Flocking for multi-robots without distinguishing robots and obstacles

Daito Sakai, Hiroaki Fukushima, *Member, IEEE*, and Fumitoshi Matsuno, *Member, IEEE*

Abstract—**Most existing studies of multiple mobile robots assume that robots can distinguish between other robots and obstacles. However, if a flocking algorithm were available that did not require this ability, easier implementation could be expected. In this paper, we propose a flocking algorithm that does not distinguish between a robot and an obstacle. In other words, all detected objects are regarded as obstacles in the proposed algorithm. Thus, velocity information on neighboring robots is not required. We also show that the proposed algorithm maintains the desired properties of existing flocking algorithms, even though only limited information is used in the proposed algorithm. Furthermore, unlike many previous studies, the effectiveness of the algorithm is demonstrated not only by simulations, but also in real robot experiments. In the experiments, neighboring robots and obstacles are detected using only laser range finders, since the robots do not need to distinguish between another robot and an obstacle.**

Index Terms—**multi-robot system, flocking, cohesion, velocity consensus, obstacle avoidance.**

I. INTRODUCTION

For decades, flocking behavior observed in nature has inspired researchers in many different fields (see e.g. [1]–[4]). In the field of control theory, feedback control algorithms to achieve flocking behavior have been proposed, and the stability of the closed-loop has been analyzed [5]–[15]. Related topics such as consensus [16]–[19] and formation control of multi-agent systems [20]–[26] have been studied intensively. This paper focuses on flocking algorithms based on the classical model by Reynolds [1], which consists of three heuristic rules: 1) Separation: steer to avoid crowding local flockmates; 2) Alignment: steer towards the average heading of local flockmates; 3) Cohesion: steer to move toward the average position of local flockmates.

Tanner et al. [9]–[11] proposed a decentralized control law consisting of a velocity feedback term for alignment and a position feedback term based on an artificial potential field to achieve separation and cohesion. Olfati-Saber[12] suggested that an additional feedback term to track a virtual leader was necessary to avoid fragmentation for a generic set of initial states. A flocking algorithm capable of performing obstacle avoidance was also presented by creating a virtual agent on the boundary of each nearby obstacle. Su et al. [13] considered the case where only a small fraction of agents have information about the virtual leader, and provided a stability analysis of the main algorithm in [12] for a free space. Furthermore, Zavlanos et al. [14] proposed a flocking algorithm that guaranteed the connectivity of a proximity-based graph. In [15], a connectivity-preserving flocking algorithm without velocity measurements was proposed.

One of the possible applications of flocking algorithms is cooperative control of mobile robots. Although the flocking algorithms mentioned above assume the ability to distinguish between a neighbor

and an obstacle, this is not as easy for robots as it is for animals such as birds. A reason is that laser range finders, which are commonly used on robots, cannot distinguish obstacles and other robots, since they only return range data. Furthermore, it is often difficult to extract the velocity information of each neighbor from sensor data, which is required in the existing flocking algorithms.

In this paper, we propose a flocking algorithm that does not distinguish between a robot and an obstacle. In other words, all detected objects are regarded as obstacles in the proposed algorithm. Thus, velocity information on neighboring robots is not required. We also give theoretical results on the stability of the closed-loop system, which is divided into two parts. The first part considers the case where no obstacle exists in the neighborhood of each robot. Although the proposed method is developed to apply in the presence of obstacles, the stability analysis in the case of no obstacle is still meaningful, since velocity matching of robots is typically achieved only after the robots avoid all obstacles. The second part gives analysis in the presence of obstacles in the neighborhoods of the robots. Unfortunately, our analysis suggests that the velocity matching is difficult to be guaranteed in this case. Instead, we derive conditions for the closed-loop system to be dissipative, and provide results on group cohesion and collision avoidance when the closed-loop system is dissipative. Furthermore, unlike many previous studies, we demonstrate the effectiveness of the algorithm is demonstrated not only with simulations but also in real robot experiments. In the experiments, neighboring robots and obstacles are detected using only laser range finders, since robots do not need to distinguish between other robots and obstacles.

II. PROBLEM SETTING

Consider n robots, which we call α agents. Let $q_i(t) \in \mathbb{R}^m$ denote the position at time t of a point that is fixed on robot i ($i = 1, \ldots, n$). The movement of robot i is described as

$$\dot{q}_i = p_i, \quad \dot{p}_i = u_i \tag{1}$$

where $u_i \in \mathbb{R}^m$ and $p_i \in \mathbb{R}^m$ denote the control input and velocity, respectively, of α agent i. We assume that there are ℓ static obstacles in the environment. Thus, when an α agent and an obstacle are not distinguished, we refer to them as object j ($j = 1, \ldots, n + \ell$). Without loss of generality, we assume that objects $1, \ldots, n$ are α agents. The shape of objects is assumed to be convex.

We also assume that α agents are able to detect an object within a given maximum range, r. The set of neighbors of robot i is defined as the following set of detected objects

$$N_i := \{j \in \{1, \ldots, n + \ell\} \backslash \{i\} : \|\hat{q}_{ij} - q_i\| < r\}, \tag{2}$$

where $\hat{q}_{ij}$ is the closest point on the object j to q_i. We also define the following subsets of N_i

$$N_i^\alpha := \{j \in \{1, \ldots, n\} \backslash \{i\} : \|\hat{q}_{ij} - q_i\| < r\} \tag{3}$$

$$N_i^\beta := \{j \in \{n + 1, \ldots, n + \ell\} : \|\hat{q}_{ij} - q_i\| < r\} \tag{4}$$

which are the sets of the α agents and the obstacles, respectively, in N_i. Furthermore, we define

$$\mathcal{S}_i^\alpha := \{1, \ldots, n\} \backslash \{i\}, \quad \mathcal{S}^\beta := \{n+1, \ldots, n+\ell\}$$
$$\mathcal{S}_i := \mathcal{S}_i^\alpha \cup \mathcal{S}^\beta = \{1, \ldots, n+\ell\} \backslash \{i\}, \quad i = 1, \ldots, n$$

for later use.

As in previous studies [12], [13], a virtual leader, called a γ agent, is introduced to represent the target position and velocity of the group. More precisely, the γ agent is a point described as

$$\dot{q}_\gamma = p_\gamma, \quad \dot{p}_\gamma = u_\gamma \tag{5}$$

where $q_\gamma \in \mathbb{R}^m$ and $p_\gamma \in \mathbb{R}^m$ are the position and velocity of the γ agent. The control input $u_\gamma \in \mathbb{R}^m$ is decided depending on the target trajectory of the group.

In this paper, we propose a control algorithm for α agents to follow the γ agent without distinguishing between obstacles and other α agents in the neighborhood. The algorithm must be constructed without losing the desirable properties guaranteed with the previous algorithms [12], [13].

III. OVERVIEW OF PREVIOUS FLOCKING ALGORITHMS IN OBSTACLE ENVIRONMENTS

Our flocking algorithm is constructed by modifying the control law in [12], which considers obstacle environments. In this section, we give a brief overview of the algorithm in [12], for the sake of clarity.

In previous studies, including [12], an α agent and an obstacle in the neighborhood are distinguished in the flocking algorithm. Thus, N_i^α in (3) and N_i^β in (4) are used in the algorithm instead of N_i in (2). Since an α agent is assumed to be a particle, q_j $(j = 1, \ldots, n)$ is equal to the position of the particle α agent j. Thus, $\hat{q}_{ij}$ in (3), which is the closest point to q_i on α agent j, is equal to q_j.

The control input u_i is composed of three terms as follows.

$$u_i = u_i^\alpha + u_i^\beta + u_i^\gamma. \tag{6}$$

The term u_i^α in (6) can be divided into two components as

$$u_i^\alpha = f_i^g + f_i^d.$$

The first component f_i^g is defined as

$$f_i^g := -c_1^\alpha \sum_{j \in N_i^\alpha} \nabla_{q_i} \psi_\alpha(\|q_j - q_i\|_\sigma) \tag{7}$$

where $c_1^\alpha > 0$ is a feedback gain and ψ_α is an artificial potential function for collision avoidance with neighbors. Furthermore, $\|z\|_\sigma$ is defined as

$$\|z\|_\sigma = \frac{1}{\epsilon_1}\left[\sqrt{\epsilon_2^2 + \epsilon_1 \|z\|^2} - \epsilon_2\right] \tag{8}$$

for positive constants ϵ_1 and ϵ_2. ($\epsilon_1 = 0.1$ and $\epsilon_2 = 1$ are used in [12].) The reason why $\|q_j - q_i\|_\sigma$ is used in (7) instead of the Euclidean norm is that $\|q_j - q_i\|_\sigma$ is differentiable everywhere while $\|q_j - q_i\|$ is not for $q_i = q_j$. Since the desired interagent distance is given as d_α, the artificial potential function $\psi_\alpha(z)$ in (7) has the minimum value at $z = \|d_\alpha\|_\sigma$, where $\|d_\alpha\|_\sigma = \left(\sqrt{\epsilon_2^2 + \epsilon_1 d_\alpha^2} - \epsilon_2\right)/\epsilon_1$. For collision avoidance, the value of $\psi_\alpha(z)$ monotonically increases as z goes from $z = \|d_\alpha\|_\sigma$ to $z = 0$. In order to avoid the discontinuity of f_i^g at $\|q_j - q_i\| = r$, the potential function ψ_α is decided such that the gradient of $\psi_\alpha(z)$ vanishes for $z \geq \|r\|_\sigma$. For further details on how to construct $\psi_\alpha(z)$, see [12].

The second component f_i^d is introduced for velocity matching with other α agents as

$$f_i^d := c_2^\alpha \sum_{k \in N_i^\alpha} a_{ij}(q)(p_j - p_i), \quad q := \text{col}\{q_1, \cdots, q_n\} \tag{9}$$
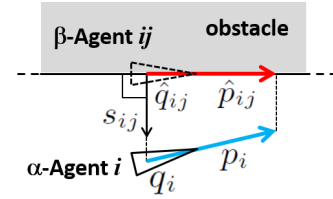

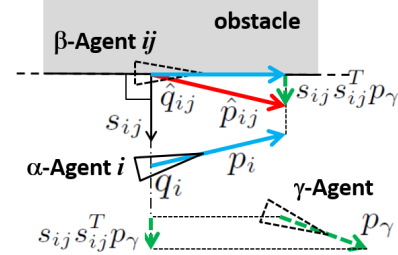Fig. 1. Position and velocity of β-agent (existing method).

Fig. 2. Position and velocity of β-agent (proposed method).

where $c_2^\alpha > 0$ is a feedback gain and $a_{ij}(q)$ is defined as

$$a_{ij}(q) = \rho_h(\|\hat{q}_{ij} - q_i\|_\sigma / r_\sigma) \tag{10}$$

by using the bump function

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

for $h \in (0, 1)$. Since $a_{ij}(q)$ is a positive definite function that decreases as $\|q_j - q_i\|_\sigma$ grows, a larger control input is generated to match the velocity with a closer agent.

We next define u_i^β in (6). When an α agent detects an obstacle, it generates a virtual agent called a β agent at the closest point on the obstacle. We refer to the β agent generated on obstacle $j \in N_i^\beta$ as β agent ij. In the case where an obstacle is an infinite wall (see Fig 1 for an example of $m = 2$), the position of the β agent ij is $\hat{q}_{ij}$, and the velocity is

$$\hat{p}_{ij} = p_i - s_{ij} s_{ij}^T p_i \tag{12}$$

where s_{ij} is a unit vector from $\hat{q}_{ij}$ to α agent i. As illustrated in Fig. 1, $\hat{p}_{ij}$ in (12) is the projection of p_i onto the wall. By using $\hat{q}_{ij}$ and $\hat{p}_{ij}$, the term u_i^β in (6) is defined as

$$u_i^\beta = -c_1^\beta \sum_{j \in N_i^\beta} \nabla_{q_i} \psi_\beta(\|\hat{q}_{ij} - q_i\|_\sigma)$$
$$+ c_2^\beta \sum_{j \in N_i^\beta} a_{ij}(q)(\hat{p}_{ij} - p_i) \tag{13}$$

where ψ_β is a repulsive potential function for obstacle avoidance. For a given positive constant $d_\beta(< r)$, the value of $\psi_\beta(z)$ monotonically increases as z goes from $z = \|d_\beta\|_\sigma$ to $z = 0$. The gradient of $\psi(z)$ vanishes at $z = \|d_\beta\|_\sigma$ and remains zero for $z > \|d_\beta\|_\sigma$.

The term u_i^γ is introduced to track the γ agent. In [12], the following u_i^γ is used for flocking with obstacle avoidance.

$$u_i^\gamma = -c_1^\gamma \mu(q_i - q_\gamma) - c_2^\gamma(p_i - p_\gamma) \tag{14}$$

where $\mu(z) := \frac{z}{\sqrt{1+\|z\|^2}}$, and c_1^γ, $c_2^\gamma > 0$ are feedback gains.

IV. PROPOSED METHOD

A distinguishing feature of the proposed method as compared to the standard method in Section III is that an α agent generates a β agent on each object in the neighborhood, regardless of whether the object is an α agent or an obstacle. While $\hat{q}_{ij}$ is generated in the same way as in Section III, the velocity $\hat{p}_{ij}$ of β agent ij is

$$\hat{p}_{ij} = p_i - s_{ij}s_{ij}^T p_i + s_{ij}s_{ij}^T p_\gamma \tag{15}$$

where

$$s_{ij} := \begin{cases} \frac{q_i - \hat{q}_{ij}}{\|q_i - \hat{q}_{ij}\|}, & \text{if } \|q_i - \hat{q}_{ij}\| \neq 0 \\ 0, & \text{otherwise.} \end{cases} \tag{16}$$

The difference from velocity $\hat{p}_{ij}$ in (12) is that the third term is added on the right hand side of (15). As illustrated in Fig. 2, the third term $s_{ij}s_{ij}^T p_\gamma$ in (15) is the projection of p_γ onto the line connecting q_i and $\hat{q}_{ij}$, as shown by the dash-dotted line in Fig. 2. By using $\hat{q}_{ij}$ and $\hat{p}_{ij}$, the proposed control law is described as

$$u_i = u_i^\beta + u_i^\gamma \tag{17}$$

$$u_i^\beta = -c_1^\beta \sum_{j \in N_i} \nabla_{q_i} \psi(\|\hat{q}_{ij} - q_i\|_\sigma)$$

$$+ c_2^\beta \sum_{j \in N_i} a_{ij}(q)(\hat{p}_{ij} - p_i) \tag{18}$$

where N_i^β in (13) is replaced by N_i, since an α agent and an obstacle are not distinguished. The definition of a_{ij} and u_i^γ are the same as in (10) and (14), respectively. The artificial potential function ψ is defined in the same way as ψ_β in [12], although a slightly simplified definition is used for ψ. More precisely, for a given positive constant $d(< r)$, $\psi(z)$ is defined as

$$\psi(z) = \int_{d_\sigma}^z \phi(s)ds \tag{19}$$

using $d_\sigma := \|d\|_\sigma$ and the following function $\phi(z)$

$$\phi(z) := \rho_h(z/d_\sigma) \, \phi_0(z) \tag{20}$$

$$\phi_0(z) := \frac{-1}{z + \epsilon_0}, \quad \epsilon_0 > 0 \tag{21}$$

where a small number ϵ_0 is added in the denominator of $\phi(z)$ to prevent $\phi(z)$ from diverging as $z \to 0$. Since $\phi(z) = 0$ for $z \geq d_\sigma$, the gradient of $\psi(z)$ vanishes for $z \geq d_\sigma$. It is seen from $\phi_0(z) < 0$ ($z \geq 0$) that $\psi(\|\hat{q}_{ij} - q_i\|_\sigma)$ monotonically decreases for $0 < \|\hat{q}_{ij} - q_i\|_\sigma < d_\sigma$. Fig. 3 illustrates $\psi(z)$ and $\rho(z/d_\sigma)$ for $d_\sigma \simeq 1.5$ and $\epsilon_0 = 0.001$.

One of the differences between the proposed algorithm and that described in Section III is that the proposed algorithm determines the control input by generating a β agent not only for an obstacle but also for an α agent in the neighborhood. Thus, information on the velocities of other α agents is not required to determine the control input. Another difference is that the velocity of a β agent in (12) is modified by adding the third term on the right hand side in (15). The purpose of this modification is to maintain the desirable properties of the algorithms in [12], [13], as described in the next section.

V. STABILITY ANALYSIS

As described in Section IV, the proposed algorithm does not use the velocity information of other α agents and generates a β agent for an α agent in the neighborhood, since all detected objects are regarded as obstacles. Thus, it is important to discuss if the proposed algorithm using such limited information guarantees stability results similar to those of the previous studies.

In [12], theoretical results on group cohesion, velocity matching, and collision avoidance were provided in the case of no obstacles
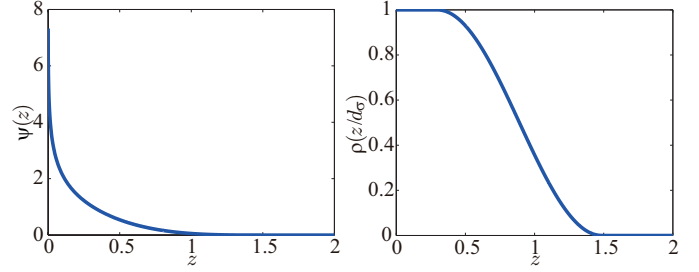

Fig. 3. $\psi(z)$ (left) and $\rho(z/d_\sigma)$ (right) for $d_\sigma \simeq 1.5$ and $\epsilon_0 = 0.001$.

and particle α agents. Those results were derived in the case where the velocity of the γ agent is time varying. However, a limitation is that the relationship between the asymptotic velocity of α agents and the γ agent's velocity is not clear. Su et al. [13] modified the results of [12] so that all α agents asymptotically had the same velocity as the γ agent's, by assuming the γ agent has a constant velocity. As in [13], we assume that the γ agent has a constant velocity and prove similar results on group cohesion, velocity matching, and collision avoidance.

Although the proposed method is developed to apply in obstacle environments, stability analysis in the case of no obstacles is still meaningful. A reason is that velocity matching of robots is typically achieved only after the robots avoid all obstacles, as seen in the simulation and the experiment in Sections VI-VII. Since the robots do not distinguish between other robots and obstacles, they do not recognize if all obstacles are avoided. As a result, the robots keep using the proposed method, even if there are no more obstacles around them. Furthermore, since the proposed algorithm is quite different from the previous ones, it is not straightforward to prove stability results similar to those in the previous studies. Theorem 1 below suggests that the properties such as velocity matching are not proved, if we adopt the velocity of β agents in (12), in the same way as in [12]. In other words, it is necessary to add the term $s_{ij}s_{ij}^T p_\gamma$ to (12), as described in (15), to obtain Theorem 1.

In the presence of obstacles, [12] discusses conditions for the system of α agents to be dissipative, while it is difficult to prove the properties such as velocity matching. Similarly, we derive conditions for the proposed algorithm to make the system dissipative, and give results on group cohesion and collision avoidance when the system is dissipative.

Before presenting the stability analysis, we define the Hamiltonian as

$$H(q, p) = \frac{1}{2} \sum_{i=1}^n \left[U_i(q) + (p_i - p_\gamma)^T (p_i - p_\gamma) \right] \tag{22}$$

where

$$U_i(q) = c_1^\beta \left(V_i^\alpha(q) + 2V_i^\beta(q) \right) + 2c_1^\gamma \psi_\gamma(\|q_i - q_\gamma\|) \tag{23}$$

$$V_i^\alpha(q) = \sum_{j \in \mathcal{S}_i^\alpha} \psi(\|\hat{q}_{ij} - q_i\|_\sigma) \tag{24}$$

$$V_i^\beta(q) = \sum_{j \in \mathcal{S}^\beta} \psi(\|\hat{q}_{ij} - q_i\|_\sigma), \quad \psi_\gamma(z) := \sqrt{1 + z^2} - 1. \tag{25}$$

We also define the Hamiltonian at $t = 0$ as H_0.

In the case of particle α agents, $\hat{q}_{ij}$ in (24) is equal to q_j. Further, in the absence of obstacles, $V_i^\beta(q) = 0$ in (23), since $\mathcal{S}^\beta$ is empty. In such cases, we obtain the following result.

Theorem 1: Suppose that each α agent is a particle and consider an environment without obstacles. We also assume that p_γ is constant,

and that the initial Hamiltonian H_0 is finite. Then, the following statements hold.

i) The distance between each α agent and the γ agent is not greater than $\sqrt{\left(\frac{H_0}{c_1^\gamma}+1\right)^2 - 1}$.

ii) The velocities of all α agents approach the desired velocity p_γ asymptotically.

iii) The position of each α agent asymptotically converges to an equilibrium point satisfying $\nabla U(q) = 0$ where $U(q) = \sum_{i=1}^n U_i(q)$.

iv) If $H_0 < c_1^\beta(k+1)\psi(0)$ for some positive integer k, then at most k distinct pairs of agents could possibly collide ($k = 0$ guarantees a collision free motion).

Proof: See Appendix A. ∎

We next present a result in the presence of obstacles. In the same way as in [12], we consider two types of obstacles, for which the position of the β agent and its derivative are derived as follows.

i) For an obstacle with a hyperplane boundary that has a unit normal $\boldsymbol{a}_j$ and passes through a fixed point y_j, it holds that

$$\hat{q}_{ij} = (I - \boldsymbol{a}_j \boldsymbol{a}_j^T)q_i + \boldsymbol{a}_j \boldsymbol{a}_j^T y_j, \quad \dot{\hat{q}}_{ij} = (I - \boldsymbol{a}_j \boldsymbol{a}_j^T)p_i. \quad (26)$$

ii) For a spherical obstacle with radius R_j centered at y_j, we have

$$\hat{q}_{ij} = \lambda q_i + (1 - \lambda)y_j, \quad \dot{\hat{q}}_{ij} = \lambda(I - \boldsymbol{a}_j \boldsymbol{a}_j^T)p_i. \quad (27)$$

where $\lambda := R_j/\|q_i - y_j\|$ and $\boldsymbol{a}_j := (q_i - y_j)/\|q_i - y_j\|$.

In both cases, s_{ij} in (16) coincides with $\boldsymbol{a}_j$, and is perpendicular to $\dot{\hat{q}}_{ij}$.

Theorem 2: Under the assumptions in Theorem 1, except that there are two types of obstacles described above, we have $\dot{H}(q,p) \leq 0$, if

$$\sum_{i=1}^n \left[c_1^\beta p_\gamma^T \nabla_{q_i} V_i^\beta - c_2^\beta \sum_{j \in \mathcal{S}_i} a_{ij}\left(s_{ij}^T \tilde{p}_i\right)^2 - c_2^\gamma \tilde{p}_i^T \tilde{p}_i \right] \leq 0 \quad (28)$$

where $\tilde{p}_i := p_i - p_\gamma$. Furthermore, if there exists a finite time $t_1 \geq 0$ such that $\dot{H}(q,p) \leq 0, \forall t \geq t_1$, the following statements hold for $t \geq t_1$.

i) The distance between each α agent and the γ agent is not greater than $\sqrt{\left(\frac{H_1}{c_1^\gamma}+1\right)^2 - 1}$, where H_1 is the value of Hamiltonian at t_1.

ii) If $H_1 < c_1^\beta(k+1)\psi(0)$ for some positive integer k, then at most k collisions (inter agent collision or collision with an obstacle) occur concurrently ($k = 0$ guarantees a collision free motion).

Proof: See Appendix B. ∎

While the second and the third terms in the brackets in (28) do not have positive values, the first term

$$\sum_{i=1}^n c_1^\beta p_\gamma^T \nabla_{q_i} V_i^\beta = c_1^\beta \sum_{i=1}^n \sum_{j \in \mathcal{S}^\beta} p_\gamma^T \nabla_{q_i} \psi_\beta(\|\hat{q}_{ij} - q_i\|_\sigma) \quad (29)$$

might have a positive value. The value of (29) increases when obstacles in the neighborhood of each α agent are located in the direction of p_γ, i.e., the direction in which each α agent should move. Since the left hand side of (28) is equal to $\dot{H}$, the positive value of (29) makes it more difficult for $\dot{H} \leq 0$ to hold. Furthermore, even if H converges, it does not mean that velocity matching is achieved, if (29) has a positive value. Thus, velocity matching is not guaranteed before all obstacles in the environment are avoided. In other words, if robots permanently encounter obstacles, velocity matching is difficult to be achieved.
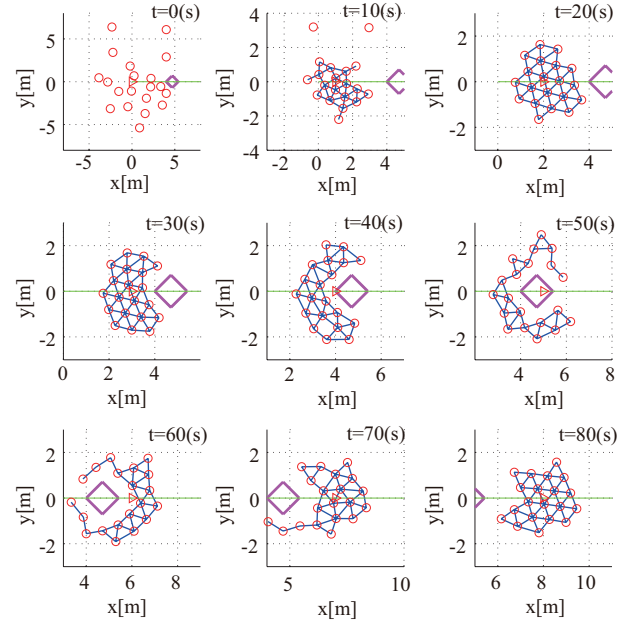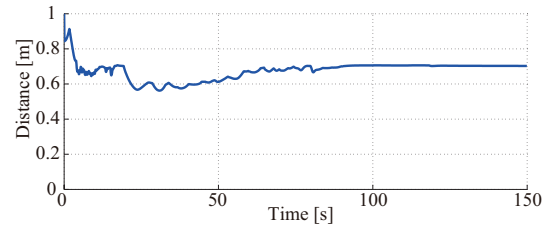

Fig. 4. Snapshots of a simulation.

Fig. 5. Minimum distance between objects (simulation).

VI. SIMULATION

In this section, we show simulation results in the case of a two-dimensional space, i.e., $m = 2$. The values of the design parameters are set as $\epsilon_0 = 0.001, \epsilon_1 = 0.1, \epsilon_2 = 0.01, h = 0.2, d = 0.8, r = 1.0, c_1^\beta = 2.5, c_1^\gamma = 0.2, c_2^\beta = 2\sqrt{c_1^\beta}$, and $c_2^\gamma = \sqrt{c_1^\gamma}$. The initial position of each α agent is randomly generated, while the initial velocity is zero. The γ agent moves from the origin along the x axis at a constant velocity of 0.1 m/s. The step size of the simulation is 0.1 s.

Fig. 4 shows snapshots of a simulation in the presence of a diamond-shaped obstacle and 20 α particle agents. The positions of an α agent and the γ agent are shown by "∘" and "▷", respectively. If a pair of α agents are interacting, i.e., the distance is less than r, they are connected by a line segment. Fig. 5 shows the minimum distance from the α-agents to nearby objects, i.e., nearby α agents and obstacles. It is shown from this figure that there is no collision with obstacles or other α agents. The average of the velocity of the α agents is shown by the solid line in Fig. 6. We also compute the sample standard deviation $\hat{s}$ of the velocities of α agents at each time step, and show the $\pm \hat{s}$ interval by the dash-dotted lines. This figure demonstrates that the velocity of each α agent converges to the velocity of the γ agent shown in the dashed line after the obstacle avoidance at around 30-80 s.

We next show simulation results in the presence of multiple obstacles. A problem in this case is that if the distance between two obstacles is too small, even a single particle α agent cannot pass
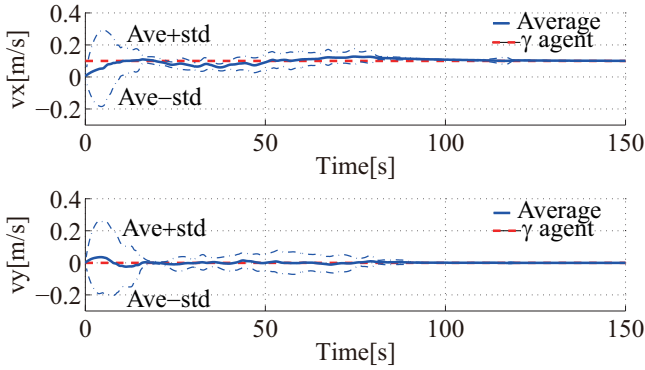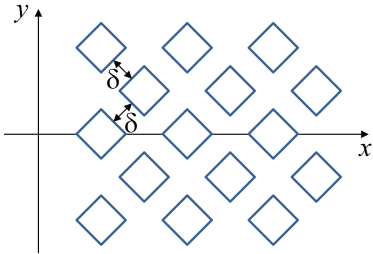
Fig. 6. Velocities of agents (simulation).

Fig. 7. Obstacles with the minimum distance δ in multiple obstacle cases.

Fig. 8. Rate of α agents that are not trapped in the obstacle area.

Fig. 9. Minimum distance between objects in multiple obstacle cases.

between the obstacles due to strong repulsive forces from them. As a result, the two obstacles constitute one concave obstacle that traps α agents. To illustrate this problem, we performed simulations for different values of the minimum distance δ of the uniformly placed fifteen obstacles as shown in Fig. 7. In Fig. 8, we show the average and standard deviation of the rate of α agents that passed through the obstacle area without being trapped, for 100 initial positions of α agents generated randomly. The simulation results show that it is difficult for α agents to pass between obstacles for $\delta < 1.4$. Fig. 9 shows the average and standard deviation of the the minimum distance between objects, i.e., α agents and obstacles, which implies that the collision avoidance performance is not significantly affected by δ. It is also seen from Fig. 9 that as the number of the α-agents increases, the minimum distance between objects decreases. A possible reason for this is that an α agent in the front side of the group is pushed toward an obstacle by a larger number of other α-agents. In all cases, the minimum distance between objects is larger than 0.47, which implies that collision avoidance is achieved. Furthermore, the velocity and position error between the leader and each α-agent that passed through the obstacle area are less than 4.4×10^{-2} m/s and 2.9 m respectively at the final simulation time (300 sec), which implies that velocity matching and group coherence are achieved. An example of paths of α agents for $\delta = 1.4$ and $n = 20$ is shown by blue lines in Fig. 10. If the distance between obstacles is large enough for an α agent to pass through, no significant difference between performances in single and multiple obstacle cases is found from these simulation results except for the increase in the number of the split/rejoin maneuvers for obstacle avoidance that are similar to the one in Fig. 4.

The algorithm is also tested for different layouts of obstacles other than that in Fig. 7. The positions of fifteen obstacles are selected from randomly generated coordinates such that the minimum distance between obstacles does not exceed $\delta = 1.4$ in order to make sure that robots can pass between obstacles. The initial states of α-
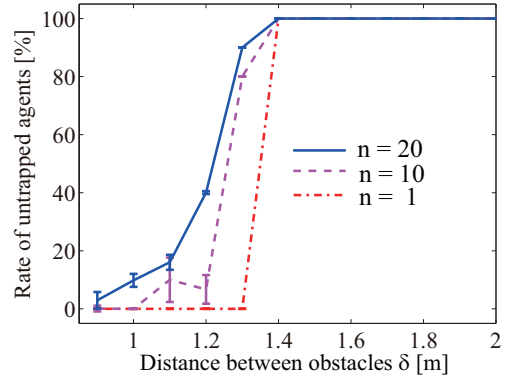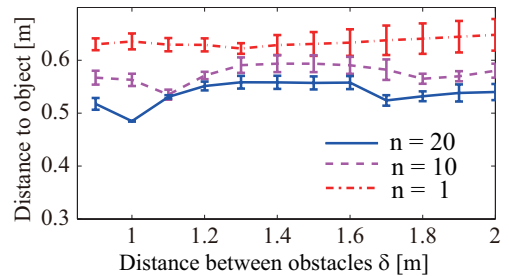
agents are also selected randomly. We performed 100 simulations in each case of $n = 1$, $n = 10$ and $n = 20$. In all cases, the minimum distance between objects is larger than 0.44, which implies that collision avoidance is achieved. Furthermore, the velocity and position error between the leader and each α-agent are less than 4.0×10^{-2} m/s and 2.6 m respectively at the final simulation time (500 sec), which implies that velocity matching and group coherence are achieved. From these results, the effectiveness of the proposed algorithm has been verified except for in the cases where the minimum distance between obstacles is too small for a robot to pass through. To overcome the problem that robots are trapped by obstacles, an emergency algorithm is required to detect and escape from such a deadlock situation, although we do not focus on that in this paper.

VII. EXPERIMENT

The proposed method was applied to a group of 8 robots. For each robot, we used a mobile robot platform (Kobuki, Yujin Robot). The model of this robot is described as

$$\begin{bmatrix} \dot{X}_i \\ \dot{Y}_i \\ \dot{\phi}_i \end{bmatrix} = \begin{bmatrix} \cos\phi_i & 0 \\ \sin\phi_i & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v_i \\ \omega_i \end{bmatrix}, \quad i = 1, \dots, 8 \quad (30)$$

where (X_i, Y_i) denotes the position of the center of the axle and ϕ_i is the orientation of robot i. The control inputs of this system are the translational and angular velocities (v_i, ω_i). To obtain the model as in (1) for this system, we define $q_i := [X_i + l\cos\phi_i, Y_i + l\sin\phi_i]^T$, which is the point with an offset $l = 0.1$ [m] from the center of the axle. Since it holds from (30) that

$$\dot{q}_i = B_i \begin{bmatrix} v_i \\ \omega_i \end{bmatrix} = p_i, \quad B_i := \begin{bmatrix} \cos\phi_i & -l\sin\phi_i \\ \sin\phi_i & l\cos\phi_i \end{bmatrix} \quad (31)$$
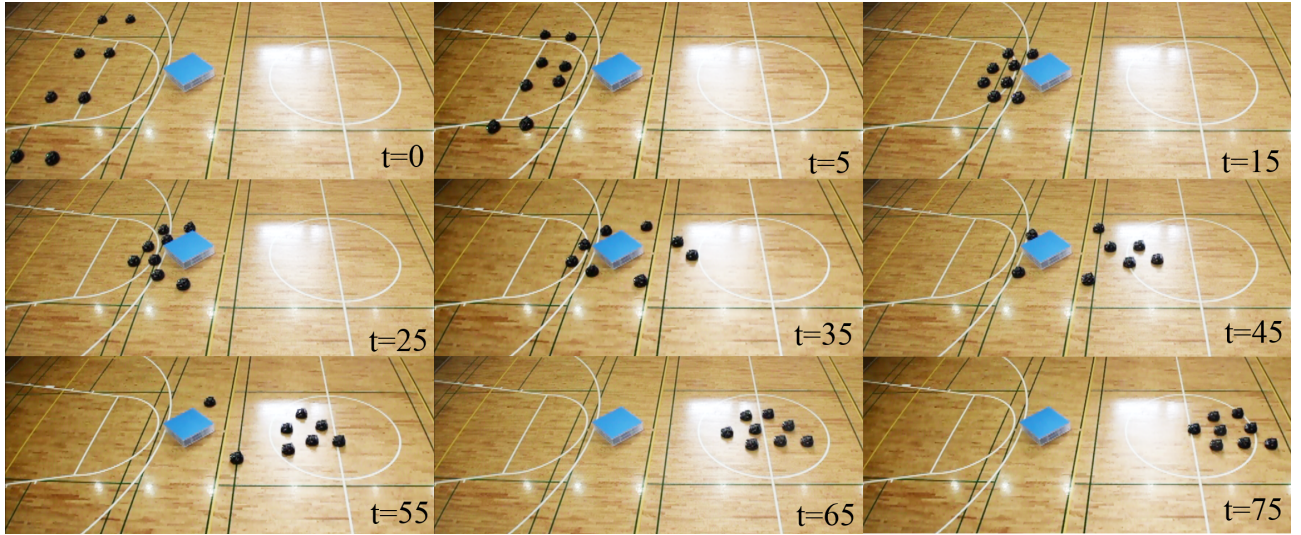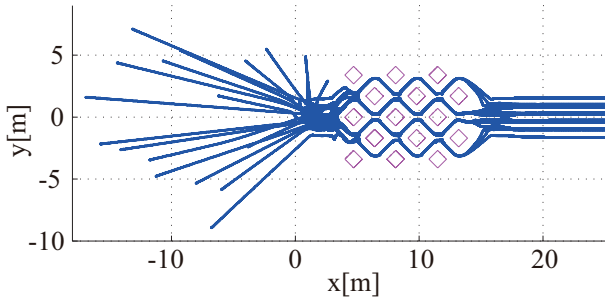
Fig. 11. Snapshots of an experiment.

Fig. 10. Example of paths of α agents for $\delta = 1.4$ (simulation).

Fig. 12. Minimum distance between agents and obstacles (experiment)

we obtain

$$w_i = B_i^{-1} p_i, \quad w_i := [v_i, \omega_i]^T. \tag{32}$$

Thus, p_i is computed by numerical integration of $\dot{p}_i = u_i$. We then apply w_i obtained by (32) to the system. It should be noted that although this is a simple way to linearize the system as in (1), it has a limitation that the internal dynamics in terms of ϕ_i is not always stable (i.e., the robots possibly spin about q_i) [27]. The position and orientation of the vehicle are measured by Gyrodometry [28] based on a gyro sensor (CRS09-22, Silicon Sensing Systems). Two laser range finders (URG-04LX, Hokuyo) are used to detect objects around each robot, since the field of view of each sensor is 240 deg. Point clouds from laser range finders on each robot are classified in such a way that two points of distance less than a given threshold (0.05 m) belong to the same group. Each group of point clouds is recognized as one object, and the closest point in each group is used as $\hat{q}_{ij}$ ($j \in N_i$, $i = 1, \ldots, 8$). The control input w_i above is computed by a computer (BOXDC53427HYE, Intel) mounted on each robot. The same values of the design parameters in the control law as in Section VI are used.

Fig. 11 contains snapshots of an experiment in the presence of an obstacle; Fig. 12 shows the minimum distance from the center of a robot to objects (i.e., nearby robots and obstacles). Since each robot has a circular shape with a radius of less than 0.18 m, Fig. 12 indicates that no collision occurred. The average velocity of the α agents is shown by the solid line in Fig. 13, and the dash-dotted lines
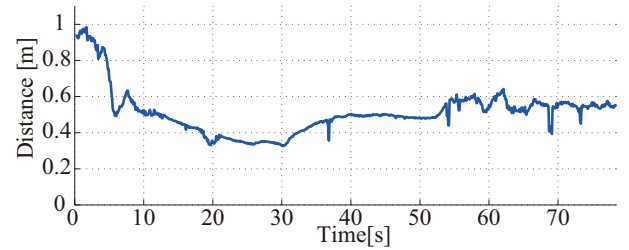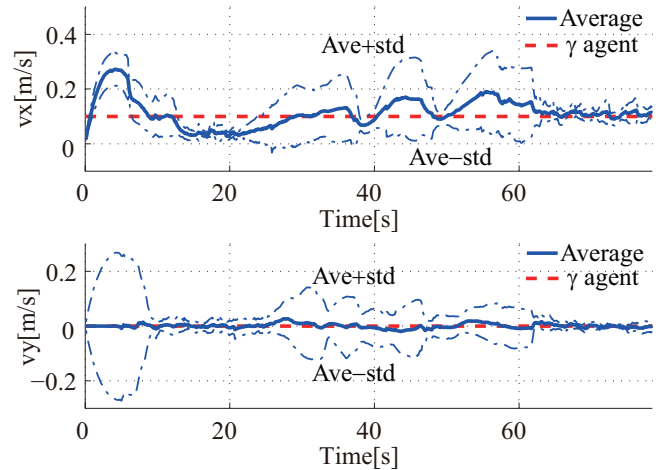

Fig. 13. Velocities of agents (experiment)

indicate the $\pm \hat{s}$ interval for the sample standard deviation $\hat{s}$ of the velocities of α agents. The figure demonstrates that the velocity of each α agent is kept around the velocity of the γ agent shown in the dashed line after the obstacle avoidance at around 20-65 s, although some small fluctuation in the velocities are inevitable, unlike in the simulation.

VIII. Conclusions

In this paper, we have proposed a flocking algorithm that does not distinguish between robots and obstacles. Since all detected objects are regarded as obstacles in the proposed algorithm, velocity information on the neighboring robots is not necessary. We have demonstrated that the proposed algorithm maintains the desired properties of existing flocking algorithms despite the limited information used. The effectiveness of the algorithm has been demonstrated by both simulations and real robot experiments in the presence of a single obstacle. We have also verified by simulations that the proposed algorithm is effective in multiple obstacle cases, if the distance between obstacles is large enough for an α agent to pass through. On the other hand, if the distance between two obstacles is too small, α agents are trapped by the two obstacles that constitute one concave obstacle. To overcome this problem, we need to add an emergency algorithm for robots to detect and escape from such a deadlock situation, which is a possible future work.

Appendix A
Proof of Theorem 1

We first show that $\dot{H}(q, p) \leq 0$. In the case of no obstacles and particle α agents, $U_i(q)$ in (23) is written as

$$U_i(q) = c_1^\beta V_i^\alpha(q) + 2c_1^\gamma \psi_\gamma(\|\tilde{q}_i\|) \tag{33}$$

$$V_i^\alpha(q) = \sum_{j \in \mathcal{S}_i^\alpha} \psi(\|q_j - q_i\|_\sigma) = \sum_{j \in \mathcal{S}_i^\alpha} \psi(\|\tilde{q}_j - \tilde{q}_i\|_\sigma) \tag{34}$$

where $\tilde{q}_i := q_i - q_\gamma$. Differentiating U_i gives

$$\sum_{i=1}^n \dot{U}_i(q) = c_1^\beta \sum_{i=1}^n \sum_{j \in \mathcal{S}_i^\alpha} \left(\dot{\tilde{q}}_i^T \nabla_{\tilde{q}_i} \psi(\|\tilde{q}_j - \tilde{q}_i\|_\sigma) \right.$$
$$\left. + \dot{\tilde{q}}_j^T \nabla_{\tilde{q}_j} \psi(\|\tilde{q}_j - \tilde{q}_i\|_\sigma) \right) + 2c_1^\gamma \sum_{i=1}^n \dot{\tilde{q}}_i^T \nabla_{\tilde{q}_i} \psi_\gamma(\|\tilde{q}_i\|)$$
$$= 2c_1^\beta \sum_{i=1}^n \dot{\tilde{q}}_i^T \nabla_{q_i} V_i^\alpha(q) + 2c_1^\gamma \sum_{i=1}^n \dot{\tilde{q}}_i^T \mu(\tilde{q}_i). \tag{35}$$

From the assumption that p_γ is constant, we have

$$\dot{\tilde{p}}_i = \dot{p}_i - \dot{p}_\gamma = \dot{p}_i = u_i. \tag{36}$$

Thus, it holds from $\dot{\tilde{q}}_i = \tilde{p}_i$ that

$$\dot{H}(q, p) = \sum_{i=1}^n \tilde{p}_i^T \left[c_1^\beta \nabla_{q_i} V_i^\alpha(q) + c_1^\gamma \mu(\tilde{q}_i) + u_i \right]. \tag{37}$$

In the case of no obstacles and particle α agents, the control input u_i in (17) can be written as

$$u_i = -c_1^\beta \sum_{j \in N_i^\alpha} \nabla_{q_i} \psi(\|q_j - q_i\|_\sigma)$$
$$+ c_2^\beta \sum_{j \in N_i^\alpha} a_{ij}(q)(\hat{p}_{ij} - p_i) - c_1^\gamma \mu(\tilde{q}_i) - c_2^\gamma \tilde{p}_i \tag{38}$$

by replacing $\hat{q}_{ij}$ and N_i in (18) by q_j and N_i^α, respectively. Further, since it follows from the definitions of ψ and a_{ij} that

$$\nabla_{q_i} \psi(\|q_j - q_i\|_\sigma) = 0, \quad a_{ij}(q) = 0, \quad \forall j \notin N_i^\alpha \tag{39}$$

we can replace N_i^α in (38) by $\mathcal{S}_i^\alpha$. Thus, we have

$$u_i = -c_1^\beta \nabla_{q_i} V_i^\alpha(q) + c_2^\beta \sum_{j \in \mathcal{S}_i^\alpha} a_{ij}(q)(\tilde{p}_{ij} - \tilde{p}_i)$$
$$- c_1^\gamma \mu(\tilde{q}_i) - c_2^\gamma \tilde{p}_i \tag{40}$$

where $\tilde{p}_{ij} := \hat{p}_{ij} - p_\gamma$ which is rewritten as

$$\tilde{p}_{ij} = \tilde{p}_i - s_{ij} s_{ij}^T \tilde{p}_i \tag{41}$$

from (15). By substituting (40) into (37), we have

$$\dot{H}(q, p) = \sum_{i=1}^n \left[c_2^\beta \sum_{j \in \mathcal{S}_i^\alpha} a_{ij}(q) \tilde{p}_i^T (\tilde{p}_{ij} - \tilde{p}_i) - c_2^\gamma \tilde{p}_i^T \tilde{p}_i \right]. \tag{42}$$

It holds from (41) that

$$\tilde{p}_i^T (\tilde{p}_{ij} - \tilde{p}_i) = -\tilde{p}_i^T s_{ij} s_{ij}^T \tilde{p}_i = -\left(s_{ij}^T \tilde{p}_i\right)^2 \leq 0. \tag{43}$$

Thus, since $c_2^\beta > 0$ and $a_{ij}(q) \geq 0$, the first term in the right hand side of (42) is not positive. Therefore, since $-c_2^\gamma \tilde{p}_i^T \tilde{p}_i \leq 0$ for the second term, we have $\dot{H}(q, p) \leq 0$.

Once $\dot{H}(q, p) \leq 0$ is proved, the statements i)-iv) can be proved in the same way as in [13], as briefly described below.

i) It holds from $\dot{H}(q, p) \leq 0$ that $H(q(t), p(t)) \leq H_0$ at $t \geq 0$. Since $V_i^\alpha(q) \geq 0$, $\tilde{p}_i^T \tilde{p}_i \geq 0$, and $c_1^\beta, c_1^\gamma > 0$ in (22), we have

$$c_1^\gamma \psi_\gamma(\|q_i - q_\gamma\|) \leq H(q, p) \leq H_0. \tag{44}$$

Thus, from the definition of ψ_γ in (25), we obtain

$$\|\tilde{q}_i\| = \|q_i - q_\gamma\| \leq \sqrt{\left(\frac{H_0}{c_1^\gamma} + 1\right)^2 - 1} \tag{45}$$

which completes the proof of i). Similarly, it holds from (22) that

$$\tilde{p}_i^T \tilde{p}_i \leq 2H(q, p) \leq 2H_0 \tag{46}$$

which implies

$$\|\tilde{p}_i\| = \|p_i - p_\gamma\| \leq \sqrt{2H_0}. \tag{47}$$

ii) From $\dot{H}(q, p) \leq 0$, the following set Ω is an invariant set.

$$\Omega = \left\{ \left[\tilde{q}^T, \tilde{p}^T\right]^T : H(q, p) \leq H_0 \right\} \tag{48}$$

Since (45) and (47) imply that the relative position and velocity from each agent to the γ agent is bounded, Ω is a compact set. From the LaSalle Invariance Principle [29], all trajectories of the α agents that start from Ω will converge to the largest invariant set inside the following region S.

$$S = \left\{ \left[\tilde{q}^T, \tilde{p}^T\right]^T : \dot{H}(q, p) = 0 \right\} \tag{49}$$

Since $\dot{H} = 0$ implies $\tilde{p}_i = 0, i = 1, 2, \cdots, n$ from (42)-(43), it is proved that ii) is satisfied.

iii) From ii) proved above, it holds asymptotically that $\tilde{p}_i = 0$, i.e. $p_i = p_\gamma$. Then, since $u_i = \dot{p}_i = \dot{p}_\gamma = 0$ under the assumption that p_γ is constant, it holds from (40) that

$$0 = -c_1^\beta \nabla_{q_i} V_i^\alpha(q) - c_1^\gamma \mu(\tilde{q}_i). \tag{50}$$

Therefore, since it holds that

$$\sum_{j \in \mathcal{S}_i^\alpha} \nabla_{q_i} U_j = c_1^\beta \sum_{j \in \mathcal{S}_i^\alpha} \sum_{k \in \mathcal{S}_j^\alpha} \nabla_{q_i} \psi(\|q_k - q_j\|_\sigma)$$
$$= c_1^\beta \sum_{j \in \mathcal{S}_i^\alpha} \nabla_{q_i} \psi(\|q_i - q_j\|_\sigma) = c_1^\beta \nabla_{q_i} V_i^\alpha(q) \tag{51}$$

we have

$$\nabla_{q_i} U(q) = \sum_{j=1}^n \nabla_{q_i} U_j(q) = \nabla_{q_i} U_i(q) + \sum_{j \in \mathcal{S}_i^\alpha} \nabla_{q_i} U_j(q)$$
$$= 2\left(c_1^\beta \nabla_{q_i} V_i^\alpha(q) + c_1^\gamma \mu(\tilde{q}_i)\right) = 0 \tag{52}$$

for U_i in (23), which implies that iii) is satisfied.

iv) For a proof by contradiction, suppose that more than k distinct pairs of α agents collide at a time t^*. Since $q_i = q_j$ when a pair of α agents (i, j) collide, we obtain

$$H(q, p) \geq \frac{c_1^\beta}{2} \sum_{i=1}^n \sum_{j \in \mathcal{S}_i^\alpha} \psi(\|q_j - q_i\|_\sigma) \geq c_1^\beta (k+1) \psi(0).$$

Thus, it holds from $H_0 \geq H(q, p)$ that $H_0 \geq c_1^\beta (k+1) \psi(0)$. This contradicts the assumption that $H_0 < c_1^\beta (k+1) \psi(0)$. Hence, it is proved that iv) is satisfied.

APPENDIX B
PROOF OF THEOREM 2

By differentiating V_i^β in (25), we have

$$\dot{V}_i^\beta = \sum_{j \in \mathcal{S}^\beta} \left(\dot{\hat{q}}_{ij}^T \nabla_{\hat{q}_{ij}} \psi + \dot{q}_i^T \nabla_{q_i} \psi \right). \tag{53}$$

It follows from the definition of ψ in (19) that

$$\nabla_{\hat{q}_{ij}} \psi(\|\hat{q}_{ij} - q_i\|_\sigma) = \frac{\phi(\|\hat{q}_{ij} - q_i\|_\sigma)}{\sqrt{\epsilon_2^2 + \epsilon_1 \|\hat{q}_{ij} - q_i\|^2}} (\hat{q}_{ij} - q_i) \tag{54}$$

which is parallel to s_{ij} in (16) and perpendicular to $\dot{\hat{q}}_{ij}$ in (26) and (27). Thus, we have $\dot{\hat{q}}_{ij}^T \nabla_{\hat{q}_{ij}} \psi = 0$, which implies

$$\dot{V}_i^\beta = \dot{q}_i^T \nabla_{q_i} V_i^\beta(q) = \dot{\hat{q}}_i^T \nabla_{q_i} V_i^\beta(q) + \dot{q}_\gamma^T \nabla_{q_i} V_i^\beta(q). \tag{55}$$

Therefore, it follows from (37) and (55) that

$$\dot{H}(q, p) = \sum_{i=1}^n \tilde{p}_i^T \left[c_1^\beta \left(\nabla_{q_i} V_i^\alpha(q) + \nabla_{q_i} V_i^\beta(q) \right) + c_1^\gamma \mu(\tilde{q}_i) + u_i \right]$$
$$+ \sum_{i=1}^n c_1^\beta p_\gamma^T \nabla_{q_i} V_i^\beta(q). \tag{56}$$

Since u_i in (17) can be written as

$$u_i = -c_1^\beta \left(\nabla_{q_i} V_i^\alpha(q) + \nabla_{q_i} V_i^\beta(q) \right)$$
$$+ c_2^\beta \sum_{j \in \mathcal{S}_i} a_{ij}(q)(\hat{p}_{ij} - p_i) - c_1^\gamma \mu(\tilde{q}_i) - c_2^\gamma \tilde{p}_i \tag{57}$$

it holds from (15) and (56) that

$$\dot{H}(q, p) = \sum_{i=1}^n \left[c_1^\beta p_\gamma^T \nabla_{q_i} V_i^\beta(q) - c_2^\beta \sum_{j \in \mathcal{S}_i} a_{ij} \left(s_{ij}^T \tilde{p}_i \right)^2 - c_2^\gamma \tilde{p}_i^T \tilde{p}_i \right].$$

Therefore, $\dot{H}(q, p) \leq 0$, if (28) is satisfied. Once $\dot{H}(q, p) \leq 0$ is proved, i) can be proved in the same way as in Theorem 1 by replacing H_0 by H_1.

To prove ii) by contradiction, suppose that more than k collisions occur at a time t^*. Then, we have

$$H(q, p) \geq c_1^\beta \sum_{i=1}^n \left[\sum_{j \in \mathcal{S}_i^\alpha} \frac{\psi(\|q_j - q_i\|_\sigma)}{2} + \sum_{j \in \mathcal{S}^\beta} \psi(\|q_j - q_i\|_\sigma) \right]$$
$$\geq c_1^\beta (k+1) \psi(0) \tag{58}$$

which implies $H_1 \geq c_1^\beta (k+1) \psi(0)$. This contradicts the assumption that $H_1 < c_1^\beta (k+1) \psi(0)$. Hence, it is proved that ii) is satisfied.

REFERENCES

[1] C. W. Reynolds, "Flocks, herds, and schools: A distributed behavioral model," *Comput. Graph.*, vol. 21, no. 4, pp. 25–34, Jul, 1987.
[2] T. Vicsek, A. Czirok, E. B. Jacob, I. Cohen, and O. Schochet, "Novel type of phase transition in a system of self-driven particles," *Phys. Rev. Lett.*, vol. 75, no. 6, pp. 1226–1229, Aug., 1995.
[3] J. Toner and Y. Tu, "Flocks, herds, and schools: A quantitative theory of flocking," *Phys. Rev. E*, vol. 58, no. 4, pp. 4828–4858, Oct, 1998.
[4] A. Mogilner and L. Edelstein-Keshet, "A non-local model for a swarm," *J. Math. Biol.*, vol. 38, pp. 534–570, 1999.
[5] N. E. Leonard and E. Friorelli, "Virtual leaders, artificial potentials and coordinated control of groups," in *Proc. 40th IEEE Conf. Decision Control*, Dec. 2001, pp. 2968–2973.
[6] Y. Liu, K. M. Passino, and M. M. Polycarpou, "Stability analysis of M-dimensional asynchronous swarms with a fixed communication topology," *IEEE Trans. Autom. Contr.*, vol. 48, no. 1, pp. 76–95, Jan. 2003.
[7] V. Gazi and K. M. Passino, "Stability analysis of swarms," *IEEE Trans. Autom. Contr.*, vol. 48, no. 4, pp. 692–697, Apr. 2003.
[8] A. Jadbabaie, J. Lin, and A. S. Morse, "Coordination of groups of mobile autonomous agents using nearest neighbor rules," *IEEE Trans. Autom. Contr.*, vol. 48, no. 6, pp. 988–1001, Jun. 2003.
[9] H. G. Tanner, A. Jadbabaie, and G. J. Pappas, "Stable flocking of mobile agents, part II: Dynamic topology," in *Proc. 42nd IEEE Conf. Decision Control*, Dec. 2003, pp. 2016–2021.
[10] H. G. Tanner, A. Jadbabaie, and G. J. Pappas, "Flocking in fixed and switching networks," *IEEE Trans. Autom. Contr.*, vol. 52, no. 5, pp. 863–868, May 2007.
[11] H. G. Tanner, "Flocking with obstacle avoidance in switching networks of interconnected vehicles," in *Proc. IEEE Int. Conf. Robot. and Autom.*, Apr. 2004, pp. 3006–3011.
[12] R. Olfati-Saber, "Flocking for multi-agent dynamic systems: Algorithms and theory," *IEEE Trans. Autom. Contr.*, vol. 51, no. 3, pp. 401–420, Mar. 2006.
[13] H. Su, X. Wang, and Z. Lin, "Flocking of multi-agents with a virtual leader," *IEEE Trans. Autom. Contr.*, vol. 54, no. 2, pp. 293–307, Feb. 2009.
[14] M. M. Zovlanos, H. G. Tanner, A. Jadbabaie, and G. J. Pappas, "Hybrid control for connectivity preserving flocking," *IEEE Trans. Autom. Contr.*, vol. 54, no. 12, pp. 2869–2875, Dec. 2009.
[15] H. Su, X. Wang, and G. Chen, "A connectivity-preserving flocking algorithm for multi-agent systems based only on position measurements," *International Journal of Control*, vol. 82, no. 7, pp. 1334–1343, Jul. 2009.
[16] R. Olfati-Saber and R. M. Murray, "Consensus problems in networks of agents with switching topology and time-delays," *IEEE Trans. Autom. Contr.*, vol. 49, no. 9, pp. 1520–1533, Sep. 2004.
[17] W. Ren and R. W. Beard, "Consensus seeking in multi-agent systems under dynamically changing interaction topologies," *IEEE Trans. Autom. Contr.*, vol. 50, no. 5, pp. 655–661, May 2005.
[18] W. Ren, "On consensus algorithms for double-integrator dynamics," *IEEE Trans. Autom. Contr.*, vol. 53, no. 6, pp. 1503–1509, Jul. 2008.
[19] J. Qin, W. Zheng, H. Gao, "Consensus of multiple second-order vehicles with a time-varying reference signal under directed topology," *Automatica*, vol. 47, no. 9, pp. 1983–1991, 2011.
[20] J. P. Desai, J. P. Ostrowski, and V. Kumar, "Modeling and control of formations of nonholonomic mobile robots," *IEEE Trans. Robot. Autom.*, vol. 17, no. 6, pp. 905–908, Dec. 2001.
[21] P. Ögren, M. Egerstedt, and X. Hu, "A control Lyapunov function approach to multiagent coordination," *IEEE Trans. Robot. Autom.*, vol. 18, no. 5, pp. 847–851, Oct. 2002.
[22] W. Ren and R. Beard, "Decentralized scheme for spacecraft formation flying via the virtual structure approach," *AIAA Journal of Guidance, Control, and Dynamics*, vol. 27, no. 1, pp. 73–82, 2004.
[23] J. A. Fax and R. M. Murray, "Information flow and cooperative control of vehicle formations," *IEEE Trans. Robot. Autom.*, vol. 49, no. 9, pp. 1465–1476, Sep. 2004.
[24] H. G. Tanner, G. J. Pappas, and V. Kumar, "Leader-to-formation stability," *IEEE Trans. Robot. Autom.*, vol. 20, no. 3, pp. 443–455, Jun. 2004.
[25] H. Fukushima, K. Kon, and F. Matsuno, "Model predictive formation control using branch-and-bound compatible with collision avoidance problems," *IEEE Trans. Robot.*, vol. 29, no. 5, pp. 1308–1317, 2013.
[26] K. Yoshida, H. Fukushima, K. Kon, and F. Matsuno, "Control of a group of mobile robots based on formation abstraction and decentralized locational optimization," *IEEE Trans. Robot.*, vol. 30, no. 3, pp. 550–565, Jun. 2014.
[27] X. Yun and Y. Yamamoto, "Internal dynamics of a wheeled mobile robot," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Jul. 1993, pp. 1288–1294.
[28] J. Borenstein and L. Feng, "Gyrodometry: a new method for combining data from gyros and odometry in mobile robots," in *Proc. IEEE Int. Conf. Robot. and Autom.*, Apr. 1996, pp. 423-428.
[29] H. K. Khalil, *Nonlinear Systems*, Upper Saddle River, NJ: Prentice Hall, 2002.