance in Section 2 is that the position of β-agent is not the projection position of α-agent on an obstacle surface but the position of the obstacle boundary point within the sensing radius of α-agent. Each obstacle boundary point within the sensing radius of α-agent corresponds to a β -agent. When the distance between α-agents or the distance between α-agent and β-agent is less than the minimum early-warning distance d_e, the possibility of collision between them may be greatly increased. To further reduce the possibility of collision, we divide the sensing radius region of α-agent into a separation region, an early-warning region, and an attraction region (see Figure 2), and define a nonnegative subsection function to adjust the weight of attractive/repulsive potential in these three regions.

The nonnegative subsection function $\eta_1^\alpha(d_{ij})$ is defined as

$$\eta_1^\alpha(d_{ij}) = \begin{cases} \tau_a^\alpha, & d_{ij} \in (d, r], \\ \tau_e^\alpha, & d_{ij} \in [d_e, d], \\ \tau_s^\alpha, & d_{ij} \in (0, d_e), \end{cases} \tag{12}$$

where τ_a^α, τ_e^α, and τ_s^α are the weight factors, $\tau_s^\alpha \ge \tau_e^\alpha \ge \tau_a^\alpha \ge 0$, $d_e = d_{\min} + 2p_i{}^{\max}\Delta t$, $d_{\min}$ is the minimum safe distance, and $d_{ij} = \|q_j - q_i\|$ is the distance between α-agent i and α -agent j.

The gradient-based term ω_i^α in formula (8) can be defined as

$$\omega_i^\alpha = -c_1^\alpha \sum_{j\in N_i^\alpha} \eta_1^\alpha(d_{ij})\nabla_{q_i}\psi_\alpha\left(\left\|q_j - q_i\right\|_\sigma\right). \tag{13}$$

Let the attractive/repulsive potential $\psi_\beta(\|q_k - q_i\|_\sigma)$ between α-agent i and β-agent k be similar to $\psi_\alpha(\|q_j - q_i\|_\sigma)$ and $\eta_1^\beta(d_{i,k})$ be similar to $\eta_1^\alpha(d_{ij})$, then the gradient-based term ω_i^β in formula (10) can be described as

$$\omega_1^\beta = -c_1^\beta \sum_{k\in N_i^\beta} \eta_1^\beta(d_{ik})\nabla_{q_i}\psi_\beta\left(\left\|q_k - q_i\right\|_\sigma\right), \tag{14}$$

where $d_{i,k} = \|q_k - q_i\|$ is the distance between α-agent i and β -agent k.

3.2. Consensus Term. Another distinguishing feature of the proposed algorithm is that $\widehat{p}_{i,k}$ in formula (10) is not the velocity vector of β-agent, but is the expected velocity of α-agent for all β -agents. If the expected velocity of α-agent for all β-agents at the current time is equal to the distance from the current position of α-agent to the expected position of α-agent at the next time divided by step time Δt, then the key to the proposed algorithm is determining the expected position of α-agent.

Before presenting the method to determine the expected position of α-agent, we define the split point S_i of α-agent i as the position where α-agent i was at the rejoin point R_i or did not perceive the obstacles preventing it from following γ-agent at the previous time, but it perceives the obstacles preventing it from following at the current time. In addition, we define the rejoin point R_i of α-agent i as the position where the distance between α-agent i and γ-agent is less than or equal to the distance between the split point S_i and γ-agent. The generation process of the split point S_i and rejoin point R_i are shown in Figure 3.

To make the expected position choice of α-agent directional, we translate the coordinate system to make the origin of the coordinate system become the position of α-agent, rotate the coordinate system to make the positive direction of the x-axis point to the split point when α-agent is located in the open interval between the split point and rejoin point, and rotate the coordinate system to make the positive direction of the x-axis point to γ-agent when α-agent is located in the other interval. The expected position determination method of α-agent is defined as follows in two cases

(1) If the α-agent i is located at the split point (A1 scenario), the open interval between the split point and rejoin point (A2 scenario), or the rejoin point where it is prevented from following by some obstacles (A3 scenario) in Figure 4, we use the phase comparison method to determine the expected position of the α-agent i. Suppose that M_i is the number of obstacle boundary points perceived by α-agent i at the current time, $\theta_{i,k}(m)$ is the phase between α-agent i and obstacle boundary point k at sequence number m, θ_i is the phase set between α-agent i and obstacle boundary points perceived by α-agent i, $\theta_i = \text{sort}\left\{\theta_{i,k}(m), \forall k \in N_i^\beta \text{ and } m \in \{1, \cdots, M_i\}\right\}$, sort$\{\cdot\}$ is the ascending function, and $\Delta\theta$ is the discrimination threshold of phase discontinuity. If α-agent i chooses to bypass obstacles based on counterclockwise motion, the phases are compared from small to large. When $\theta_{i,k}(m)$ at sequence number m minus $\theta_{i,j}(m+1)$ at sequence number $m+1$ is greater than or equal to $\Delta\theta$, the expected position vector $\widehat{q}_i$ of α-agent i is defined as the position vector of obstacle boundary point k corresponding to $\theta_{i,k}(m)$. It can be expressed as

$$\widehat{q}_i = q_k, \forall\theta_{i,k}(m) - \theta_{i,j}(m+1) \ge \Delta\theta \text{ and } m = 1 \longrightarrow M_i, \tag{15}$$

where q_k is the position vector of obstacle boundary point k, $1 \longrightarrow M_i$ is the sequence number increasing

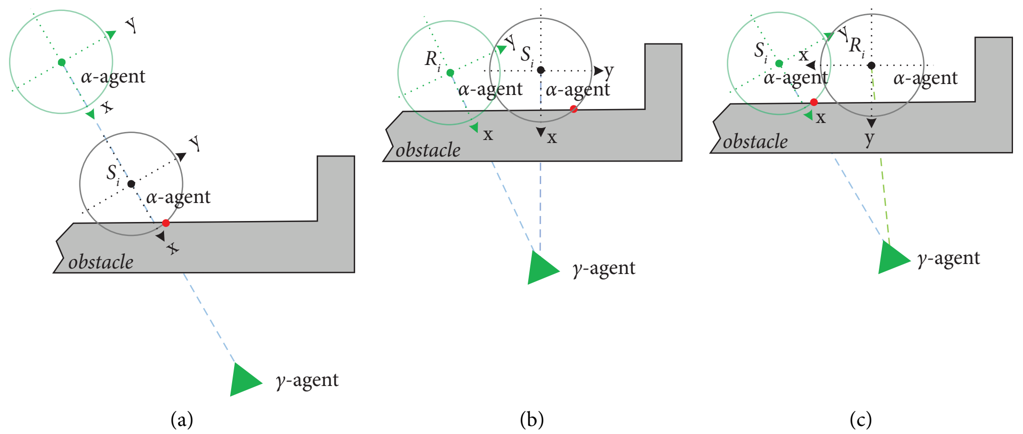


Figure 3: Generation process of split point and rejoin point. (a) A split point generation example in which α-agent did not perceive the obstacles preventing it from following at the previous time. (b) Another split point generation example in which α-agent was at the rejoin point at the previous time. (c) A rejoin point generation example.

from 1 to M_i. Similarly, if α-agent i chooses to bypass obstacles based on clockwise motion, the phases are compared from large to small. When $\theta_{i,k}(m)$ at sequence number m minus $\theta_{i,j}(m-1)$ at sequence number $m-1$ is greater than or equal to $\Delta\theta$, the expected position vector $\widehat{q}_i$ of α-agent i is defined as the position vector of obstacle boundary point k corresponding to $\theta_{i,k}(m)$. It can be expressed as

$$\widehat{q}_i = q_k, \forall\theta_{i,k}(m) - \theta_{i,j}(m-1) \geq \Delta\theta \text{ and } m = M_i \longrightarrow 1, \tag{16}$$

where $M_i \longrightarrow 1$ denotes the sequence number decreasing from M_i to 1.

(2) If the α-agent i perceiving some obstacles is located in the other interval (A4 Scenario), we define the expected position vector $\widehat{q}_i$ of α-agent i as the position vector of γ-agent. It can be expressed as

$$\widehat{q}_i = q_\gamma. \tag{17}$$

Thus, the consensus term ϖ_i^β in formula (10) can be modified as

$$\varpi_i^\beta = c_2^\beta \sum_{k\in N_i^\beta} b_{i,k}(q)\left(\widehat{p}_{i,k} - p_i\right), \tag{18}$$

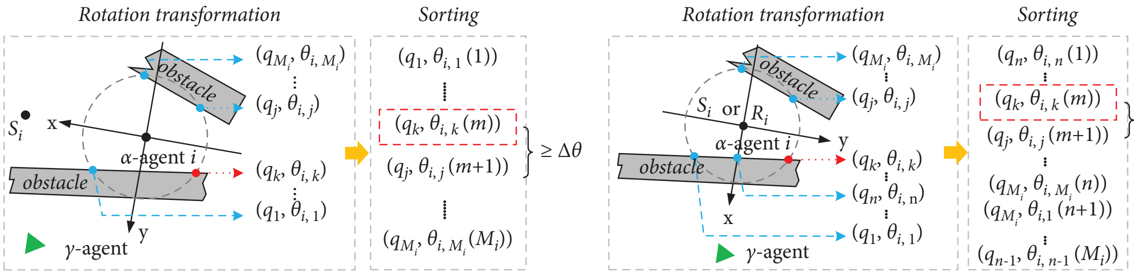
where the expected velocity vector $\widehat{p}_{i,k} = \widehat{q}_i - q_i/\Delta t$ for all β-agents within the sensing radius of α-agent i, and the expected position vector $\widehat{q}_i$ under four scenarios are summarized in Table 2.

In addition, the (α, γ) interaction term u_i^γ of the proposed algorithm is defined as $u_i^\gamma = -c_1^\gamma(q_i - q_\gamma) - c_2^\gamma(p_i - p_\gamma)$. Using formulas (7), (13), (14), and (18), the control input u_i of the proposed algorithm can be expressed as

$$\begin{cases} u_i = u_i^\alpha + u_i^\beta + u_i^\gamma, \\ u_i^\alpha = -c_1^\alpha \displaystyle\sum_{j\in N_i^\alpha} \eta_1^\alpha(d_{ij})\nabla_{q_i}\psi_\alpha\left(\left\|q_j - q_i\right\|_\sigma\right) + c_2^\alpha \sum_{j\in N_i^\alpha} a_{ij}(q)\left(p_j - p_i\right), \\ u_i^\beta = -c_1^\beta \displaystyle\sum_{k\in N_i^\beta} \eta_1^\beta(d_{ik})\nabla_{q_i}\psi_\beta\left(\left\|q_k - q_i\right\|_\sigma\right) + c_2^\beta \sum_{k\in N_i^\beta} b_{i,k}(q)\left(\widehat{p}_{i,k} - p_i\right), \\ u_i^\gamma = -c_1^\gamma\left(q_i - q_\gamma\right) - c_2^\gamma\left(p_i - p_\gamma\right). \end{cases} \tag{19}$$

4. Properties Analysis

Before presenting the properties analysis under the proposed algorithm, we define the system total energy $H(q, p)$ as the sum of the total potential energy between α-agents, the total potential energy between α-agent and β-agent, the total potential energy between α-agent and γ-agent, and the kinetic energy between α-agent and γ-agent. It can be expressed as



(a)

(b)

Figure 4: The process of determining the expected position. (a) A2 scenario. (b) A1 and A3 scenarios.

Table 2: The expected position vector under four scenarios.

Scenario	Motion mode	Expected position vector $\widehat{q}_i$
A1, A2, and A3	Counterclockwise	$\widehat{q}_i = q_k, \forall \theta_{i,k}(m) - \theta_{i,j}(m+1) \geq \Delta\theta$ and $m = 1 \longrightarrow M_i$
A1, A2, and A3	Clockwise	$\widehat{q}_i = q_k, \forall \theta_{i,k}(m) - \theta_{i,j}(m-1) \geq \Delta\theta$ and $m = M_i \longrightarrow 1$
A4	—	$\widehat{q}_i = q_\gamma$

$$H(q,p) = \frac{1}{2}\sum_{i=1}^{N}\left[c_1^\alpha V_i^\alpha(q) + 2c_1^\beta V_i^\beta(q) + c_1^\gamma \left(q_i - q_\gamma\right)^T \left(q_i - q_\gamma\right) + \left(p_i - p_\gamma\right)^T \left(p_i - p_\gamma\right)\right], \tag{20}$$

where

$$V_i^\alpha(q) = \sum_{j\in N_i^\alpha} \eta_1^\alpha\left(d_{ij}\right)\psi_\alpha\left(\left\|q_j - q_i\right\|_\sigma\right), \tag{21}$$

$$V_i^\beta(q) = \sum_{k\in N_i^\beta} \eta_1^\beta\left(d_{ik}\right)\psi_\beta\left(\left\|q_k - q_i\right\|_\sigma\right). \tag{22}$$

We also define the system total energy at $t = 0$ as H_0.

Theorem 1. *Consider a system with N α-agents applying the control input (19) with multiagents motion model (2). Suppose that the β-agent and γ-agent are the static agent, the weight factor τ_s^α and τ_s^β are big enough, and H_0 is finite, then the following statements hold:*

(i) The distance between α-agent and γ-agent is not larger than $\sqrt{2H_0/c_1^\gamma}$ for all $t \geq 0$;

(ii) There is no collision between α-agents and between α-agent and β-agent at any time.

Proof. We first prove the part (i) of Theorem 1. By differentiating $H(q,p)$ in formula (20), we have

$$\dot{H}(q,p) = \frac{1}{2}\sum_{i=1}^{N} c_1^\alpha \dot{V}_i^\alpha(q) + \sum_{i=1}^{N} c_1^\beta \dot{V}_i^\beta(q) + \sum_{i=1}^{N} c_1^\gamma {p_i}^T\left(q_i - q_\gamma\right) + \sum_{i=1}^{N} {p_i}^T u_i. \tag{23}$$

Since β-agent and γ-agent are the static agent, then we have $\dot{q}_k = p_k = 0$ and $\dot{q}_\gamma = p_\gamma = 0$. In addition, since $\nabla_{q_j - q_i} V_i^\alpha(q) = \nabla_{q_j} V_i^\alpha(q) = -\nabla_{q_i} V_i^\alpha(q)$ and $\nabla_{q_k - q_i} V_i^\beta(q) = \nabla_{q_k} V_i^\beta(q) = -\nabla_{q_i} V_i^\beta(q)$, formula (23) can be rewritten as

$$\dot{H}(q,p) = c_1^\alpha \sum_{i=1}^{N} {p_i}^T \nabla_{q_i} V_i^\alpha(q) + c_1^\beta \sum_{i=1}^{N} {p_i}^T \nabla_{q_i} V_i^\beta(q) + \sum_{i=1}^{N} c_1^\gamma {p_i}^T\left(q_i - q_\gamma\right) + \sum_{i=1}^{N} {p_i}^T u_i. \tag{24}$$

Since $\widehat{p}_{i,k}$ for all β-agents within the sensing radius of α-agent i are equal, then let $\widetilde{p} = col\left(p_1, \cdots, p_N, \widehat{p}_{1,k}, \cdots \widehat{p}_{N,k}\right)$. Formulating (19), (21), and (22) into (24), we have

Table 3: Simulation parameters.

Parameter	Value
N	30
r	6
d	4
ε	0.5
h	0.2
a	2
b	20
c_1^α	1
c_2^α	1
c_1^β	1
c_2^β	1
c_1^γ	1
c_2^γ	1
τ_a^α	0.1
τ_e^α	10
τ_s^α	10000
$\Delta\theta$	$\pi/4$
$p_i{}^{\max}$	5 m/s
$d_{\min}$	1 m
Δt	0.1 s

$$\begin{aligned}\dot{H}(q,p) &= -c_2^\alpha \sum_{i=1}^{N} \sum_{j\in N_i^\alpha} a_{ij}(q) p_i{}^T\left(p_i - p_j\right) - c_2^\beta \sum_{i=1}^{N} \sum_{k\in N_i^\beta} b_{i,k}(q) p_i{}^T\left(p_i - \widehat{p}_{i,k}\right) - c_2^\gamma \sum_{i=1}^{N} p_i{}^T p_i \\ &= -\widetilde{p}^T (L\otimes I_m)\widetilde{p},\end{aligned} \tag{25}$$

where $L = \begin{pmatrix} c_2^\alpha(\Delta(A)-A) + c_2^\beta\Delta(B) + c_2^\gamma E_{N\times N} & 0_{N\times N} \\ -c_2^\beta B & 0_{N\times N} \end{pmatrix}$, $\Delta(A)$ is a diagonal matrix with diagonal elements $\sum_{j\in N_i^\alpha} a_{ij}(q)$, A is an adjacency matrix, $A = (a_{ij}(q))$, $\Delta(B)$ is a diagonal matrix with diagonal elements $\sum_{k\in N_i^\beta} b_{i,k}(q)$, B is an adjacency matrix, $B = (b_{i,k}(q))$, $E_{N\times N}$ is a N-order identity matrix, and $\otimes$ is the Kronecker product.

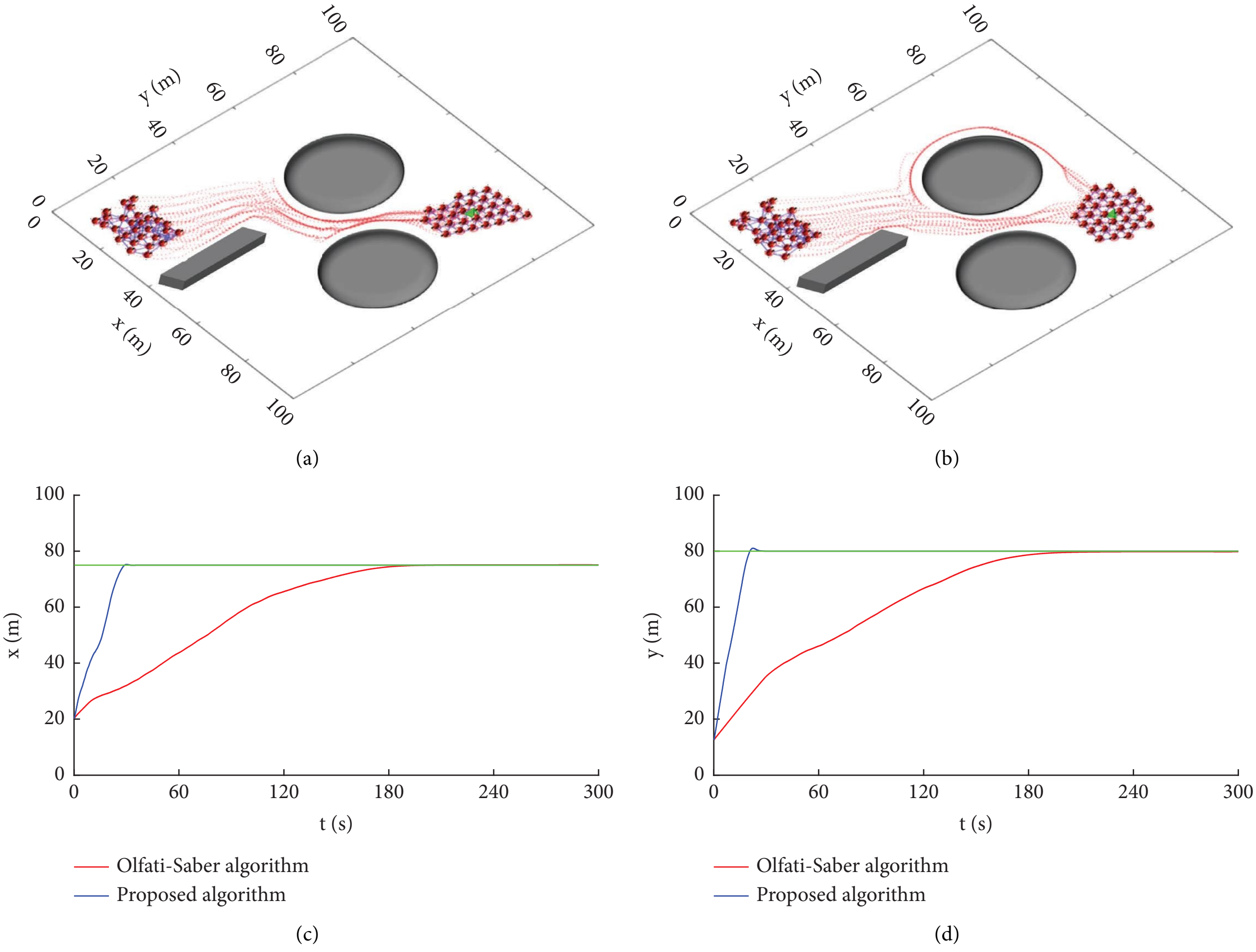
Since c_2^α, c_2^β, and c_2^γ are the positive numbers, and $\Delta(A) - A$ and $\Delta(B)$ are the positive semidefinite matrix, then L is a positive semidefinite matrix. Consequently, we have $-\widetilde{p}^T(L\otimes I_m)\widetilde{p} \le 0$, that is $\dot{H}(q,p) \le 0$, which implies that the system total energy $H(q,p)$ is a nonincreasing function, and thus $H(q,p) \le H_0$ for all $t \ge 0$. From formula (20), we have $c_1^\gamma (q_i - q_\gamma)^T (q_i - q_\gamma) \le 2H_0$ for any α-agent i. Hence, the distance between α-agent and γ-agent is not greater than $\sqrt{2H_0/c_1^\gamma}$ for all $t \ge 0$.

Next, we prove the part (ii) of Theorem 1 by contradiction. Suppose that α-agent i collides with α-agent j or obstacle boundary point k at t^*. Since the weight factor τ_s^α and τ_s^β are big enough, then $V_i^\alpha(q)$ or $V_i^\beta(q)$ tends to infinity. In addition, since H_0 is finite, we have $H(q,p) > H_0$. This contradicts the proof that $H(q,p)$ is a nonincreasing function. Consequently, there is no collision at any time.

5. Simulations

In this section, we first illustrate the effectiveness of the proposed algorithm by comparing it with the Olfati-Saber flocking algorithm with obstacle avoidance. In this example, the simulation is performed on 30 α-agents and a γ-agent moving in a two-dimensional Euclidean space with two spherical obstacles and a wall obstacle. The initial position vectors of 30 α-agents are selected randomly from $[10,30] \times [5,20]$, the initial velocity vectors of 30 α-agents are selected randomly from $[0,2] \times [0,2]$, the initial position vector of γ-agent is $(75,80)^T$, and the initial velocity vector of γ-agent is zero. The parameters of the proposed algorithm and the Olfati-Saber algorithm are shown in Table 3. The experiments were run continuously for 300 s under the three scenarios of spherical and wall obstacle independence, spherical obstacle combination, and spherical and wall obstacle combination, and the results are shown in Figures 5 and 6.

In Figures 5(a) and 5(b), the black cylinder is the spherical obstacle, the black cuboid is the wall obstacle, the green triangle is the γ-agent, the red *Coccinella septempunctata* is the α-agent, the blue solid line between red circles is the neighborhood relationship between α-agents, and the red dotted lines are the motion trajectories of 30 α-agents. It can be seen from these figures that the two algorithms can enable all α-agents to bypass independent



(a)

(b)

(c)

(d)

Figure 5: Motion process of the two algorithms under the independence scenario of spherical and wall obstacles. (a) Olfati-Saber algorithm. (b) Proposed algorithm. (c) x trajectories. (d) y trajectories.

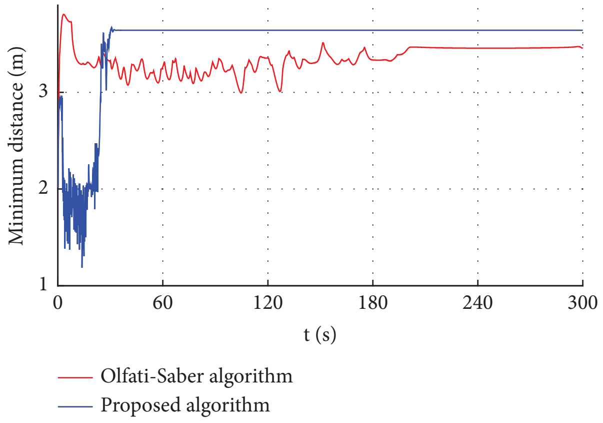


Figure 6: Minimum distance between α-agent and objects.

spherical or wall obstacles on their way. In Figures 5(c) and 5(d), the red, blue, and green solid lines represent the center positions $(1/N\sum_{i=1}^{N} q_i)$ of 30 α-agents in the Olfati-Saber flocking algorithm with obstacle avoidance, the center positions of 30 α-agents in the proposed algorithm, and the position of γ-agent, respectively. It can be observed that the convergence time of the proposed algorithm is decreased by more than 80% compared with the Olfati-Saber flocking algorithm with obstacle avoidance, which indicates that the proposed algorithm can make α-agent bypass obstacles faster. Figure 6 shows the minimum distance between α -agent and objects (e.g., other α-agents or obstacles), and it can be observed that the minimum distances of the two algorithms are greater than the minimum safe distance $d_{\min}$. This indicates that there is no α-agent that collided with other α-agents or obstacles, and it also demonstrates that the proposed algorithm can ensure that α-agent bypass obstacles without collision.

In Figures 7 and 8, the black circle is the spherical obstacle, the black solid line is the wall obstacle, the green triangle is the γ-agent, and the red circle is the α-agent. Figures 7 and 8 show that many α-agents stop near the combination position of two obstacles, and it is difficult to bypass these obstacles in the Olfati-Saber flocking algorithm with obstacle avoidance. A possible reason for this is that the projection velocity here does not point in the direction of bypassing these obstacles. However, the proposed algorithm can enable all α-agents to bypass these obstacles, as well as enable all α-agents to regroup. This demonstrates that the

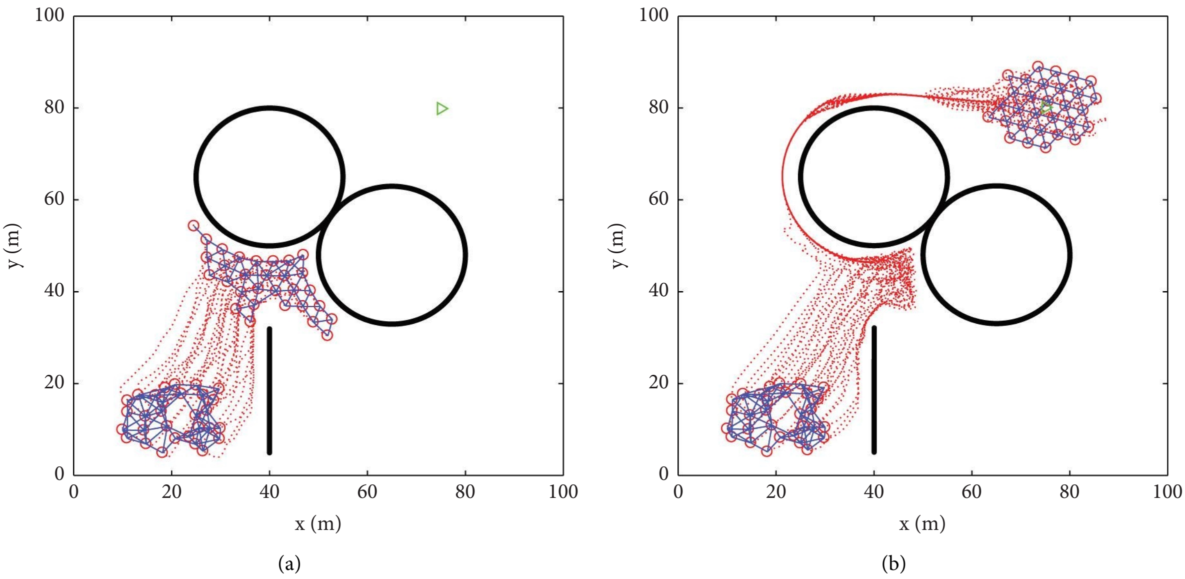


Figure 7: Motion process of the two algorithms under the combination scenario of two spherical obstacles. (a) Olfati-Saber algorithm. (b) Proposed algorithm.

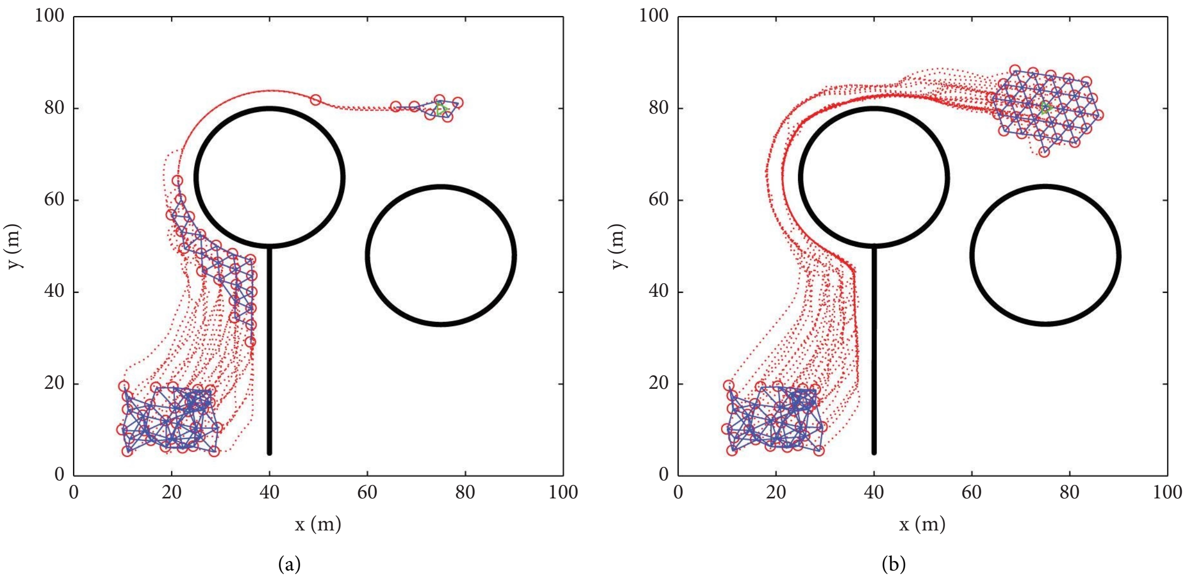


Figure 8: Motion process of the two algorithms under the combination scenario of spherical and wall obstacles. (a) Olfati-Saber algorithm. (b) Proposed algorithm.

proposed algorithm can relax the constraints of obstacle shape and boundary in the Olfati-Saber flocking algorithm with obstacle avoidance and has better environmental adaptability.

To further demonstrate the environmental adaptability of the proposed algorithm, we selected 30–60 α-agents and a γ-agent to move in a two-dimensional Euclidean space with multiple obstacles such as a diamond obstacle, a spherical obstacle, and a C-shaped obstacle. The algorithm parameters are shown in Table 3. The initial position vectors of 30–60 α-agents are selected randomly from $[2, 30] \times [5, 30]$, the initial velocity vectors of 30–60 α-agents are selected randomly from $[0, 2] \times [0, 2]$, the initial position vector of γ-agent is $(75, 25)^T$, and the initial velocity vector of γ-agent is zero. The experiments were run continuously for 300 s.

Figure 9 shows the motion trajectories of 30 and 60 α-agents under multiple obstacles. It can be seen that all α-agents can move along the boundary of obstacles, bypass obstacles, and finally converge on the γ-agent at the center position of all α-agents. In Figure 10, we show the minimum distance between α-agent and objects (e.g., other α-agents or obstacles) for randomly generated 30, 40, 50, and 60 α-agents following the γ-agent 100 times under multiple obstacles. It can be observed

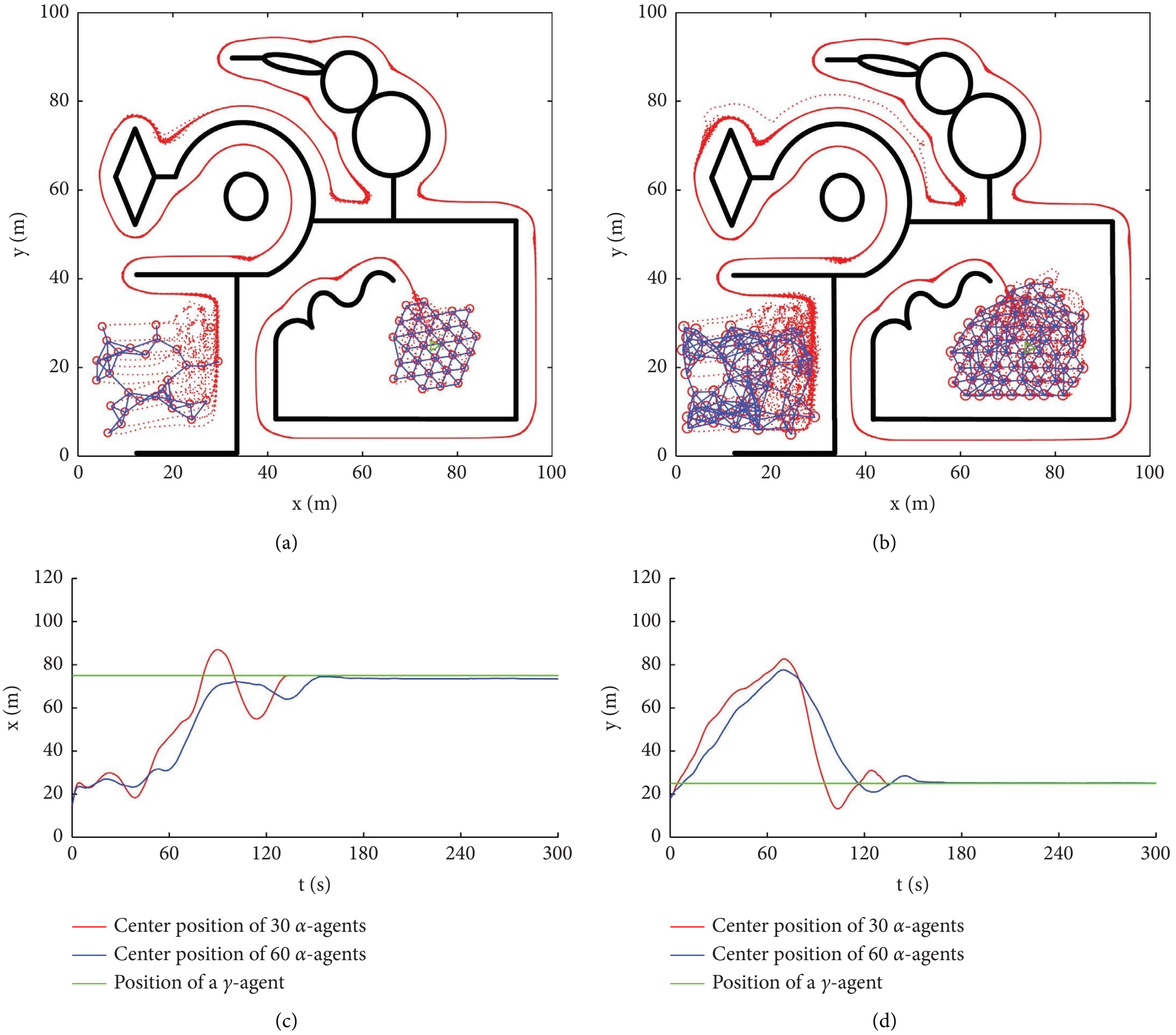


Figure 9: Motion process of 30 and 60 α-agents under multiple obstacles. (a) 30 α-agents. (b) 60 α-agents. (c) x trajectories. (d) y trajectories.

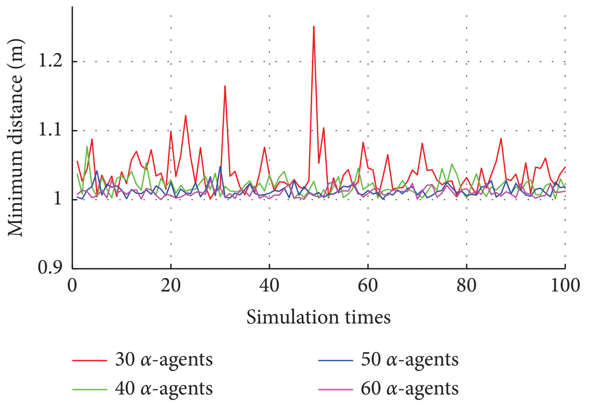


Figure 10: Minimum distance between α-agent and objects in 100 simulation times.

from the figure that the minimum distance is always greater than the minimum safe distance $d_{\min}$, which illustrates that the proposed algorithm can reduce the influence of the number of α -agents on the minimum distance and can also find a reasonable route to bypass obstacles without collision and follow the γ -agent in the complex environment.

6. Conclusions

In this paper, we regarded each obstacle boundary point within the sensing radius of α-agent as a virtual β-agent and designed an obstacle boundary point and an expected velocity-based flocking algorithm for multiagents with obstacle avoidance. Under the proposed algorithm, we proved that the collision-free motion of multiagents can be realized when the sufficient conditions in Theorem 1 hold. Three examples under the independence scenario of spherical and wall obstacles, the combination scenario of two spherical obstacles, and the combination scenario of spherical and wall obstacles are presented to show that the proposed algorithm can relax the constraints of obstacle shape and boundary in Olfati-Saber's flocking algorithm with obstacle avoidance. Besides, an example under multiple obstacles is given to illustrate that the proposed algorithm with better environmental adaptability can make α-agent quickly bypass obstacles without collision and follow the γ-agent. Future works will consider the influence of time delay and external disturbance in the proposed algorithm.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Authors' Contributions

Jianhui Wu was in charge of conceptualization, methodology, investigation, and writing the original draft. Yuanfa Ji handled conceptualization, supervision, writing the review, and editing. Xiyan Sun was in charge of formal analysis, writing the review, and editing. Weibin Liang handled investigation and validation.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (grant nos. 61861008, 62161007, and 62061010) and the Foundation from the Guangxi Zhuang Autonomous Region (grant nos. AA19182007, AA19254029, AA20302022, and AB21196041).

References

[1] H. Qiu and H. Duan, "Pigeon interaction mode switch-based UAV distributed flocking controlunder obstacle environments," *ISA Transactions*, vol. 71, pp. 93–102, 2017.

[2] H. Wang, J. Wu, K. Fang, L. Cai, L. S. Wang, and Z. D. Dai, "Application of robo-pigeon in ethological studies of bird flocks," *Journal of Integrative Neuroscience*, vol. 19, no. 3, pp. 443–448, 2020.

[3] Y. Jia and L. Wang, "Leader-follower flocking of multiple robotic fish," *IEEE*, vol. 20, no. 3, pp. 1372–1383, 2015.

[4] G. M. Ginnaw, I. K. Davidson, H. R. Harding et al., "Effects of multiple stressors on fish shoal collective motion are independent and vary with shoaling metric," *Animal Behaviour*, vol. 168, pp. 7–17, 2020.

[5] M. B. Pfeiffer, R. B. Iglay, T. W. Seamans, B. F. Blackwell, and T. L. DeVault, "Deciphering interactions between white-tailed deer and approaching vehicles," *Transportation Research Part D: Transport and Environment*, vol. 79, pp. 102251–102259, 2020.

[6] G. Ariel, M. Sidortsov, S. D. Ryan, S. Heidenreich, M. Bär, and A. Be'er, "Collective dynamics of two-dimensional swimming bacteria: experiments and models," *Physical Review A*, vol. 98, no. 3, pp. 032415–32510, 2018.

[7] M. Tian, C. Zhang, R. Zhang, and J. Yuan, "Collective motion enhances chemotaxis in a two-dimensional bacterial swarm," *Biophysical Journal*, vol. 120, no. 9, pp. 1615–1624, 2021.

[8] B. S. Park and S. J. Yoo, "Connectivity-maintaining and collision-avoiding performance function approach for robust leader-follower formation control of multiple uncertain underactuated surface vessels," *Automatica*, vol. 127, Article ID 109501, 109510 pages, 2021.

[9] S. Yang, W. Bai, T. Li et al., "Neural-network-based formation control with collision, obstacle avoidance and connectivity maintenance for a class of second-order nonlinear multi-agent systems," *Neurocomputing*, vol. 439, pp. 243–255, 2021.

[10] Y. Yang, Y. Y. Chen, and H. Y. Yang, "Robust flocking of multiple intelligent agents with multiple disturbances," *International Journal of Intelligent Systems*, vol. 37, no. 10, pp. 7571–7583, 2022.

[11] B. Boardman, T. Harden, and S. Martínez, "Multi-agent motion planning with sporadic communications for collision avoidance," *IFAC Journal of Systems and Control*, vol. 15, Article ID 100126, 100212 pages, 2021.

[12] D. Bhattacharjee, A. Chakravarthy, and K. Subbarao, "Nonlinear model predictive control and collision-cone-based missile guidance algorithm," *Journal of Guidance, Control, and Dynamics*, vol. 44, no. 8, pp. 1481–1497, 2021.

[13] J. A. Douthwaite, S. Zhao, and L. S. Mihaylova, "Velocity obstacle approaches for multi-agent collision avoidance," *Unmanned Systems*, vol. 07, no. 01, pp. 55–64, 2019.

[14] A. Chakravarthy and D. Ghose, "Collision cone-based net capture of a swarm of unmanned aerial vehicles," *Journal of Guidance, Control, and Dynamics*, vol. 43, no. 9, pp. 1688–1710, 2020.

[15] X. Xu, W. Pan, Y. Huang, and W. Zhang, "Dynamic collision avoidance algorithm for unmanned surface vehicles via layered artificial potential field with collision cone," *Journal of Navigation*, vol. 73, no. 6, pp. 1306–1325, 2020.

[16] P. Chen, Y. Huang, E. Papadimitriou, J. Mou, and P. van Gelder, "Global path planning for autonomous ship: a hybrid approach of fast marching square and velocity obstacles methods," *Ocean Engineering*, vol. 214, Article ID 107793, 107813 pages, 2020.

[17] J. López, P. Sanchez-Vilariño, M. D. Cacho, and E. L. Guillén, "Obstacle avoidance in dynamic environments based on velocity space optimization," *Robotics and Autonomous Systems*, vol. 131, Article ID 103569, 103621 pages, 2020.

[18] M. Fuad, T. Agustinah, and D. Purwanto, "Modified headed social force model based on hybrid velocity obstacles for mobile robot to avoid disturbed groups of pedestrians," *International Journal of Intelligent Engineering and Systems*, vol. 14, no. 3, pp. 222–241, 2021.

[19] S. D. He, C. Dong, S. L. Dai, and T. Zou, "Cooperative deterministic learning and formation control for underactuated USVs with prescribed performance," *International Journal of Robust and Nonlinear Control*, vol. 32, no. 5, pp. 2902–2924, 2022.

[20] S. D. He, M. Wang, S. L. Dai, and F. Luo, "Leader-follower formation control of USVs with prescribed performance and collision avoidance," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 1, pp. 572–581, 2019.

[21] D. Sakai, H. Fukushima, and F. Matsuno, "Flocking for multirobots without distinguishing robots and obstacles," *IEEE Transactions on Control Systems Technology*, vol. 25, no. 3, pp. 1019–1027, 2017.

[22] S. Nath, M. Baishya, and D. Ghose, "Decentralised coverage of a large structure using flocking of autonomous agents having a dynamic hierarchy model," *Autonomous Robots*, vol. 46, no. 5, pp. 617–643, 2022.

[23] A. D. Dang, H. M. La, T. Nguyen, and J. Horn, "Formation control for autonomous robots with collision and obstacle avoidance using a rotational and repulsive force-based approach," *International Journal of Advanced Robotic Systems*, vol. 16, no. 3, pp. 172988141984789–16, 2019.

[24] Q. Wang, J. Chen, H. Fang, and Q. Ma, "Flocking control for multi-agent systems with stream-based obstacle avoidance," *Transactions of the Institute of Measurement and Control*, vol. 36, no. 3, pp. 391–398, 2014.

[25] R. Olfati-Saber, "Flocking for multi-agent dynamic systems: algorithms and theory," *IEEE Transactions on Automatic Control*, vol. 51, no. 3, pp. 401–420, 2006.

[26] J. J. Li, W. Zhang, H. S. Su, and Y. P. Yang, "Flocking of partially-informed multi-agent systems avoiding obstacles with arbitrary shape," *Autonomous Agents and Multi-Agent Systems*, vol. 29, no. 5, pp. 943–972, 2015.

[27] R. Marino, F. Mastrogiovanni, A. Sgorbissa, and R. Zaccaria, "A minimalistic quadrotor navigation strategy for indoor multi-floor scenarios," *Intelligent Autonomous Systems 13*, vol. 302, pp. 1561–1570, 2016.

[28] R. B. Grando, J. C. de Jesus, V. A. Kich, A. H. Kolling, and P. L. J. Drews-Jr, "Double critic deep reinforcement learning for mapless 3D navigation of unmanned aerial vehicles," *Journal of Intelligent and Robotic Systems*, vol. 104, no. 2, pp. 29–14, 2022.

[29] K. Taylor and S. M. LaValle, "Intensity-based navigation with global guarantees," *Autonomous Robots*, vol. 36, no. 4, pp. 349–364, 2014.

[30] S. Lee, T. M. Adams, and B. Y. Ryoo, "A fuzzy navigation system for mobile construction robots," *Automation in Construction*, vol. 6, no. 2, pp. 97–107, 1997.

[31] Q. l. Xu, "Randombug: novel path planning algorithm in unknown environment," *The Open Electrical and Electronic Engineering Journal*, vol. 8, no. 1, pp. 252–257, 2014.

[32] W. Mao, B. Sun, G. Xu, C. Liu, C. Si, and W. Wang, "Understanding effects of collaborations in developing mobile computing systems: popularity, efficiency, and quality," *IEEE Access*, vol. 7, pp. 33380–33392, 2019.

[33] K. N. McGuire, G. C. H. E. de Croon, and K. Tuyls, "A comparative study of bug algorithms for robot navigation," *Robotics and Autonomous Systems*, vol. 121, Article ID 103261, 17 pages, 2019.