ARTICLE TEMPLATE

Decentralized Navigation and Collision Avoidance for Robotic Swarm with Heterogeneous Abilities

Yoshinori Sano[a], Takahiro Endo[a], Takumi Shibuya[a], and Fumitoshi Matsuno[a]

[a]Dept. of Mechanical Engineering and Science, Kyoto University, Kyoto, Japan

ARTICLE HISTORY
Compiled August 21, 2022

ABSTRACT
This paper proposes a decentralized navigation method with collision avoidance for a robotic swarm whose individuals possess heterogeneous abilities, such as sensing range and maximum speed. In this method, each agent distributedly constructs and maintains a local directed connection with another agent using only local information, which is relative distance. Moreover, all agents always maintain some distance from other agents to avoid collision. As a result, one leader robot can guide an entire swarm of robots to their destination, and the other robots can follow the leader while maintaining connectivity and not colliding with others. We prove the above mathematically, and we demonstrate the validity of the proposed method by numerical simulation and experimentation.

KEYWORDS
Robotic swarms; Decentralized control; Guidance navigation and control; Collision avoidance

1. Introduction

Research on robotic swarms has increased in recent years. Robotic swarms are useful from several viewpoints: they are robust against failures of individual agents, flexible with respect to environments and tasks, and easily scalable [1–3]. Scalability is possible because individual agents act in a distributed manner using only local information. Thus, it is important to control the robots by a decentralized controller.

Thanks to their flexibility in adapting to different environments and tasks, robotic swarms are expected to be applicable to main tasks such as cooperative coverage [4], surveillance [5], and transportation [6]. The most basic task common to them is moving to a destination together as a group. Each agent can acquire only local information using onboard sensors or wireless communication equipment.

There are many studies on the navigation of robotic swarms while maintaining connectivity [7–14], but in most of them, all of the agents in a swarm have the same homogeneous abilities. Some of those studies considered distributed connectivity control for agents, where the communication or sensing region of each agent is limited, but the region is equal for all agents. In practice, however, when performing actual tasks, all agents will not always have the same abilities. In addition, a group of agents with different abilities will have a greater potential to do a wider range of tasks than a

CONTACT T. Endo. Email: endo@me.kyoto-u.ac.jp

group of agents with homogeneous abilities [15,16]. For instance, in our prior work [17], we proposed a navigation method for a robotic swarm whose individuals possess different abilities. Then, in our subsequent study [18], we added an FoV constraint and collision avoidance between an agent and its target. In addition, a distributed flocking controller has been proposed for agents with limited communication and sensing regions that are equal for all agents but heterogeneous input constraints, where each agent has a different maximum input constraint [19]. However, neither of these works considered collision avoidance between an agent and the other non-target agents. A previous work [20] also proposed a method of navigating a robotic swarm whose individuals possessed heterogeneous abilities while maintaining connectivity, but it did not consider collision avoidance.

When considering the navigation of a real robotic swarm, it is crucial not to adopt a trajectory where agents will collide, as this will lead to failure. Previous studies have dealt with collision avoidance between agents during the guidance of group robots [21–25]. However, those studies have limitations. In several of them, the agents in a swarm have homogeneous abilities [23–25]. In [21], A. Filotheou et al. proposed a connectivity maintenance method with collision avoidance of a robotic swarm whose individuals possessed heterogeneous abilities. However, the controller requires the current and desired global positions and the velocities of the agents. In general, for a controller to know this information, it must receive global information from all robots via a common coordinate system. In addition, connectivity preservation and collision avoidance methods have been proposed for a robotic swarm [22]. However, collision avoidance between a follower and the other non-target followers has not been considered, and the robotic swarm has no heterogeneous ability such as velocity constraint.

To our best knowledge, there has not yet been any research about a navigation method without global information and with collision avoidance of robotic swarms having heterogeneous abilities, such as sensing range and maximum speed.

In this paper, we propose a distributed guidance control method for a robotic swarm having such heterogeneous abilities. A swarm has only one leader that knows the maximum speed limit of all agents, but it cannot access their real-time information such as position and speed. The leader can move freely under the maximum speed limit. On the other hand, each follower selects a target agent and maintains connectivity with it so as not to collide with other agents while satisfying maximum speed limitations. In addition, the controller of the followers does not need global information.

The main features of this paper are as follows. First, we consider a robotic swarm having heterogeneous abilities such as sensing range and maximum speed. Second, we propose a distributed navigation method that keeps the group connected and prevents agents from colliding with each other. In this method, robots do not require communication equipment and global information, and robots with various mobilities and sensing functions can be treated as a swarm. However, by simply combining the existing methods, we could not prove collision avoidance while maintaining group connectivity. Therefore, we realized collision avoidance between agents by switching the connectivity topology according to the situation.

This paper is organized as follows. In Section 2, we show the problem setting, and we propose a navigation method in Section 3. Section 4 describes the mathematical analysis. Section 5 shows the simulation result, and Section 6 presents the experimental result. These results show the effectiveness of the proposed method. Finally, Section 7 concludes the paper.

2. Problem Settings

2.1. Model of Agents

This paper considers a leader-follower problem in which one or more followers are guided by one leader. Let us consider a two-dimensional planar environment $D \in \mathbb{R}^2$ with no obstacles as the environment in which the robot moves. The number of agents is $n+1$. ID $1, 2, \cdots, n$ are assigned to the followers, and $n+1$ to a leader. This number is given for convenience of description: it is not necessary for the agent to actually identify itself or to identify other agents. Hereafter, the set of all agent numbers is expressed as $\mathcal{A} = \{1, 2, \cdots, n+1\}$ and the set of follower numbers as $\mathcal{F} = \{1, 2, \cdots, n\}$.

Agent $i \in \mathcal{A}$ is assumed to be circular with radius $\rho_i^{\rm size}$ whose center position is $\boldsymbol{x}_i(t) \in D$. The dynamics of agent i is described as follows:

$$\dot{\boldsymbol{x}}_i(t) = \boldsymbol{u}_i(t), \tag{1}$$

where $\boldsymbol{u}_i(t) \in \mathbb{R}^2$ is the velocity input. This is a simple dynamics, but many realistic second-order ones can be converted into this form using a speed-reference-type motor amplifier with speed feedback and the high-gain characteristics of the amplifier [26].

Follower $i \in \mathcal{F}$ has the following velocity constraint.

$$\|\boldsymbol{u}_i(t)\| \leq U_i, \tag{2}$$

where U_i is the maximum speed. In addition, the follower $i \in \mathcal{F}$ has the following sensing region $S_i(t)$:

$$S_i(t) := \{\boldsymbol{X} \in D \mid \|\boldsymbol{x}_i(t) - \boldsymbol{X}\| \leq \rho_i,\ \rho_i > 0\}. \tag{3}$$

That is, the follower $i \in \mathcal{F}$ can sense within the circle of radius ρ_i centered on its center. If the agent $j \in \mathcal{A}$ is in $S_i(t)$, the follower $i \in \mathcal{F}$ can measure the following relative position of the agent j:

$$\boldsymbol{x}_j^i(t) = \boldsymbol{x}_j^{i\min}(t) - \boldsymbol{x}_i(t). \tag{4}$$

Here, $\boldsymbol{x}_j^{i\min}(t)$ is the closest point on the agent j from the follower i and defined as

$$\boldsymbol{x}_j^{i\min}(t) = \arg\min_{\boldsymbol{x} \in \mathcal{O}_j^i(t)} \|\boldsymbol{x} - \boldsymbol{x}_i(t)\|, \tag{5}$$

where $\mathcal{O}_j^i(t)$ is the set of points on the agent j detected by the follower i at time t in the environment D. As long as an agent on $S_i(t)$ continues to be in $S_i(t)$, the individual can be distinguished from other agents in $S_i(t)$.

2.2. Connectivity

Graph representation is used to express connectivity between agents. Let us consider a directed graph $\mathcal{G}(t) = \mathcal{N} \times \mathcal{E}(t)$, where $\mathcal{N} = \{N_1, N_2, \cdots, N_{n+1}\}$ is a set of nodes and $\mathcal{E}(t)$ is a set of directed edges between nodes at time t. The directional edge from N_j to N_i is expressed as $E_{ji} \in \mathcal{E}(t)$. If E_{ji} exists, N_j is said to be the parent of N_i. A path that can follow a directed edge is called a directed path, and a node that can be

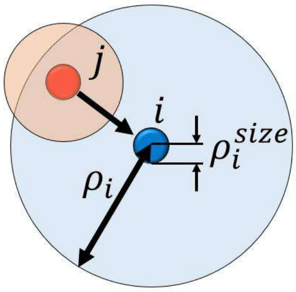


Figure 1. ij semi-connection.

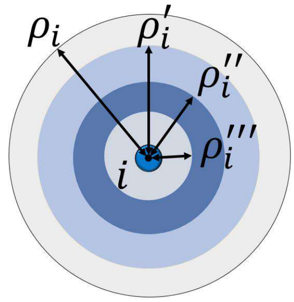


Figure 2. Various distances of agent i.

reached from N_j via a directed path is called a descendant of N_j. When N_{n+1} has no parent and all nodes except N_{n+1} have only one parent and are descendants of N_{n+1}, $\mathcal{G}(t)$ is said to be a spanning tree that is rooted at N_{n+1}.

The above expression is applied to a robotic swarm by making agent $i \in \mathcal{A}$ correspond to N_i. Suppose that there is a directional edge $E_{ji} \in \mathcal{E}(t)$ from N_j to N_i when $\boldsymbol{x}_j^{i\mathrm{min}}(t) \in S_i(t)$.

Definition 1 (Leader Semi-Connected [17])**.** When E_{ji} exists, agents $i \in \mathcal{A}$ and $j \in \mathcal{A}$ are said to be ij semi-connected (see Fig. 1). When N_i is a descendant of N_{n+1}, the follower $i \in \mathcal{F}$ is said to be leader semi-connected (LSC). When any follower is LSC, that is, when $\mathcal{G}(t)$ contains a spanning tree rooted in N_{n+1}, the agent group is LSC.

Now, we define positive constants ρ_i', ρ_i'', ρ_i''' such that $\rho_i^{\mathrm{size}} \leq \rho_i''' < \rho_i'' < \rho_i' < \rho_i$ (Fig. 2). Here, ρ_i' and ρ_i'' are the distances for switching control input. Using our method, none of the agents comes closer than ρ_i'''. For follower $i \in \mathcal{F}$, we define the following areas:

$$\begin{cases} S_i'(t) & := \{\boldsymbol{X} \in D \,|\, \|\boldsymbol{x}_i(t) - \boldsymbol{X}\| \leq \rho_i'\}, \\ S_i''(t) & := \{\boldsymbol{X} \in D \,|\, \|\boldsymbol{x}_i(t) - \boldsymbol{X}\| \leq \rho_i''\}. \end{cases} \tag{6}$$

And we set the boundaries of these areas as follows:

$$\begin{cases} \partial S_i'(t) & := \{\boldsymbol{X} \in D \,|\, \|\boldsymbol{x}_i(t) - \boldsymbol{X}\| = \rho_i'\}, \\ \partial S_i''(t) & := \{\boldsymbol{X} \in D \,|\, \|\boldsymbol{x}_i(t) - \boldsymbol{X}\| = \rho_i''\}. \end{cases} \tag{7}$$

Now we introduce the assumption used in this paper.

Assumption 1 (Initial Position)**.** Suppose that at $t = 0$, for any follower $i \in \mathcal{F}$, there is at least one agent in $S_i'(0) \setminus \{\partial S_i'(0) \cup S_i''(0)\}$ and that no agents exist in $S_i''(0)$. Also assume that all agents stop at $t = 0$.

With this assumption, every follower can sense at least one candidate agent of the target at $t = 0$.

2.3. Control Objective

In this study, we propose the leader's constraint and the follower's distributed control rules. Based on these rules, for any t, all followers maintain the velocity constraint (2), remain LSC, and avoid colliding with other agents. The leader moves freely without sensing, and each follower moves according to information about itself and the relative positions of other agents obtained by sensing.

3. Proposed Method

We propose a method for constructing and maintaining a spanning tree $\mathcal{G}(t)$ with collision avoidance. Each follower selects another agent as a target, and moves to maintain semi-connection with that agent while switching the target appropriately. As a result, the agent group becomes LSC, and there is no collision between agents. The proposed method consists of the velocity constraint for the leader, the target determination of each follower, and a control input.

3.1. Leader's Constraint

To maintain LSC of the agents, the following constraint is imposed on the leader's maximum speed U_{n+1}.

Assumption 2 (Leader's maximum speed)**.** The leader's maximum speed U_{n+1} satisfies the following constraint:

$$U_{n+1} \leq \min_{i \in \mathcal{F}} U_i. \tag{8}$$

This is needed when we prove the connectivity and the velocity constraint (2).

3.2. Target Determination

The follower $i \in \mathcal{F}$ determines, as its target, one agent $j \in \mathcal{A}$ when $\boldsymbol{x}_j^{i\min}(t) \in \{\partial S_i'(t) \cup \partial S_i''(t)\}$ holds for the first time at $t > 0$. If multiple agents hold this correlation at the same time, we can select any one of them as the target. Here, each follower only chooses the moving agent as its target. In addition, note that it is impossible for two agents to pick each other as the target. Since it is the leader that begins to move first, Assumption 1 and this procedure ensure that at least one follower targets the leader, and each follower achieves LSC by remaining semi-connected with its target, and $\mathcal{G}(t)$ becomes a spanning tree.

Let $t = t_i$ be the time when agent i decided on a target.

3.3. Control Input

The control input to the follower $i \in \mathcal{F}$ is denoted by the following form:

$$\boldsymbol{u}_i(t) = u_{ir}(t)\boldsymbol{e}_{ir}(t) + u_{i\theta}(t)\boldsymbol{e}_{i\theta}(t), \tag{9}$$

where the target of agent i is agent j, $r_i(t) := \|\boldsymbol{x}_j^i(t)\|$ and $\boldsymbol{e}_{ir}(t) := \boldsymbol{x}_j^i(t)/r_i(t)$. $\boldsymbol{e}_{i\theta}(t)$ is a unit vector obtained by rotating $\boldsymbol{e}_{ir}(t)$ by $\pi/2$ counterclockwise (Fig. 3). In addition, we define $d_i(t) = \|\boldsymbol{x}_k^i(t)\|$ and the angle $\varphi_i(t) \in (-\pi, \pi]$ formed by $\boldsymbol{x}_j^i(t)$ and $\boldsymbol{x}_k^i(t)$ for the agent k with the smallest distance to the agent i other than the target (Fig. 4). $u_{ir}(t)$ and $u_{i\theta}(t)$ are determined as follows. Here, the control input is divided into three cases. The control input in case (I) is the control input before

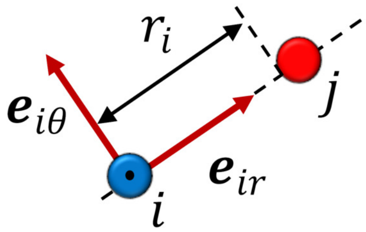


Figure 3. Local basis vectors $\boldsymbol{e}_{ir}$ and $\boldsymbol{e}_{i\theta}$ of agent i.

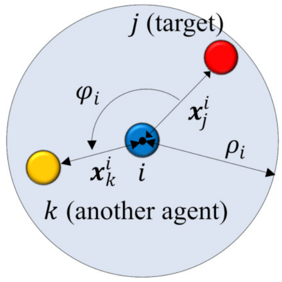


Figure 4. Agents' positions.

determining the target. The control input in case (II) is for maintaining connectivity during navigation. Finally, the control input in case (III) is for collision avoidance.

- Case (I) $t < t_i$:

$$u_{ir}(t) = 0, \qquad u_{i\theta}(t) = 0. \tag{10}$$

- Case (II) $t \geq t_i \cap d_i(t) := \|\boldsymbol{x}_k^i(t)\| > \rho_i''$:
  (a) When $\rho_i^{e-}(t) \leq r_i(t) := \|\boldsymbol{x}_j^i(t)\| \leq \rho_i''$:

$$u_{ir}(t) = \tfrac{U_i(r_i(t)-\rho_i'')}{\rho_i''-\rho_i'''}, \qquad u_{i\theta}(t) = 0. \tag{11}$$

  (b) When $\rho_i'' < r_i(t) \leq (\rho_i'' + \rho_i')/2$:

$$u_{ir}(t) = 0, \qquad u_{i\theta}(t) = \tfrac{2\sigma_i(t)U_i'(t)}{\rho_i'-\rho_i''}(r_i(t) - \rho_i''). \tag{12}$$

  (c) When $(\rho_i'' + \rho_i')/2 < r_i(t) < \rho_i'$:

$$u_{ir}(t) = 0, \qquad u_{i\theta}(t) = \tfrac{2\sigma_i(t)U_i'(t)}{\rho_i'-\rho_i''}(\rho_i' - r_i(t)). \tag{13}$$

  (d) When $\rho_i' \leq r_i(t) \leq \rho_i^c(t)$:

$$u_{ir}(t) = \tfrac{U_i(r_i(t)-\rho_i')}{\rho_i-\rho_i'}, \qquad u_{i\theta}(t) = \sigma_i(t)u_{ir}(t). \tag{14}$$

  (e) When $\rho_i^c(t) < r_i(t) \leq \rho_i^{e+}(t)$:

$$u_{ir}(t) = \tfrac{U_i(r_i(t)-\rho_i')}{\rho_i-\rho_i'}, \qquad u_{i\theta}(t) = \sigma_i(t)(U_i'(t) - u_{ir}(t)) \tag{15}$$

  where,

$$U_i'(t) := \max_{0\leq\tau\leq t}\|u_{ir}(\tau)\|, \tag{16}$$

$$\begin{cases} \rho_i^{e-}(t) := \rho_i'' - \frac{U_i'(t)}{U_i}(\rho_i'' - \rho_i'''), & \rho_i^c(t) := \rho_i' + \frac{U_i'(t)}{2U_i}(\rho_i - \rho_i'), \\ \rho_i^{e+}(t) := \rho_i' + \frac{U_i'(t)}{U_i}(\rho_i - \rho_i'). \end{cases} \tag{17}$$

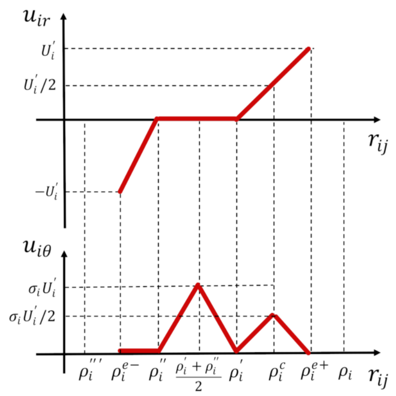


Figure 5. Control input for TM.

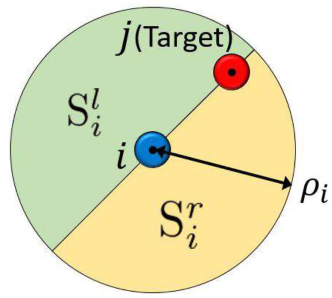


Figure 6. Definitions of S_i^r and S_i^l.

Here note that, when $r_i(t) \leq \rho_i''$, $r_i(t) \geq \rho_i^{e-}$ always holds from (11) and (16). Further, when $r_i(t) \geq \rho_i'$, $r_i(t) \leq \rho_i^{e+}$ always holds from (14), (15), and (16). Thus, $r_i(t)$ satisfies $\rho_i^{e-}(t) < r_i(t) < \rho_i^{e+}(t)$ as long as there exists a ij semi-connection, and it is sufficient to design the input in this range.

Here, $\sigma_i(t) \in [-1, 1]$ is a parameter to change the shape of the swarm. The larger the absolute value of $\sigma_i(t)$ becomes, the wider the shape of the swarm becomes. This value is changed to prevent the distance to other agents from approaching. The method for changing $\sigma_i(t)$ is described in Section 3.4.

Hereinafter, Case (I) and Case (II) are collectively called Tracking Mode (TM). Control input (10) is introduced for Assumption 1. The control input $u_{ir}(t)$ is designed to guarantee that followers maintain LSC. On the other hand, $u_{i\theta}(t)$ is designed to widen the swarm shape and to prevent it from becoming a chain structure. As a result, the follower behaves as follows in each case. In Case (II)–(a), the follower moves away from the target to avoid colliding with it. In Case (II)–(b) and (c), the follower moves away from the other agents to keep from approaching them. In Case (II)–(d) and (e), the follower moves away from the other agents while approaching the target to maintain connectivity. An outline of TM input is shown in Fig. 5.

- Case (III) $t \geq t_i \cap d_i(t) \leq \rho_i''$:

$$\boldsymbol{u}_i(t) = \boldsymbol{u}_i^j + \tfrac{t - t_i^{\text{start}}}{T_i}(\boldsymbol{u}_i^k(t) - \boldsymbol{u}_i^j), \tag{18}$$

where $T_i = h_i/(\zeta U_i)$, $h_i = \min\{\rho_i - \rho_i', \rho_i'' - \rho_i'''\}$, ζ is a positive constant that satisfies $2\pi(1 - \mathrm{e}^{-1/\zeta})/\zeta \leq 1$ ($\zeta = 2.253$ is used in this paper). Further, let t_i^{start} be the time when $d_i(t) = \rho_i''$, and agent k be the new target at $t = t_i^{\text{start}}$. Here note that if another agent comes into $S_i''(t)$, it becomes a new target. From the implementation points of view, it is difficult to switch targets instantly. Thus, the input is changed from the previous target reference to the new target reference over T_i. Redefine $r_i(t) = \|\boldsymbol{x}_k^i(t)\|$, $\boldsymbol{e}_{ir}(t) = \boldsymbol{x}_k^i(t)/r_i(t)$, and unit vector $\boldsymbol{e}_{i\theta}(t)$ by rotating $\boldsymbol{e}_{ir}(t)$ counterclockwise by $\pi/2$, and define each component of the input as follows:

$$\boldsymbol{u}_i^j := \boldsymbol{u}_i(t_i^{\text{start}}), \qquad \boldsymbol{u}_i^k(t) = u_{ir}(t)\boldsymbol{e}_{ir}, \tag{19}$$

where $u_{ir}(t)$ is defined in Case (II). Agent i uses the input (18) when $t_i^{\text{start}} \leq t < t_i^{\text{start}} + T_i$, returns to the TM at $t = t_i^{\text{start}} + T_i$, and redefines this time as $t = t_i$.

To summarize here, t_i is the time when follower i's target is decided or follower i's target-switching mode ends. t_i^{start} is, on the other hand, the time when follower i's target-switching mode starts. Below, Case (III) is called the Switching Target Mode (SM). Agents that could not be avoided by input $u_{i\theta}(t)$ in TM are set as new targets. By doing so, they avoid collision. Here, we introduce the assumption for SM.

Assumption 3 (Target Switching)**.** Regarding SM, three things are assumed. First, two or more agents do not start this mode at the same time; i.e., $t_i^{\text{start}} \neq t_j^{\text{start}}$ if $i \neq j$. Second, agent i's new target is not a descendant of itself. Third, no other agents enter $S_i''(t) \setminus \partial S_i''(t)$ during this mode.

With this assumption, LSC is maintained even after the SM is executed.

3.4. Changing $\sigma_i(t)$ Method

Now, we propose an algorithm that changes $\sigma_i(t)$. Here, note that the method of changing $\sigma_i(t)$ is not unique, as $\sigma_i(t)$ can be changed to satisfy the limitation $\sigma_i(t) \in [-1, 1]$. In particular, if $\sigma_i(t) \in [-1, 1]$ holds, the proofs in Section 4 are satisfied.

Since $\sigma_i(t)$ is not included in the input of the SM, we set $\sigma_i(t_i) = 0$. In the TM, on the other hand, $\sigma_i(t)$ can be changed. To avoid fast changes of $\sigma_i(t)$, we make the following restriction:

$$\alpha_i < \dot{\sigma}_i(t) < \beta_i, \tag{20}$$

where lower limit α_i is a negative constant and upper limit β_i is a positive constant. In our algorithm, when three or more followers are targeting the same agent, the distance between them is almost equal, and thus the collisions between them are avoided.

Here, $S_i^r(t)$ and $S_i^l(t)$ are defined as follows (Fig. 6):

$$\begin{cases} S_i^r(t) := \{\boldsymbol{X} \in D \mid \|\boldsymbol{x}_i(t) - \boldsymbol{X}\| \leq \rho_i \,, \varphi_i(t) > 0\} \,, \\ S_i^l(t) := \{\boldsymbol{X} \in D \mid \|\boldsymbol{x}_i(t) - \boldsymbol{X}\| \leq \rho_i \,, \varphi_i(t) < 0\} \,, \end{cases} \tag{21}$$

where $\varphi_i(t) \in (-\pi, \pi]$ is the angle formed by $\boldsymbol{x}_j^i(t)$ and $\boldsymbol{x}_k^i(t)$ for the agent k with the smallest distance to the agent i other than the target (Fig. 4).

Based on the above, we define the event $R_i(t)$ as follows using the manner of logical operation: $R_i(t) = R_i^1(t)$ if $t = t_i$, and $R_i^2(t) \vee R_i^3(t)$ otherwise. Here, $R_i^1(t_i) = 1$ if the target is in the region $S_i^r(t_i)$ at time $t = t_i$, and 0 otherwise. $R_i^2(\tau) = 1$ if the closest agent enters region $S_i^r(\tau)$ from outside of region $S_i(\tau)$ at time $t = \tau > t_i$, and 0 otherwise. $R_i^3(\tau) = 1$ if the closest agent changes from the agent in $S_i^l(\tau)$ to the agent in $S_i^r(\tau)$ at time $t = \tau > t_i$, and 0 otherwise. Similarly, we define a symmetrical event $L_i(t)$; that is, for $L_i(t)$ we replace S_i^r with S_i^l and vice versa. We count, at time t, the number of times $R_i(t) = 1$ is satisfied in $t_i \leq \tau < t$ and make this number $n_i^r(t)$. The same applies to $L_i(t)$ and the number becomes $n_i^l(t)$. Here, $n_i^r(t)$ is, in brief, the number of times other agents approached from the right with respect to (w.r.t.) the target.

The desired value of $\sigma_i(t)$, $\sigma_i^{\text{des}}(t) = \tan(\theta_i^{\text{des}}(n_i^r, n_i^l))$, is defined as follows:

1. $n_i^r + n_i^l \leq 1$:

$$\theta_i^{\rm des}(0,0) = 0, \quad \theta_i^{\rm des}(1,0) = \pi/4, \quad \theta_i^{\rm des}(0,1) = -\pi/4. \tag{22}$$

2. $n_i^r + n_i^l \geq 2 \cap \theta_i^{\rm des}(n_i^r, n_i^l) < 0$:

$$\theta_i^{\rm des}(n_i^r+1, n_i^l) = -\frac{1}{2}\theta_i^{\rm des}(n_i^r, n_i^l), \quad \theta_i^{\rm des}(n_i^r, n_i^l+1) = \frac{1}{2}\left[-\frac{\pi}{4} + \theta_i^{\rm des}(n_i^r, n_i^l)\right]. \tag{23}$$

3. $n_i^r + n_i^l \geq 2 \cap \theta_i^{\rm des}(n_i^r, n_i^l) > 0$:

$$\theta_i^{\rm des}(n_i^r + 1, n_i^l) = \frac{1}{2}\left[\frac{\pi}{4} + \theta_i^{\rm des}(n_i^r, n_i^l)\right], \quad \theta_i^{\rm des}(n_i^r, n_i^l + 1) = -\frac{1}{2}\theta_i^{\rm des}(n_i^r, n_i^l). \tag{24}$$

Set $\sigma_i(t_i) = \sigma_i^{\rm des}(t_i) = n_i^r(t_i) = n_i^l(t_i) = 0$ and update $\theta_i^{\rm des}(t)$ and $\sigma_i^{\rm des}(t)$. Then we make $\sigma_i(t)$ follow $\sigma_i^{\rm des}(t)$ satisfying a constraint of $\alpha_i < \dot{\sigma}_i(t) < \beta_i$. There are numerous ways to realize this, but we adopt the discrete method from the implementation point of view. In particular, we calculate this value as follows: $\sigma_i(t) = \sigma_i(t - \Delta t) + \alpha_i \Delta t$ if $\sigma_i(t) > \sigma_i^{\rm des}(t)$, $\sigma_i(t) = \sigma_i(t - \Delta t) + \beta_i \Delta t$ if $\sigma_i(t) < \sigma_i^{\rm des}(t)$, and $\sigma_i(t) = \sigma_i^{\rm des}(t)$ otherwise. By changing $\sigma_i(t)$ in this way, the swarm shape can be expanded. Especially when three or more followers are targeting the same agent, the distance between them is almost equal.

3.5. Summarizing the Proposed Method

Here, we summarize the proposed control method.

The leader moves in D satisfying the constraint (8) from Assumption 2.

The followers determine their targets using the target determination method in Section 3.2. In addition, followers move in D using the following control inputs:

- Case (I): (9) and (10).
- Case (II): (9) and (11)–(15), where $\sigma_i(t)$ is determined by the method in Section 3.4.
- Case (III): (18).

4. Guarantees by the Proposed Method

Now we prove that LSC is maintained, the followers' velocity constraint (2) is satisfied, and each follower avoids colliding with the others. Assumptions 1, 2, and 3 are satisfied, and all followers determine the target according to the method in Section 3.2, and then move according to the control input in Section 3.3. Then, at any time $t \geq t_i$, the following lemmas and theorem are satisfied.

In the following, we consider agent i and define its target as agent j and the target after SM as agent k. Further, the argument (t) is omitted unless specifically emphasized. The inputs of agents j and k are decomposed into the r direction and θ direction w.r.t. agent i as

$$\boldsymbol{u}_j = u_{jr}\boldsymbol{e}_{ir} + u_{j\theta}\boldsymbol{e}_{i\theta}, \qquad \boldsymbol{u}_k = u_{kr}\boldsymbol{e}_{ir} + u_{k\theta}\boldsymbol{e}_{i\theta}. \tag{25}$$

Here, note that $\boldsymbol{u}_j$ and $\boldsymbol{u}_k$ are represented using the local basis vectors $\boldsymbol{e}_{ir}$ and $\boldsymbol{e}_{i\theta}$ of agent i. Further, we assume that

$$\|\boldsymbol{u}_j\| \le U_{n+1}, \qquad \|\boldsymbol{u}_k\| \le U_{n+1}. \tag{26}$$

We show that this assumption always holds after Lemma 4.4.

First, we consider TM.

Lemma 4.1 (Connectivity in TM)**.** *Under* (26), $\rho_i''' < r_i < \rho_i$ *holds for agent* i *in TM.*

Proof. First, we prove the lower bound. When we consider $r_i \le \rho_i''$, the following equation is obtained from the control law (11):

$$\dot{r}_i = u_{jr} - u_{ir} = u_{jr} - \tfrac{U_i}{\rho_i''-\rho_i'''}(r_i - \rho_i''). \tag{27}$$

We set $t = t_i'$ for the first time when $r_i \le \rho_i''$ in $t \ge t_i$. Solving (27) with $r_i(t_i') = \rho_i''$ as the initial condition, we obtain the following estimation:

$$r_i = \rho_i'' + \int_{t_i'}^{t} \mathrm{e}^{\frac{U_i}{\rho_i''-\rho_i'''}(\tau-t)} u_{jr}(\tau)\mathrm{d}\tau > \rho_i'' - \tfrac{U_{n+1}}{U_i}(\rho_i'' - \rho_i''') \ge \rho_i'''. \tag{28}$$

Here, we used $u_{jr} \le U_{n+1}$, derived from (26), and (8). In addition, when $r_i > \rho_i''$ holds, $r_i > \rho_i'''$ holds. Therefore, $r_i > \rho_i'''$ always holds.

Next, we consider the upper bound. When $r_i \ge \rho_i'$, we obtain

$$\dot{r}_i = u_{jr} - \tfrac{U_i}{\rho_i-\rho_i'}(r_i - \rho_i') \tag{29}$$

from the control laws (14) and (15). Here, we set $t = t_i'$ for the first time when $r_i \ge \rho_i'$ in $t \ge t_i$. Solving (29) with the initial condition $r_i(t_i') = \rho_i'$ and using $u_{jr} \le U_{n+1}$ and (8), we obtain

$$r_i < \rho_i' + \tfrac{U_{n+1}}{U_i}(\rho_i - \rho_i') \le \rho_i' + \rho_i - \rho_i' = \rho_i. \tag{30}$$

When $r_i < \rho_i'$ holds, $r_i < \rho_i$ holds. Thus, the upper bound of r_i is ρ_i. □

Lemma 4.2 (Velocity Constraint in TM)**.** *Under* (26), $\|\boldsymbol{u}_i\| < U_{n+1}$ *holds for agent* i *in TM.*

Proof. This is proved separately for each control input.

- Case (I): $\|\boldsymbol{u}_i\| = 0 < U_{n+1}$ holds from (10).
- Case (II)-(a): We have $|u_{ir}| < U_{n+1}$ because of (11) and $r_i > \rho_i'' - U_{n+1}(\rho_i'' - \rho_i''')/U_i$ from (28). Thus, $\|\boldsymbol{u}_i\| = |u_{ir}| < U_{n+1}$ holds.
- Case (II)-(d): $|u_{ir}| < U_i'/2$ holds from (14) and ρ_i^c, and $U_i' < U_{n+1}$ holds from (30) and (14). Thus, we obtain $|u_{ir}| < U_{n+1}/2$. Therefore, the following inequality holds: $\|\boldsymbol{u}_i\| = \sqrt{{u_{ir}}^2 + {u_{i\theta}}^2} < U_{n+1}/\sqrt{2} < U_{n+1}$. Here note that $|\sigma_i| \le 1$ holds.
- Case (II)-(e): From (15), $U'/2 < |u_{ir}| \le U_i'$, and therefore the following inequality holds: $\|\boldsymbol{u}_i\|^2 = u_{ir}^2 + \sigma_i^2(U_i' - u_{ir})^2 \le U_i'^2$ because of $|\sigma_i| \le 1$. Hence, $U_i' < U_{n+1}$ from (30) and (15). So, $\|\boldsymbol{u}_i\| \le U_i' < U_{n+1}$ holds.

- Case (II)-(b),(c): From (12) and (13), $\|\boldsymbol{u}_i\| = |u_{i\theta}| \leq |\sigma_i U_i'| < U_{n+1}$ holds, because $|\sigma_i| \leq 1$ holds. Here, we used $U_i' < U_{n+1}$ because $|u_{ir}| < U_{n+1}$ holds in all cases.

From the above, $\|\boldsymbol{u}_i\| < U_{n+1}$ always holds. $\square$

Second, we consider SM. Here, the inputs of agent i, $\boldsymbol{u}_i^j$ and $\boldsymbol{u}_i^k$ are decomposed into the r direction and θ direction w.r.t. agent i as follows:

$$\boldsymbol{u}_i^j = (\boldsymbol{u}_i^j \cdot \boldsymbol{e}_{ir})\boldsymbol{e}_{ir} + (\boldsymbol{u}_i^j \cdot \boldsymbol{e}_{i\theta})\boldsymbol{e}_{i\theta}, \qquad \boldsymbol{u}_i^k(t) = u_{ir}\boldsymbol{e}_{ir}. \tag{31}$$

Note that $\boldsymbol{u}_i^j$ is not a function of t (see (19)). We set

$$\delta_i = \frac{t - t_i^{\text{start}}}{T_i}. \tag{32}$$

Lemma 4.3 (Connectivity in SM)**.** *Under* (26)*,* $\rho_i''' < r_i < \rho_i$ *holds for agent* i *in SM.*

Proof. We will prove the lower bound. From (18), as long as $r_i \leq \rho_i''$, r_i follows $\dot{r}_i = u_{kr} - u_{ir} = u_{kr} - [\boldsymbol{u}_i^j \cdot \boldsymbol{e}_{ir} + \delta_i(u_{ir} - \boldsymbol{u}_i^j \cdot \boldsymbol{e}_{ir})]$, that is, $\dot{r}_i + \delta_i U_i(r_i - \rho_i'')/(\rho_i'' - \rho_i''') = u_{kr} - (1-\delta_i)\,(\boldsymbol{u}_i^j \cdot \boldsymbol{e}_{ir})$, where we used (11). Solving this differential equation under the initial condition $r_i(t_i^{\text{start}}) = \rho_i''$ and using $0 \leq 1 - \delta_i \leq 1$ and (26), the following estimation holds:

$$\begin{aligned} r_i &= \rho_i'' + \mathrm{e}^{-\Gamma_i t^2} \int_0^t \left[u_{kr} - \left(1 - \tfrac{\tau}{T_i}\right) (\boldsymbol{u}_i^j \cdot \boldsymbol{e}_{ir}) \right] \mathrm{e}^{\Gamma_i \tau^2} \mathrm{d}\tau \\ &\geq \rho_i'' - 2U_{n+1} \mathrm{e}^{-\Gamma_i t^2} \int_0^t \mathrm{e}^{\Gamma_i \tau^2} \mathrm{d}\tau, \end{aligned} \tag{33}$$

where $\Gamma_i = U_i/\{2T_i(\rho_i'' - \rho_i''')\}$ and we set t as $t - t_i^{\text{start}}$. Here, set $I = \int_0^t \mathrm{e}^{\Gamma_i \tau^2} \mathrm{d}\tau = \int_{-t}^0 \mathrm{e}^{\Gamma_i \tau^2} \mathrm{d}\tau$ and the following inequality holds: $(2I)^2 = \int_{-t}^t \int_{-t}^t \mathrm{e}^{x^2+y^2} \mathrm{d}x\mathrm{d}y < \int_0^{\sqrt{2}t} \int_0^{2\pi} \mathrm{e}^{\Gamma_i r^2} r \mathrm{d}\theta \mathrm{d}r$, that is, $I < \{\pi(\mathrm{e}^{2\Gamma_i t^2} - 1)/\Gamma_i\}^{1/2}/2$. Therefore, from the definition of ζ, that satisfies $2\pi(1 - \mathrm{e}^{-1/\zeta})/\zeta \leq 1$, and $T_i = h_i/(\zeta U_i)$,

$$r_i \geq \rho_i'' - 2U_{n+1} \mathrm{e}^{-\Gamma_i t^2} I \geq \rho_i'' - U_{n+1}(\rho_i'' - \rho_i''')/U_i > \rho_i'''. \tag{34}$$

Note that even if $r_i > \rho_i''$ holds while $t_i^{\text{start}} < t < t_i^{\text{start}} + T_i$, the bound can be proved in the same way by changing the time when $r_i = \rho_i''$ holds again to $t = t_i^{\text{start}}$. On the other hand, using the same approach, it is easy to prove the upper bound of r_i. $\square$

Lemma 4.4 (Velocity Constraint in SM)**.** *Under* (26)*,* $\|\boldsymbol{u}_i\| < U_{n+1}$ *holds for agent* i *in SM.*

Proof. From Lemma 4.3 and (16), we can show that $|u_{ir}| \leq U_i' < U_{n+1}$ holds in SM as shown in proof of Lemma 4.2. Thus, from (18), (26), $0 \leq \delta_i \leq 1$ and $|u_{ir}| < U_{n+1}$, we obtain the following:

$$\|\boldsymbol{u}_i\|^2 = (1-\delta_i)^2 \|\boldsymbol{u}_i^j\|^2 + \delta_i^2 u_{ir}^2 + 2\delta_i (1-\delta_i)(\boldsymbol{u}_i^j \cdot \boldsymbol{e}_{ir}) u_{ir} < U_{n+1}^2. \tag{35}$$

Therefore, the velocity constraint is satisfied in SM. $\square$

Note that all constraints are satisfied from the end of the SM until $r_i = \rho_i''$ or $r_i = \rho_i'$ for the first time. The connectivity maintenance can be proved by solving (27) under the initial condition $r_i(t = t_i^{\mathrm{start}} + T_i) = r_0(\rho_i^{e-} \le r_0 \le \rho_i'')$, and the velocity constraint can be proved in the same way.

Now we show (26). From Assumption 1, the leader $n+1$ begins to move first, while other agents are stationary. Next, from the target determination in Section 3.2, at least one follower targets the leader. Now, let us consider the case where the target of agent i is the leader and agent i is in TM. At this time, $\|u_j\| \le U_{n+1}$ holds because agent j is the leader. Therefore, from Lemma 4.2, $\|u_i\| \le U_{n+1}$ holds. As agent i moves to follow j, an agent l sets i or $n+1$ as its target l'. Since $\|u_i\| \le U_{n+1}$, we can apply the same argument for j and i to l' and l. Further, we can consider any pair of such agents m' and m in the same way. Then, (26) holds for all agents. In addition, using Lemma 4.4 and the same procedure, (26) holds even if any agents are in SM.

Theorem 4.5 (LSC and constraints of whole swarm)**.** *All followers satisfy the velocity constraints* (2) *and do not collide with other agents, and the swarm forms a spanning tree.*

Proof. From Assumption 1 and the target determination in Section 3.2, the leader begins to move first, while other agents are stationary. Next, from the target determination in Section 3.2, at least one follower i, whose target is the leader moves. In the same way, follower j, whose target is the leader or the follower i, moves. Repeating these steps, and from Lemmas 4.1 and 4.3 and Assumption 3, we can prove that the swarm always forms a spanning tree and that no collisions occur between agents. That is, from Lemma 4.1, all followers have a certain distance to the target, and thus do not collide. Now let us consider follower i. When an agent k other than the target of follower i enters the area $S_i''(t)$, the agent k becomes the target of the follower i because of SM. Here, agent k is not a descendant of itself, so the LSC is maintained from Assumption 3. Furthermore, from Lemma 4.3, the distance between follower i and agent k is more than a certain distance, so no collision occurs. From this, collision avoidance is achieved.

On the other hand, from Lemmas 4.2 and 4.4, the velocity constraints of the followers are satisfied. Therefore, at any time $t > 0$, all followers satisfy the velocity constraint and, do not collide with other agents, and the swarm forms a spanning tree. $\square$

5. Simulation Results

In the simulation, we confirm that agents do not collide in the proposed method even in the motion by which agents do collide in the conventional method. Here, we define the specifications of all followers as shown in Table 1.

Now, we compare the prior method [17] and the proposed method under the same conditions. The prior method does not have SM input (18) and did not consider collision avoidance between agents. In addition, although $\sigma_i(t)$ was used in the prior method, the method of changing $\sigma_i(t)$ was not described. Therefore, in this simulation, we used the same change method of $\sigma_i(t)$ as the proposed method in the prior method. The two methods are compared in Fig. 7. The figures in the upper row show the simulation of the conventional method, and those in the lower row show that of the proposed method. The blue agent is the leader, and the lines between agents describe

Table 1. Specifications of followers in the simulation.

Agent i	1	2	3	4	5	6	7–10
ρ_i (m)	5.00	4.00	6.00	4.50	3.50	3.00	5.00
ρ'_i (m)	4.65	3.55	5.45	3.85	3.15	2.57	4.45
ρ''_i (m)	0.55	0.75	0.85	1.00	0.65	0.65	0.85
ρ'''_i (m)	0.20	0.30	0.30	0.35	0.30	0.22	0.30
ρ_i^{size} (m)	0.20	0.30	0.30	0.35	0.30	0.10	0.20
U_i (m/s)	0.90	0.60	0.50	0.30	0.70	0.20	0.20
$-\alpha_i,\ \ \beta_i$	1.00	1.00	1.00	1.00	1.00	1.00	1.00

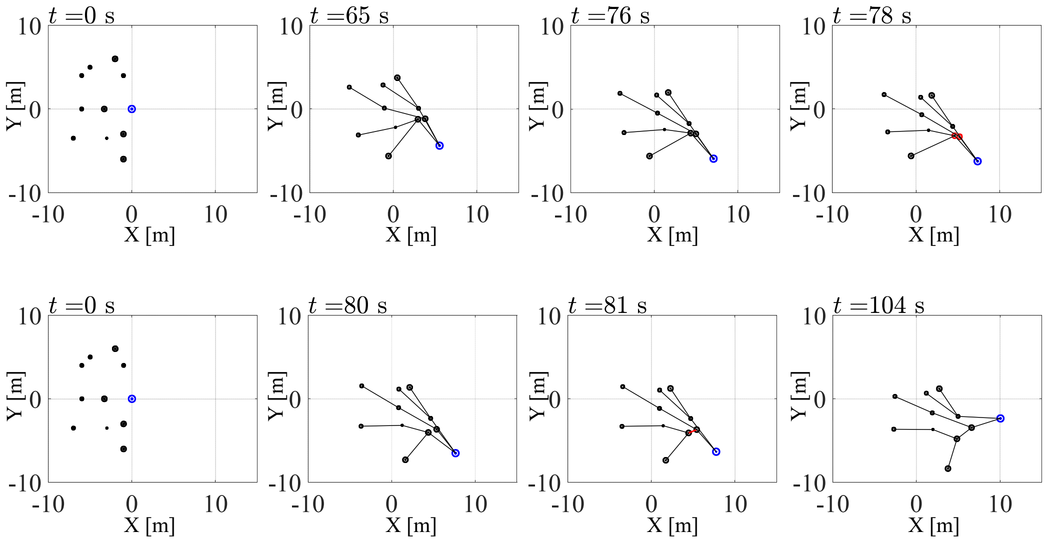
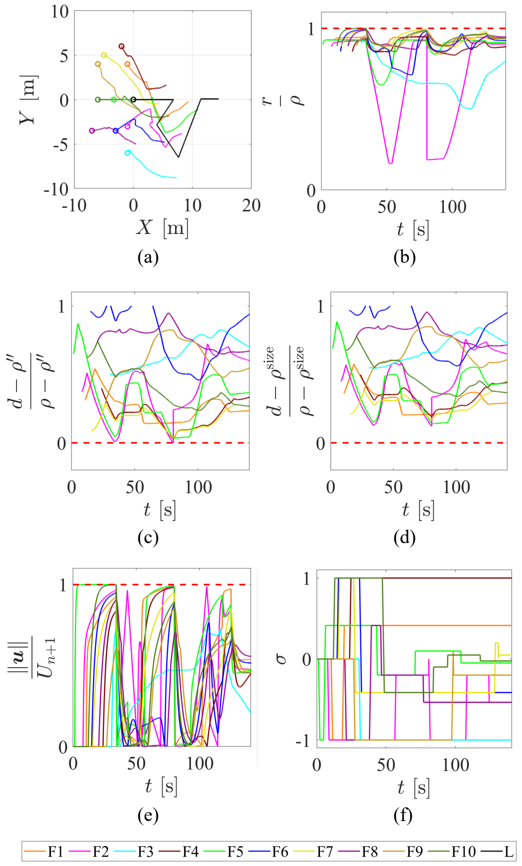


Figure 7. Screenshots of the simulation.

the connectivity. The line when the connectivity is changed by the SM is shown in red. Agents that collide with other agents are shown in red. The sampling period is set to 0.01s.

In the prior method, there is a collision between agents at time $t = 78$s, but in the proposed method, the collision is avoided by SM input (18) at time $t = 81$s.

In addition, Fig. 8 and Fig. 9 show the simulation results of the proposed method and the prior method, respectively. In both figures, (a) shows the paths of all agents, (b) shows the distance to the target, (c) shows the difference between ρ''_i and minimum distance to agents other than the target, (d) shows the difference between ρ_i^{size} and minimum distance to agents other than the target, (e) shows the norm of velocity, and (f) shows the σ. First, we consider the simulation results of the proposed method (Fig. 8). Since the minimum distance to agents is not less than ρ_i^{size} from (d), no collision between agents has occurred. In particular, from (c), the minimum distance to agents becomes smaller than ρ''_i at around $t = 81$s, and SM works to realize collision avoidance. Here, note that the value in (c) is not smaller than ρ''_i at around $t = 34$s. We also see that connectivity is maintained, and velocity constraints are satisfied from (b) and (e). Moreover, we can see in (f) that $-1 \leq \sigma \leq 1$ is always satisfied. On the other hand, in the prior method, since the minimum distance to agents is less than ρ_i^{size} at $t = 78$s from Fig. 9 (d), the collision between agents has occurred. These results show that, by the proposed method, all agents maintain semi-connection with the target, no agent collides with any other, and no velocity constraints are exceeded.



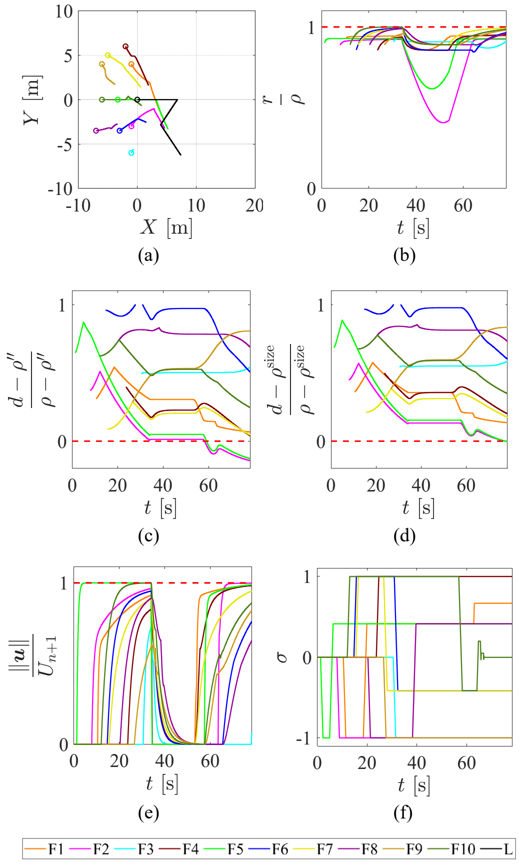


Figure 8. Simulation results by the proposed method. (a) shows the paths of the agents. Here, the circles show the initial positions. (b) shows the distances from the targets. (c) shows the differences between ρ_i'' and minimum distance from agents. (d) shows the differences between ρ_i^{size} and minimum distance from agents. (e) shows the norm of velocity. (f) shows σ.

Figure 9. Simulation results by the prior method. (a) shows the paths of the agents. Here, the circles show the initial positions. (b) shows the distances from the targets. (c) shows the differences between ρ_i'' and minimum distance from agents. (d) shows the differences between ρ_i^{size} and minimum distance from agents. (e) shows the norm of velocity. (f) shows σ.

Table 2. Specifications of followers in the experiment.

Agent i	1	2	3	4
ρ_i (m)	1.30	1.30	1.30	1.30
ρ'_i (m)	1.00	1.00	1.00	1.00
ρ''_i (m)	0.52	0.59	0.66	0.68
ρ'''_i (m)	0.29	0.32	0.34	0.36
ρ_i^{size} (m)	0.09	0.09	0.09	0.09
U_i (m/s)	0.25	0.25	0.25	0.25
$-\alpha_i,\ \beta_i$	1.00	1.00	1.00	1.00

6. Experimental Results

To confirm the effectiveness of the proposed method in a real environment, we carried out an experiment. In the experiment, we used one leader robot and four follower robots. All of these robots were omni-directional driven with three omni wheels. Since a robot needs the coordinates of other robots within its own sensing region, six AR markers were attached to each robot. The leader robot was controlled by velocity command via Bluetooth, and the operator sent the velocity commands.

On the other hand, each follower robot had three cameras to achieve a 360-degree sensing region. These cameras read the AR marker of the robot within the sensing region to obtain the robot's relative coordinates. In addition, a computer was built into each follower robot. Each follower was controlled based on the calculation of its own control input by this computer. The sampling frequency was 10Hz. The specifications of all followers are shown in Table 2.

In the experiment, we observe the motions of the four followers when the leader robot meanders so that the followers collide with each other. Fig. 10 shows screenshots of this experiment. In this figure, L is the leader, and F1, F2, F3, and F4 are the followers. White solid lines between agents describe the connectivity. Around $t = 28$s, F1 and F3 got too close to each other. For this reason, F3 changed the target from leader to F1. The red line represents the new connectivity and the white dotted line represents the old connectivity. This switching of targets prevented agents from colliding. After that, the followers maintained connectivity and followed the leader.

The experimental results are shown in Fig. 11. In Fig. 11 (c) and (d), when the minimum distance d to agents other than the target is $d > \rho$, the distance is not acquired in the implementation. Therefore, when $d > \rho$, the response was created with $d = \rho$. From these results, we found that all robots in the proposed method maintained connectivity with the target, none collided with any others, and none exceeded the velocity constraints. In particular, from (c), the L3's minimum distance to agents other than the target becomes smaller than ρ''_i at around $t = 28$s. In that time, F3 changed the target by SM and realized collision avoidance. Thus, the proposed method works effectively even in real situations. In Fig. 11 (c) and (d), there are the parts where the response oscillates. This is because the AR marker to be measured was switched when the camera measured the distance. Therefore, the proposed method works well even in the presence of measurement noise.

7. Conclusion

In this study, we proposed a control method in which a single leader guide a robotic swarm whose agents have various individual abilities and do not collide with each other.

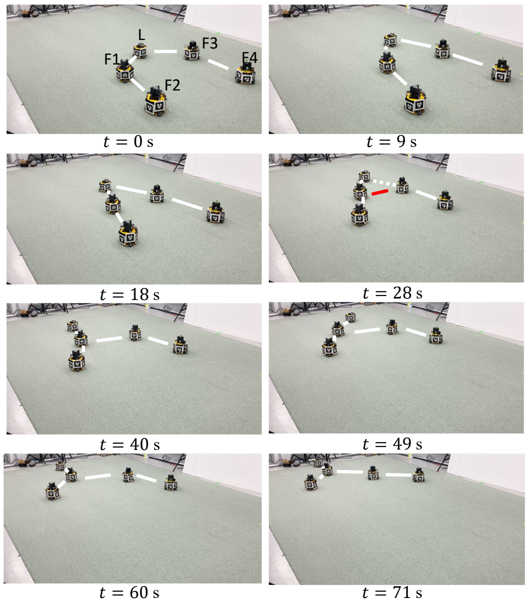


Figure 10. Screenshots of the experiment.

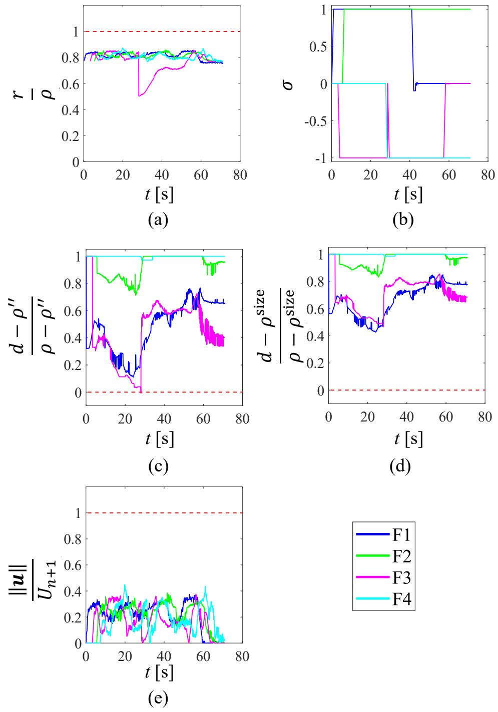


Figure 11. Experimental results. (a) shows the distances from the targets. (b) shows σ. (c) shows the differences between ρ_i'' and minimum distance from agents. (d) shows the differences between ρ_i^{size} and minimum distance from agents. (e) shows the norm of velocity.

In the proposed method, each follower chooses a target with which to maintain a connection and switches the target appropriately. The proposed method is distributed in the sense that each agent is controlled using only local information. Moreover, the effectiveness of the proposed method was confirmed by both simulation and experiment. Future issues include improving the θ direction input and making the topology more flexible to guarantee collision avoidance even in situations where it cannot be guaranteed at present, and to deal with situations in which one agent fails and loses the line of sight to its target, thus partitioning the network.

Acknowledgement

This work was supported in part by JST SICORP Grant Number JPMJSC18E4, Japan.

References

[1] Brambilla M, Ferrante E, Birattari M, et al. Swarm robotics: a review from the swarm engineering perspective. Swarm Intelligence. 2013;7:1–41.
[2] Bayindir L. A review of swarm robotics tasks. Neurocomputing. 2016;172:292–321.

[3] Nedjah N, Junior LS. Review of methodologies and tasks in swarm robotics towards standardization. Swarm and Evolutionary Computation. 2019;50:100565.
[4] Kantaros Y, Thanou M, Tzes A. Distributed coverage control for concave areas by a heterogeneous robot-swarm with visibility sensing constraints. Automatica. 2015;53:195–207.
[5] Garcia-Aunon P, Roldán JJ, Barrientos A. Monitoring traffic in future cities with aerial swarms: Developing and optimizing a behavior-based surveillance algorithm. Cognitive Systems Research. 2019;54:273–286.
[6] Farrugia JL, Fabri SG. Swarm robotics for object transportation. In: 2018 UKACC 12th International Conference on Control (CONTROL); 2018. p. 353–358.
[7] Olfati-Saber R. Flocking for multi-agent dynamic systems: Algorithms and theory. IEEE Transactions on Automatic Control. 2006;51(3):401–420.
[8] Li X, Sun D, Yang J. A bounded controller for multirobot navigation while maintaining network connectivity in the presence of obstacles. Automatica. 2013;49(1):285–292.
[9] Kan Z, Klotz JR, Shea J, et al. Decentralized rendezvous of nonholonomic robots with sensing and connectivity constraints. ASME Journal of Dynamic Systems, Measurement, and Control. 2017;139(2):024501.
[10] Gennaro MD, Jadbabaie A. Decentralized control of connectivity for multi-agent systems. In: the 45th IEEE Conference on Decision and Control; 2006. p. 3628–3633.
[11] Ji M, Egerstedt M. Distributed coordination control of multiagent systems while preserving connectedness. IEEE Transactions on Robotics. 2007;23(4):693–703.
[12] Zavlanos MM, Pappas GJ. Distributed connectivity control of mobile networks. IEEE Transactions on Robotics. 2008;24(6):1416–1428.
[13] Zavlanos MM, Egerstedt MB, Pappas GJ. Graph-theoretic connectivity control of mobile robot networks. Proceedings of the IEEE. 2011;99(9):1525–1540.
[14] Gasparri A, Sabattini L, Ulivi G. Bounded control law for global connectivity maintenance in cooperative multirobot systems. IEEE Transactions on Robotics. 2017;33(3):700–717.
[15] Dorigo M, Floreano D, Grambardella L, et al. Swarmanoid: A novel concept for the study of heterogeneous robotic swarm. IEEE Robotics Automation Magazine. 2013;20(4):60–71.
[16] Chand P, Carnegie D. Mapping and exploration in a hierarchical heterogeneous multi-robot system using limited capability robots. Robotics and Autonomous Systems. 2013; 61(6):565–579.
[17] Yoshimoto M, Endo T, Maeda R, et al. Decentralized navigation method for a robotic swarm with nonhomogeneous abilities. Autonomous Robots. 2018;42(8):1583–1599.
[18] Maeda R, Endo T, Matsuno F. Decentralized navigation for heterogeneous swarm robots with limited field of view. IEEE Robotics and Automation Letters. 2017;2(2):904–911.
[19] Ávila Martínez E, Barajas-Ramírez J. Flocking motion in swarms with limited sensing radius and heterogeneous input constraints. Journal of the Franklin Institute. 2021; 358(4):2346–2366.
[20] Sabattini L, Secchi C, Chopra N. Decentralized estimation and control for preserving the strong connectivity of directed graphs. IEEE Transactions on Cybernetics. 2014; 45(10):2273–2286.
[21] Filotheou A, Nikou A, Dimarogonas D. Robust decentralised navigation of multi-agent systems with collision avoidance and connectivity maintenance using model predictive controllers. International Journal of Control. 2020;93(6):1470–1484.
[22] Park BS, Yoo SJ. Connectivity-maintaining and collision-avoiding performance function approach for robust leader–follower formation control of multiple uncertain underactuated surface vessels. Automatica. 2021;127:109501.
[23] Guo Y, Zhou J, Liu Y. Distributed lyapunov-based model predictive control for collision avoidance of multi-agent formation. IET Control Theory & Applications. 2018; 12(18):2569–2577.
[24] Sabattini L, Chopra N, Secchi C. Decentralized connectivity maintenance for cooperative control of mobile robotic systems. The International Journal of Robotics Research. 2013; 32(12):1411–1423.

[25] Sakai D, Fukushima H, Matsuno F. Leader–follower navigation in obstacle environments while preserving connectivity without data transmission. IEEE Transactions on Control Systems Technology. 2017;26(4):1233–1248.

[26] Francis B, Maggiore M. Flocking and rendezvous in distributed robotics. Springer; 2016.