Distributed Formation for Robotic Swarms Considering Their Crossing Motion

Yuusuke Kobayashi, Takahiro Endo, and Fumitoshi Matsuno
Department of Mechanical Engineering and Science, Kyoto University,
Kyodai-katsura, Nishikyo-ku, Kyoto, Japan

Abstract

This paper discusses formation problem for robotic swarms when multiple robotic swarms cross one another's path. To realize the crossing motion, collision avoidance between agents is an important issue, with the potential to cause a general mix-up of formation during the crossing motion. To realize an orderly and well-organized crossing motion with the least mix-up, as well as collision avoidance, we propose a distributed controller. This well-organized crossing motion can realize visually appealing and highly entertaining robotic mass games. This paper proposes a distributed controller using the gradient of the cost functions about the formation maintenance, collision avoidance, and tracking to the desired trajectory. We then prove that we can achieve a well-organized crossing motion of multiple robotic swarms under several assumptions. Finally, experimental and numerical simulations are carried out to investigate whether the well-organized crossing motion can be achieved.

Keywords: Robotic swarms; Formation Control; Well-Organized Crossing; Multi-Vehicle Systems.

1. Introduction

Recently, the need for cooperative work by multi-robot systems has been highlighted in various fields, including cooperative transportation, and environmental exploration and monitoring [1-3]. In particular, robotic swarms with multi-robot systems have potential advantages such as robustness for some faults, flexibility for various environments or tasks, and scalability in terms of number of robots, compared to single-robot systems [4-7]. In terms of controlling robotic swarms, including distributed robotics, many of the major problems have already been solved, including flocking, consensus, and the rendezvous problem (see, for example, [8-14]).

On the other hand, studies have been carried out to realize visually amusing motion, like mass games, by controlling the formation of robotic swarms. A mass game is one in which many individuals act as a group, for example, dancing or doing gymnastics while maintaining population synchrony. Mass games are often performed at school and international athletic events to indicate group solidarity. Izumi *et al*. proposed distributed controllers for mass games with multi-robots by combining a coverage control method with halftone image processing, and generated formations of

agents displaying specified gray-scale images [15]. To display images and animation of a group of mobile robots, Alonso-Mora *et al*. proposed a display method that generated the desired robot positions and robot assignments to goal positions, with a distributed collision-avoidance algorithm that determined a collision-free velocity for each robot [16]. Yamane *et al*. realized the display of stick-figure animations using a centralized motion-planning algorithm [17]. Further, Rubenstein *et al*. replicated desired forms by creating a thousand real robotic swarms [18].

These studies realized visually amusing motion by displaying forms or patterns created by a group of robots. However, in a human mass game, we see not only the motion that forms a character or pattern but also crossing motions between groups without collisions. To realize such motion, including crossing, we need to consider situations in which multiple robotic swarms pass each other. van den Berg *et al*. realized crossing motion without collision with multiple robots using the mutual collision avoidance algorithm called Reciprocal Velocity Obstacles (RVO), which is based on velocity obstacles [19, 20]. In addition, this method was extended to three-dimensional (3-D) space, and applied to three unmanned aerial vehicles (UAVs) [21]. To achieve visually pleasing group-formation transition in the formation transform of multi-agents, Xu *et al*. proposed a control method based on the social-force model, with pair assignments and mutual information [22].

In the crossing motion seen in mass games, it is thought that the absence of formation mix-ups and the stationary postures in transient states are important issues for visual appeal. However, to the best of our knowledge, no research has considered such transient states in the crossing of robotic swarms. For example, previous studies [15, 16, 18] did not explicitly treat form configuration in crossing, or the transform of the formation. Thus, these controllers are not appropriate for performing motions that include crossing.

There have been many studies about the formation control of multi-agents forming one group [23-26]. One of the problems in formation control is collision avoidance between agents. For example, since the relative position between agents changes as agents generate the desired formation from their initial positions, collision avoidance between agents is essential [27-31]. However, previous studies have not addressed the crossing motion of multiple robotic swarms and formation mix-ups in transient states, nor are the finding of those studies appropriate for the performance of motion including crossing. Although it is possible that crossing occurred in [17], the target was stick figures, different from that in a general mass game. While crossing motion between groups was treated in [19-21], problems such as formation mix-up, stationary postures, and excessive circumvention in an environment where there is a high density of robots has not yet been clarified. In addition, while [22] simulated the crossing of two groups, no formation was maintained. Thus, the objectives of these studies were not directed at the realization of a visually appealing, well-organized crossing motion for mass games. For a definition of visually appealing, well-organized crossing motion, see section 2.2.

We previously proposed a control method that realizes well-organized crossing motion of multiple robotic swarms [32]. The method consists of offline path planning based on model predictive control,

online trajectory tracking to the generated path, and collision avoidance based on RVO [19]. However, the optimized solution for the mixed-integer quadratic programming problem is used for the offline path planning, and thus requires a long computational time. In particular, computational time increases substantially with the number of robots, and it is difficult to realize crossing like that seen in human mass games with many robots.

One possible solution for this is a distributed controller. Computational time in a distributed controller does not depend on the number of robots, and thus it can be adapted to bigger robotic swarms. Wang *et al*. proposed a distributed controller that solves the optimal control problem by integrating formation control, collision avoidance, and trajectory tracking to control the formation of multiple UAVs [33]. Falconi *et al*. reported a distributed controller for both formation control and collision avoidance of groups of mobile robots based on the graph Laplacian [34]. However, these studies did not focus on crossing motion, or mention mix-ups in formation.

This paper proposes a distributed formation controller that considers formation maintenance, collision avoidance, and tracking to the desired trajectory in the transient states when multiple robotic swarms cross one another. With this controller, we realize visually appealing and highly entertaining mass robotic games. In the control of multi-agent systems, collision avoidance between agents is the essential requirement, and many collision avoidance methods have been proposed using artificial potential functions and barrier functions [27-31]. Nevertheless, well-organized crossing motion by multiple robotic swarms has not been realized. The distributed controller proposed here employs the artificial potential method to realize well-organized crossing motion. It could thus be applied to robotic mass games, or to exploring and monitoring the environment.

The latter application has the potential to quickly and efficiently perform assigned tasks. However, once multiple robotic swarms have crossed one another, perhaps finishing a task in different places, they generally return to the starting point. If the multiple robotic swarms explore different rooms in the interior of a building, for example, it is a possible there could be conflict at traffic intersections on the return pathway. In particular, space will be limited inside the building, and thus crossing motion with the fewest formation mix-ups will be necessary. The main difference between the proposed controller and our previously developed one [32] is that this one is a distributed controller, and thus is scalable with a low computational cost.

The remainder of this paper is organized as follows. The problem is formulated in section 2, and the distributed controller is proposed in section 3. Section 4 presents the mathematical analysis to prove the stability of the closed-loop system. Experiments and numerical simulations with mobile robots are described in section 5 to determine whether the well-organized crossing motion of multiple robotic swarms can be achieved. Finally, we conclude the paper in section 6.

2. Problem Formulation

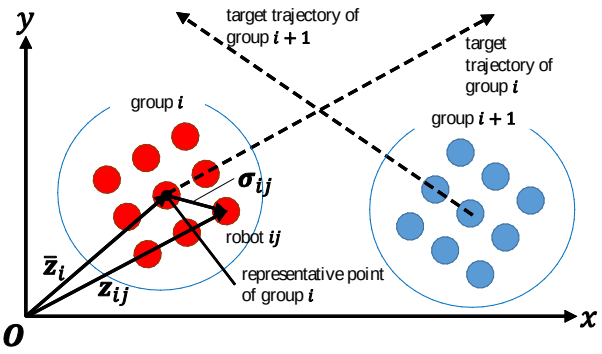


Fig. 1. Group *i* and robot *ij*.

2.1. Model of Robots and Swarms

Fig. 1 shows the *i*-th group consisting of m_i omni-directional mobile robots in the two-dimensional plane for $i = 1, \ldots, n$. We refer to the *i*-th group as group *i*. Let $\bar{\boldsymbol{z}}_i = [x_i, y_i]^T \in \mathbb{R}^{2\times 1}$ be the representative position of group *i*. The equations of motion of the *j*-th robot in group *i* are described by

$$\dot{\boldsymbol{z}}_{ij} = \boldsymbol{v}_{ij}, \tag{1}$$

$$\dot{\boldsymbol{v}}_{ij} = \boldsymbol{u}_{ij}, \tag{2}$$

where $j = 1, \ldots, m_i$, $\boldsymbol{z}_{ij} = [x_{ij}, y_{ij}]^T \in \mathbb{R}^{2\times 1}$ and $\boldsymbol{v}_{ij} = [\dot{x}_{ij}, \dot{y}_{ij}]^T \in \mathbb{R}^{2\times 1}$ are the position and velocity of the corresponding robot, respectively. In addition, $\boldsymbol{u}_{ij}=[u_{ijx}, u_{ijy}]^T \in \mathbb{R}^{2\times 1}$ is the acceleration input. The *j*-th robot in group *i* is denoted as robot *ij*. Here we assume that the size of robot *ij* fits inside a circle of radius *r*. The relationship between $\bar{\boldsymbol{z}}_i$ and position $\boldsymbol{z}_{ij}$ of robot *ij* is expressed as

$$\boldsymbol{z}_{ij} = \bar{\boldsymbol{z}}_i + \boldsymbol{\sigma}_{ij}, \tag{3}$$

where $\boldsymbol{\sigma}_{ij} \in \mathbb{R}^{2\times 1}$ is the relative position of robot *ij* from $\bar{\boldsymbol{z}}_i$.

The desired trajectory of group *i*'s representative position $\bar{\boldsymbol{z}}_i^d \in \mathbb{R}^{2\times 1}$, the number of robots, and the shape of the formation are given by the mass game program's designer. When the designer sets the desired trajectory, we assume that $\bar{\boldsymbol{z}}_i^d$ is designed without considering collision with the other groups. Here, let us set the following relationship between $\bar{\boldsymbol{z}}_i^d$ and the desired trajectory of robot *ij*, $\boldsymbol{z}_{ij}^d \in \mathbb{R}^{2\times 1}$:

$$\boldsymbol{z}_{ij}^d = \bar{\boldsymbol{z}}_i^d + \boldsymbol{\sigma}_{ij}^d, \tag{4}$$

where $\boldsymbol{\sigma}_{ij}^d \in \mathbb{R}^{2\times 1}$ is the relative desired position of robot *ij* from $\bar{\boldsymbol{z}}_i^d$, set by the designer of the mass game program. Now, we assume that $\boldsymbol{\sigma}_{ij}^d$ is constant; that is, the desired formation of the group does not change during the mass game (See the Appendix for definitions of frequently used symbols).

2.2. Well-Organized Crossing

We now consider the kind of crossing motion that will be visually appealing in a well-organized mass game. Fig. 2(a) and (b) show simulation results in which two robotic swarms with 2×4 formation cross each other. The colored circles represent the robots. A collision avoidance algorithm is installed in each robot, and thus collisions were avoided. The initial formations were the same in

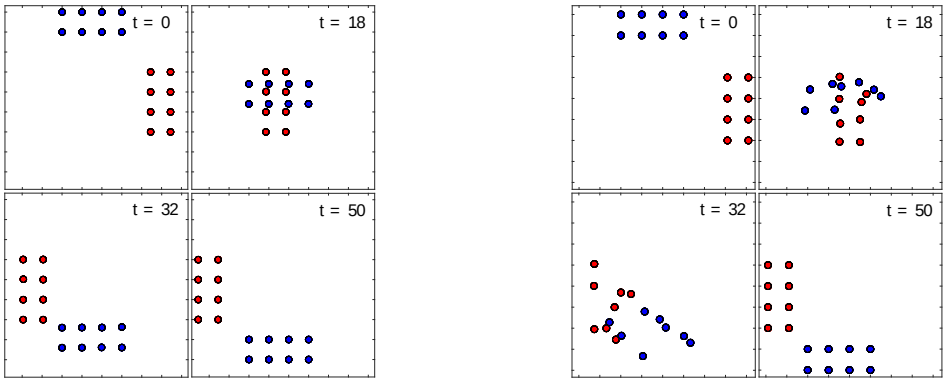
both cases, but the initial positions were not. The formation was maintained during the crossing in Fig. 2(a), but was mixed up during the crossing in 2(b). From these figures, we consider that the elements involved in the visual appeal of a well-organized crossing in a mass game include the beauty of the geometric form. Thus, we consider formation maintenance to be an important issue when deciding whether a crossing is visually appealing. As a result, we focus on formation maintenance during crossing. We also note that the non-crossing method, in which each group makes a big detour, fails in the crossing treated in this paper. It is therefore better to keep the error between the desired trajectory and the actual trajectory small.

To summarize, we set the robot crossing to be a visually appealing, well-organized crossing motion so that the formation would be maintained, and the error between the desired trajectory and actual trajectory would be as small as possible. Of course, collision avoidance is clearly essential. We propose a distributed controller satisfying the following three issues: 1) formation maintenance; 2) collision avoidance; and 3) tracking to the desired trajectory.

2.3. Control Objective

We describe the relationship between robots in group i by graph representation. Let $V_i = \{1, 2, \ldots, m_i\}$, and $E_i \subseteq V_i \times V_i$ be a set of nodes (robots) and a set of edges of undirected graph $G_i = (V_i, E_i)$. Here, V_i is the set of the robots in group i, and E_i is the set of communication links among the robots. Let $N_{ij} = \{k \in V_i : (j,k) \in E_i\}$ be the neighbors of robot ij. Thus, $l \in N_{ij} \Leftrightarrow$ robot ij can obtain the information of robot il. In this paper, we consider an undirected and connected graph for the network structure within the same group, and thus the set N_{ij} has the symmetry property: $l \in N_{ij} \Leftrightarrow j \in N_{il}$. Further, robot ij can know its own information, but $j \notin N_{ij}$. The graph Laplacian $L_i' \in \mathbb{R}^{m_i \times m_i}$ of G_i is defined as follows:

$$(L_i')_{(j,l)} = \begin{cases} |N_{ij}| & (j = l), \\ -1 & (l \in N_{ij}), \\ 0 & (\text{otherwise}), \end{cases} \quad , \tag{5}$$



(a) "well-organized" crossing. (b) not "well-organized" crossing.

Fig. 2. Crossing motion.

where $(L_i')_{(j,l)}$ represents the (j, l)-th element of matrix L_i', and $|N_{ij}|$ is the cardinal number of set N_{ij}. It is known that the graph Laplacian L_i' of G_i is symmetric and positive semidefinite, and that it has the following property: $L_i'\mathbf{1}_{m_i} = \mathbf{0}_{m_i}$, where $\mathbf{a}_k = [a,\ldots,a]^T \in \mathbb{R}^{k\times 1}$ for $a \in \mathbb{R}$ [35].

Robot ij must maintain the formation and avoid collisions using the local information of its neighbors prescribed by the Laplacian L_i' and the information of the robots in other groups. Thus, the control objective is to propose a distributed controller satisfying the followings: for all (i, j) and all (k, l) $\neq$ (i, j),

$$\left\| L_i(\boldsymbol{z}_i - \boldsymbol{\sigma}_i^d) \right\|_\infty \leq \varepsilon, \tag{6}$$

$$\left\| \boldsymbol{z}_{ij} - \boldsymbol{z}_{kl} \right\| \geq 2r, \tag{7}$$

$$\lim_{t\to\infty}(\boldsymbol{z}_{ij} - \boldsymbol{z}_{ij}^d) = \mathbf{0}, \tag{8}$$

where $\boldsymbol{z}_i = [\boldsymbol{z}_{i1}^T,\ldots,\boldsymbol{z}_{im_i}^T]^T \in \mathbb{R}^{2m_i\times 1}$, $\boldsymbol{v}_i = \dot{\boldsymbol{z}}_i \in \mathbb{R}^{2m_i\times 1}$, $\boldsymbol{u}_i = \dot{\boldsymbol{v}}_i \in \mathbb{R}^{2m_i\times 1}$, $\boldsymbol{z}_i^d = [(\boldsymbol{z}_{i1}^d)^T,\ldots,(\boldsymbol{z}_{im_i}^d)^T]^T \in \mathbb{R}^{2m_i\times 1}$, $\boldsymbol{v}_i^d = \dot{\boldsymbol{z}}_i^d \in \mathbb{R}^{2m_i\times 1}$, $\boldsymbol{u}_i^d = \dot{\boldsymbol{v}}_i^d \in \mathbb{R}^{2m_i\times 1}$, $\boldsymbol{\sigma}_i^d = [(\boldsymbol{\sigma}_{i1}^d)^T,\ldots,(\boldsymbol{\sigma}_{im_i}^d)^T]^T \in \mathbb{R}^{2m_i\times 1}$, and $L_i = L_i' \otimes \boldsymbol{I}_2 \in \mathbb{R}^{2m_i\times 2m_i}$. Here, $\boldsymbol{I}_k$ is an identity matrix of size k, and $\otimes$ is the Kronecker product. The reason that L_i is defined as $L_i = L_i' \otimes \boldsymbol{I}_2$ is because the states of robot $\boldsymbol{z}_{ij}$ and $\boldsymbol{v}_{ij}$ have two components of the x-axis and y-axis. For any vector $\boldsymbol{a} = [a_1, a_2]^T \in \mathbb{R}^{2\times 1}$,

$$L_i[\boldsymbol{a}^T,\ldots,\boldsymbol{a}^T]^T = \mathbf{0}, \tag{9}$$

holds. $\boldsymbol{\sigma}_i^d \in \mathbb{R}^{2m_i\times 1}$ is the offset vector to determine the desired form of the formation of group i.

The left-hand side (LHS) of (6) corresponds to a mix-up of the formation, and its value becomes large when the formation is disarranged. In fact, the LHS of (6) can be written as follows:

$$L_i(\boldsymbol{z}_i - \boldsymbol{\sigma}_i^d) = \begin{bmatrix} \sum_{j\in N_{i1}} \{(\boldsymbol{z}_{i1} - \boldsymbol{\sigma}_{i1}^d) - (\boldsymbol{z}_{ij} - \boldsymbol{\sigma}_{ij}^d)\} \\ \vdots \\ \sum_{j\in N_{im_i}} \{(\boldsymbol{z}_{im_i} - \boldsymbol{\sigma}_{im_i}^d) - (\boldsymbol{z}_{ij} - \boldsymbol{\sigma}_{ij}^d)\} \end{bmatrix} = \begin{bmatrix} \sum_{j\in N_{i1}} \{\boldsymbol{z}_{i1} - \boldsymbol{z}_{ij} - (\boldsymbol{\sigma}_{i1}^d - \boldsymbol{\sigma}_{ij}^d)\} \\ \vdots \\ \sum_{j\in N_{im_i}} \{\boldsymbol{z}_{im_i} - \boldsymbol{z}_{ij} - (\boldsymbol{\sigma}_{im_i}^d - \boldsymbol{\sigma}_{ij}^d)\} \end{bmatrix} \in \mathbb{R}^{2m_i\times 1}. \tag{10}$$

Each element of this vector is the sum of the error between the relative position and the desired relative position between the corresponding robot and its neighbors. Thus, the maximum value of such a vector's elements $\| L_i(\boldsymbol{z}_i - \boldsymbol{\sigma}_i^d) \|_\infty$ can be represented as a formation mix-up. In addition, we obtain $L_i(\boldsymbol{z}_i - \boldsymbol{\sigma}_i^d) = L_i[\bar{\boldsymbol{z}}_i^T,\ldots,\bar{\boldsymbol{z}}_i^T]^T = \mathbf{0}$ when the formation of the group coincides with the desired formation, $\boldsymbol{z}_i = [(\bar{\boldsymbol{z}}_i + \boldsymbol{\sigma}_{i1}^d)^T,\ldots,(\bar{\boldsymbol{z}}_i + \boldsymbol{\sigma}_{im_i}^d)^T]^T$. Considering this, condition (6) means that the mix-up of the formation is below a value ε, even if the state of the robot is transient, such as during a crossing motion. On the other hand, condition (7) means that the distance between two different robots is larger than or equal to 2r, and thus, the two robots do not collide. Further, condition (8) means that the error between $\boldsymbol{z}_{ij}$ and $\boldsymbol{z}_{ij}^d$ converges to zero as $t \to \infty$.

3. Distributed Formation Controller

In the control of multi-agent systems, collision avoidance between agents is an essential requirement, as mentioned, and many collision avoidance methods using the artificial potential functions and barrier functions have been proposed [27-31]. None, however, have realized well-organized crossing motion by multiple robotic swarms. Here, we employ the artificial potential method in designing the controller, and propose a distributed controller using the gradient of the cost functions about the conditions (6)-(8) to realize well-organized crossing motion for a mass game. In particular, the main points for realizing a well-organized crossing motion are defining an index about the mix-up of the formation by (6) and (10), and proposing the cost function for maintaining the formation with the fewest mix-ups by (14).

3.1. Cost Functions

Let us define the following cost functions:

$$J_z = J_{1,z} + J_{2,z} + J_{3,z}\,, \tag{11}$$

$$J_v = J_{1,v} + J_{2,v} + J_{3,v}\,, \tag{12}$$

where the subscripts z and v mean that the corresponding cost function is about the position and velocity, respectively. In addition, the subscripts 1, 2, and 3 show that the corresponding cost function is about formation maintenance, collision avoidance, and trajectory tracking, respectively. Using the cost functions J_z, J_v, and the desired trajectory of the acceleration $\boldsymbol{u}^d$, we design the controller as follows:

$$\boldsymbol{u} = -\frac{\partial J_z(\boldsymbol{x})}{\partial \boldsymbol{z}} - \frac{\partial J_v(\boldsymbol{x})}{\partial \boldsymbol{v}} + \boldsymbol{u}^d\,, \tag{13}$$

where $\boldsymbol{z} = [\boldsymbol{z}_1^T, \ldots, \boldsymbol{z}_n^T]^T \in \mathbb{R}^{2\bar{m}\times 1}$, $\boldsymbol{v} = \dot{\boldsymbol{z}} \in \mathbb{R}^{2\bar{m}\times 1}$, $\boldsymbol{u} = \dot{\boldsymbol{v}} \in \mathbb{R}^{2\bar{m}\times 1}$, $\boldsymbol{x} = [\boldsymbol{z}^T, \boldsymbol{v}^T]^T \in \mathbb{R}^{4\bar{m}\times 1}$, $\boldsymbol{u}^d = [(\boldsymbol{u}_1^d)^T, \ldots, (\boldsymbol{u}_n^d)^T]^T \in \mathbb{R}^{2\bar{m}\times 1}$, and $\bar{m} = \sum_{i=1}^{n}\sum_{j=1}^{m_i} 1$ is the total number of the robots in all groups.

3.2. Cost of Formation Maintenance

We introduce the following costs about the formation maintenance:

$$J_{1,z} = \frac{2\,\varepsilon\, c_{1,z}}{\pi}(\mathbf{1}_{2\bar{m}}^T \boldsymbol{D}^{cr}(\boldsymbol{z}) - 2\bar{m})\,, \tag{14}$$

$$J_{1,v} = \frac{c_{1,v}}{2}\boldsymbol{v}^T L \boldsymbol{v}\,, \tag{15}$$

where $c_{1,z}$ and $c_{1,v}$ are positive weighting coefficients. In addition,

$$\boldsymbol{D}^{cr}(\boldsymbol{z}) = \left[\frac{1}{\cos(D_{11,x})}, \frac{1}{\cos(D_{11,y})}, \frac{1}{\cos(D_{12,x})}, \frac{1}{\cos(D_{12,y})}, \cdots, \frac{1}{\cos(D_{nm_n,y})}\right]^T \in \mathbb{R}^{2\bar{m}\times 1}, \tag{16}$$

$$\begin{bmatrix} D_{ij,x} \\ D_{ij,y} \end{bmatrix} = \frac{\pi}{2\varepsilon} \sum_{l \in N_{ij}} \{(\boldsymbol{z}_{ij} - \boldsymbol{\sigma}_{ij}) - (\boldsymbol{z}_{il} - \boldsymbol{\sigma}_{il})\} = \boldsymbol{D}_{ij} \in \mathbb{R}^{2\times 1}, \tag{17}$$

$$\boldsymbol{D}(\boldsymbol{z}) = \frac{\pi}{2\varepsilon} L(\boldsymbol{z} - \boldsymbol{\sigma}) = [\boldsymbol{D}_{11}^T, \boldsymbol{D}_{12}^T, \ldots, \boldsymbol{D}_{nm_n}^T]^T \in \mathbb{R}^{2\tilde{m}\times 1}, \tag{18}$$

$$L = \mathrm{diag}(L_1, \ldots, L_n) \in \mathbb{R}^{2\tilde{m}\times 2\tilde{m}}, \qquad \boldsymbol{\sigma} = [(\boldsymbol{\sigma}_1^d)^T, \ldots, (\boldsymbol{\sigma}_n^d)^T]^T \in \mathbb{R}^{2\tilde{m}\times 1}. \tag{19}$$

Here, if any element of the vector $L(\boldsymbol{z}-\boldsymbol{\sigma})$ goes to $\pm\varepsilon$, the corresponding element of $\boldsymbol{D}$ becomes $\pm\pi/2$, and thus the corresponding element of $\boldsymbol{D}^{cr}$ diverges. That is, the motivation for selecting the cosine function in (16) is to obtain the situation in which the corresponding element of $\boldsymbol{D}^{cr}$ diverges when any element of the vector $L(\boldsymbol{z}-\boldsymbol{\sigma})$ goes to $\pm\varepsilon$. Therefore, $J_{1,z}$ diverges when (6) does not hold. On the other hand, $J_{1,v}$ represents the degree of velocity mismatch. The velocity mismatch between neighboring robots leads to a mix-up of the formation. To avoid this problem, $J_{1,v}$ is introduced.

The gradients of $J_{1,z}$ and $J_{1,v}$ can be computed as follows:

$$\frac{\partial J_{1,z}(\boldsymbol{x})}{\partial \boldsymbol{z}} = \frac{2\varepsilon c_{1,z}}{\pi} \frac{\partial \boldsymbol{D}}{\partial \boldsymbol{z}} \frac{\partial}{\partial \boldsymbol{D}} \boldsymbol{1}_{2\tilde{m}}^T \boldsymbol{D}^{cr}(\boldsymbol{z}) = c_{1,z} L \boldsymbol{D}^{sc}(\boldsymbol{z}), \tag{20}$$

$$\frac{\partial J_{1,v}(\boldsymbol{x})}{\partial \boldsymbol{v}} = c_{1,v} L \boldsymbol{v}, \tag{21}$$

where

$$\boldsymbol{D}^{sc}(\boldsymbol{z}) = \left[(\boldsymbol{D}_{11}^{sc})^T, \ldots, (\boldsymbol{D}_{nm_n}^{sc})^T \right]^T = \left[\frac{\sin(D_{11,x})}{\cos^2(D_{11,x})}, \cdots, \frac{\sin(D_{nm_n,y})}{\cos^2(D_{nm_n,y})} \right]^T \in \mathbb{R}^{2\tilde{m}\times 1}. \tag{22}$$

3.3. Cost of Collision Avoidance

We design the cost about the collision avoidance as follows:

$$J_{2,z} = c_{2,z} \sum_{i=1}^{n} \sum_{j=1}^{m_i} \sum_{(k,l)\in N_{ij}^z} G_{ij,kl}^z, \tag{23}$$

$$J_{2,v} = c_{2,v} \sum_{i=1}^{n} \sum_{j=1}^{m_i} \sum_{(k,l)\in N_{ij}^v} G_{ij,kl}^v, \tag{24}$$

where $c_{2,z}$ and $c_{2,v}$ are positive weighting coefficients, $N_{ij}^z = \{(k,l) : L_{ij,kl} < R_{\mathrm{safe}}, (k,l) \neq (i,j)\}$, $N_{ij}^v = \{(k,l) : L_{ij,kl} < R_{\mathrm{sense}}, \ k \neq i\}$, and $L_{ij,kl} = \| \boldsymbol{z}_{ij} - \boldsymbol{z}_{kl} \|$. Further, R_{safe} and R_{sense} are positive parameters larger than $2r$. For the derivation of the optimal values of R_{safe} and R_{sense}, we have not yet obtained the solution. Thus, we determined their values by trial and error in the experiment and numerical simulation. The cost (23) means that the robot keeps an appropriate distance from other robots that enter inside a circle with a radius R_{safe} around it, and (24) is the cost for the robot to avoid collisions with the robots inside a circle with a radius R_{sense} around it. Here, note that the range of the third summation in (24) has $k \neq i$, and thus the collision avoidance of (24) is not applicable to the robots belonging to same group. Thus, the collision avoidance in (24) is for robots belonging to

groups different from the robot's own group. Here, one guideline for setting R_{safe} and R_{sense} is to make sure it satisfies the following relation: $R_{\text{safe}} < R_{\text{sense}}$, where R_{sense} corresponds to the radius of the robot's sensing area. The cost (24) allows the robot to undertake collision avoidance behavior long before colliding with the robots belonging to the other group. If we consider only (23), collision avoidance is not assured before the robot could collide the other robots, and thus it is possible that the formation would be mixed up by the collision avoidance. However, combining (23) and (24) is expected to lessen this possibility. Therefore, satisfying $R_{\text{safe}} < R_{\text{sense}}$ is preferred for well-organized crossing motion with the least mix-up.

3.3.1. Collision avoidance cost by $J_{2,z}$

Let us define $G^z_{ij,kl}$ as follows:

$$G^z_{ij,kl} = \begin{cases} 0 & (L_{ij,kl} > R_{\text{safe}}), \\ g^z_{ij,kl} & (2r < L_{ij,kl} \le R_{\text{safe}}), \\ \text{not defined} & (L_{ij,kl} \le 2r), \end{cases} \tag{25}$$

$$g^z_{ij,kl} = \frac{1}{8\{R^2_{\text{safe}} - (2r)^2\}} \frac{(R^2_{\text{safe}} - L^2_{ij,kl})^2}{\{L^2_{ij,kl} - (2r)^2\}^2}. \tag{26}$$

Although $g^z_{ij,kl}$ is 0 when $L_{ij,kl} > R_{\text{safe}}$, $g^z_{ij,kl}$ increases smoothly with decreasing $L_{ij,kl}$ in the region of $L_{ij,kl} \le R_{\text{safe}}$. If $L_{ij,kl}$ increased to $2r$, that is, a collision were to occur, $g^z_{ij,kl}$ would diverge. Thus, $J_{2,z}$ diverges when (7) is not satisfied, and the input by the gradient of this cost becomes a repulsive force so that the robot moves away from another robot entering a circle of a radius R_{safe}. Here, note that the following hold from the definitions:

$$(k,l) \in N^z_{ij} \Leftrightarrow (i,j) \in N^z_{kl}, \tag{27}$$

$$L_{ij,kl} = L_{kl,ij}, \qquad G^z_{ij,kl} = G^z_{kl,ij}, \tag{28}$$

$$\frac{\partial G^z_{ij,kl}}{\partial \mathbf{z}_{ij}} = \frac{\partial G^z_{kl,il}}{\partial \mathbf{z}_{ij}}, \qquad \frac{\partial G^z_{ij,kl}}{\partial \mathbf{z}_{ij}} = -\frac{\partial G^z_{ij,kl}}{\partial \mathbf{z}_{kl}}. \tag{29}$$

So, the following equality holds:

$$\frac{\partial J_{2,z}}{\partial \mathbf{z}_{ij}} = c_{2,z} \sum_{(k,l)\in N^z_{ij}} \left(\frac{\partial G^z_{ij,kl}}{\partial \mathbf{z}_{ij}} + \frac{\partial G^z_{kl,ij}}{\partial \mathbf{z}_{ij}} \right) = 2c_{2,z} \sum_{(k,l)\in N^z_{ij}} \frac{\partial G^z_{ij,kl}}{\partial \mathbf{z}_{ij}}. \tag{30}$$

And the gradient of $J_{2,z}$ is written as

$$\frac{\partial J_{2,z}}{\partial \mathbf{z}_{ij}} = c_{2,z} \begin{bmatrix} -\sum_{(k,l)\in N^z_{11}} c^z_{11,kl}(\mathbf{z}_{11} - \mathbf{z}_{kl}) \\ \vdots \\ -\sum_{(k,l)\in N^z_{nm_n}} c^z_{nm_n,kl}(\mathbf{z}_{nm_n} - \mathbf{z}_{kl}) \end{bmatrix}, \tag{31}$$

where

$$c_{ij,kl}^{z} = \frac{(R_{\text{safe}}^2 - L_{ij,kl}^2)}{\{L_{ij,kl}^2 - (2r)^2\}^3} \in \mathbb{R} \ . \tag{32}$$

3.3.2. Collision avoidance cost by $J_{2,v}$

Next, let us define the following $G_{ij,kl}^{v}$ based on the procedures in [33]:

$$G_{ij,kl}^{v} = \begin{cases} 0 & (L_{ij,kl} > R_{\text{sense}}), \\ g_{ij,kl}^{v} & (2r < L_{ij,kl} \le R_{\text{sense}}), \\ \text{not defined} & (L_{ij,kl} \le 2r), \end{cases} \tag{33}$$

$$g_{ij,kl}^{z} = \begin{cases} 0 & (\alpha_{ij,kl} \ge \beta_{ij,kl}), \\ \dfrac{1}{4} \dfrac{(R_{\text{sense}}^2 - L_{ij,kl}^2)^2}{\{L_{ij,kl}^2 - (2r)^2\}^2} \| \boldsymbol{v}_{ij,kl} - \boldsymbol{v}_{ij,kl}^{\text{pref}} \|^2 & (\alpha_{ij,kl} < \beta_{ij,kl}), \end{cases} \tag{34}$$

where

$$\boldsymbol{v}_{ij,kl} = \boldsymbol{v}_{ij} - \boldsymbol{v}_{kl} \ , \qquad \boldsymbol{v}_{ij,kl}^{\text{pref}} = \underset{\boldsymbol{v} \in \{\boldsymbol{v}_{ij,kl}^{-}, \boldsymbol{v}_{ij,kl}^{+}\}}{\arg\min} \| \boldsymbol{v} - \boldsymbol{v}_{ij,kl} \| \ , \tag{35}$$

$$\boldsymbol{v}_{ij,kl}^{\pm} = C(\theta_{ij,kl}^{\pm}) \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} C^{-1}(\theta_{ij,kl}^{\pm}) \boldsymbol{v}_{ij,kl} \ , \tag{36}$$

$$\alpha_{ij,kl} = \arccos\left(\frac{(\boldsymbol{z}_{kl} - \boldsymbol{z}_{ij})^T \boldsymbol{v}_{ij,kl}}{\| \boldsymbol{z}_{kl} - \boldsymbol{z}_{ij} \| \| \boldsymbol{v}_{ij,kl} \|} \right), \tag{37}$$

$$\beta_{ij,kl} = \begin{cases} \arcsin\left(\dfrac{R_{\text{safe}}}{L_{ij,kl}} \right) & (L_{ij,kl} \ge R_{\text{safe}}), \\ \pi / 2 & (L_{ij,kl} < R_{\text{safe}}), \end{cases} \tag{38}$$

$$\theta_{ij,kl}^{\pm} = \operatorname{atan2}(y_{kl} - y_{ij}, x_{kl} - x_{ij}) \pm \beta_{ij,kl} \quad , \tag{39}$$

$$C(\theta) = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}, \qquad \operatorname{atan2}(y, x) = \arg(x + jy) \ , \tag{40}$$

where j in (40) is the imaginary unit. The geometric relations in (35)-(39) are shown in Fig. 3. As may be seen, $\boldsymbol{v}_{ij,kl}^{-}$ and $\boldsymbol{v}_{ij,kl}^{+}$ are projections of $\boldsymbol{v}_{ij,kl}$ onto two tangent lines of a circle of radius R_{safe} around $\boldsymbol{z}_{kl}$ through $\boldsymbol{z}_{ij}$. We set $\boldsymbol{v}_{ij,kl}^{\text{pref}}$ as the lesser value of the error norm between $\boldsymbol{v}_{ij,kl}^{-}$ and $\boldsymbol{v}_{ij,kl}$, and the error norm between $\boldsymbol{v}_{ij,kl}^{+}$ and $\boldsymbol{v}_{ij,kl}$. We show $\boldsymbol{v}_{ij,kl} - \boldsymbol{v}_{ij,kl}^{\text{pref}}$ as the red arrow in Fig. 3. In the case of Fig. 3, since $\| \boldsymbol{v}_{ij,kl} - \boldsymbol{v}_{ij,kl}^{-} \| < \| \boldsymbol{v}_{ij,kl} - \boldsymbol{v}_{ij,kl}^{+} \|$, we obtain $\boldsymbol{v}_{ij,kl}^{\text{pref}} = \boldsymbol{v}_{ij,kl}^{-}$. The collision avoidance cost, $g_{ij,kl}^{v}$, becomes zero when $\boldsymbol{v}_{ij,kl} = \boldsymbol{v}_{ij,kl}^{\text{pref}}$. So, in Fig. 3, the cost $g_{ij,kl}^{v}$ puts the relative positional relation of robot *ij* and robot *kl* on a tangential direction of a circle of radius R_{safe} around robot *kl*. In addition, the coefficient $(R_{\text{sense}}^2 - L_{ij,kl}^2)^2 / \{L_{ij,kl}^2 - (2r)^2\}^2$ increases smoothly as the distance between the robot *ij* and robot *kl* is closed. Here, note that if we set a large value for R_{safe}, it is difficult for a robot to cross another robotic swarm.

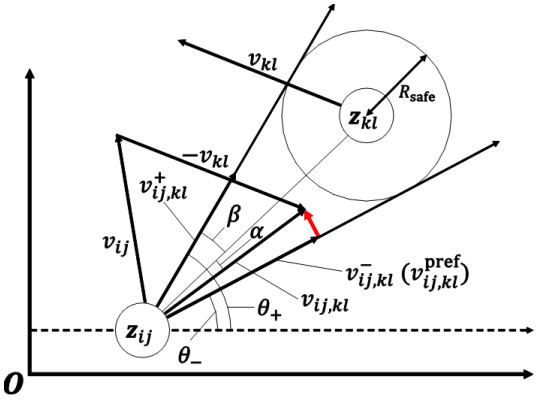


Fig. 3. Variables in collision-avoidance cost.

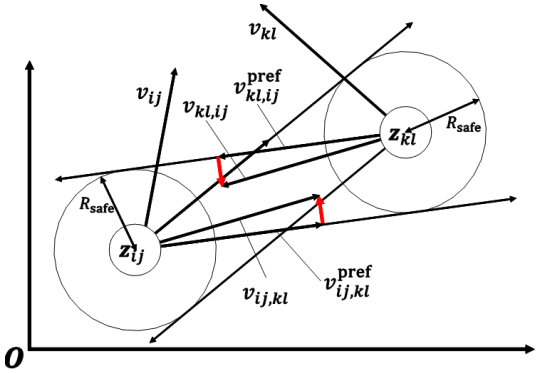


Fig. 4. $\boldsymbol{v}^{\text{pref}}_{ij,kl}$ and $\boldsymbol{v}^{\text{pref}}_{kl,ij}$.

Using the notation $\overline{N}^{v}_{ij} = N^{v}_{ij} \cap \{(k,l) : \alpha_{ij,kl} < \beta_{ij,kl}\}$ in (24) yields

$$J_{2,v} = c_{2,v} \sum_{i=1}^{n} \sum_{j=1}^{m_i} \sum_{(k,l)\in \overline{N}^{v}_{ij}} \frac{1}{4} \frac{(R^2_{\text{sense}} - L^2_{ij,kl})^2}{\{L^2_{ij,kl} - (2r)^2\}^2} \| v_{ij,kl} - v^{\text{pref}}_{ij,kl} \|^2 . \tag{41}$$

Now, let us define the matrix $C_2(\theta)$ as

$$C_2(\theta) = C(\theta) \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} C^{-1}(\theta) = \begin{bmatrix} \cos^2\theta & \cos\theta\sin\theta \\ \cos\theta\sin\theta & \sin^2\theta \end{bmatrix}. \tag{42}$$

Then, $\boldsymbol{v}^{\text{pref}}_{ij,kl}$ can be expressed as follows:

$$\boldsymbol{v}^{\text{pref}}_{ij,kl} = C_2(\theta^{\text{pref}}_{ij,kl})\boldsymbol{v}_{ij,kl}, \qquad \theta^{\text{pref}}_{ij,kl} = \underset{\theta\in\{\theta^{-}_{ij,kl},\theta^{+}_{ij,kl}\}}{\arg\min} \| (\boldsymbol{I}_2 - C_2(\theta))\boldsymbol{v}_{ij,kl} \| . \tag{43}$$

Here note that $C_2(\theta)$ and $\boldsymbol{I}_2$–$C_2(\theta)$ have the following properties regardless of θ:

$$C_2^T C_2 = C_2, \qquad (\boldsymbol{I}_2 - C_2)^T (\boldsymbol{I}_2 - C_2) = \boldsymbol{I}_2 - C_2 . \tag{44}$$

Thus, these two matrixes are semi-definite, and these properties lead to the following:

$$\| \boldsymbol{v}_{ij,kl} - \boldsymbol{v}^{\text{pref}}_{ij,kl} \|^2 = (\boldsymbol{v}_{ij,kl} - \boldsymbol{v}^{\text{pref}}_{ij,kl})^T (\boldsymbol{v}_{ij,kl} - \boldsymbol{v}^{\text{pref}}_{ij,kl}) = \boldsymbol{v}^T_{ij,kl} \left(\boldsymbol{I}_2 - C_2(\theta^{\text{pref}}_{ij,kl})\right) \boldsymbol{v}_{ij,kl} . \tag{45}$$

On the other hand, from (35), (37)-(39), and Fig. 4, which shows $\boldsymbol{v}^{\text{pref}}_{ij,kl}$ and $\boldsymbol{v}^{\text{pref}}_{kl,ij}$, we found

$$\boldsymbol{v}_{kl,ij} = -\boldsymbol{v}_{ij,kl}, \qquad \alpha_{kl,ij} = \alpha_{ij,kl}, \tag{46}$$

$$\beta_{kl,ij} = \beta_{ij,kl}, \qquad \theta^{\text{pref}}_{kl,ij} = \theta^{\text{pref}}_{ij,kl} + \pi . \tag{47}$$

Here, the following relation holds because of $C_2(\theta+\pi) = C_2(\theta)$:

$$\boldsymbol{v}^{\text{pref}}_{kl,ij} = C_2(\theta^{\text{pref}}_{kl,ij})\boldsymbol{v}_{kl,ij} = -\boldsymbol{v}^{\text{pref}}_{ij,kl} . \tag{48}$$

Further, since $(k,l) \in \overline{N}^{v}_{ij} \Leftrightarrow (i,j) \in \overline{N}^{v}_{kl}$ because of (46) and (47), we can show that $G^{v}_{ij,kl}$ has the following property:

$$G^{v}_{ij,kl} = G^{v}_{kl,ij}, \qquad \frac{\partial G^{v}_{ij,kl}}{\partial \boldsymbol{v}_{ij}} = \frac{\partial G^{v}_{kl,ij}}{\partial \boldsymbol{v}_{ij}}, \qquad \frac{\partial G^{v}_{ij,kl}}{\partial \boldsymbol{v}_{ij}} = -\frac{\partial G^{v}_{kl,ij}}{\partial \boldsymbol{v}_{kl}} . \tag{49}$$

Thus, the following equality holds:

$$\frac{\partial J_{2,v}}{\partial \boldsymbol{v}_{ij}} = 2c_{2,v} \sum_{(k,l)\in \bar{N}_{ij}^{v}} \frac{\partial G_{ij,kl}^{v}}{\partial \boldsymbol{v}_{ij}}. \tag{50}$$

From (45) and (50), we obtain

$$\frac{\partial J_{2,v}}{\partial \boldsymbol{v}} = c_{2,v} \begin{bmatrix} \sum_{(k,l)\in \bar{N}_{11}^{v}} c_{11,kl}^{v} (\boldsymbol{v}_{11,kl} - \boldsymbol{v}_{11,kl}^{\text{pref}}) \\ \vdots \\ \sum_{(k,l)\in \bar{N}_{nm_n}^{v}} c_{nm_n,kl}^{v} (\boldsymbol{v}_{nm_n,kl} - \boldsymbol{v}_{nm_n,kl}^{\text{pref}}) \end{bmatrix}, \tag{51}$$

where

$$c_{ij,kl}^{v} = \frac{(R_{\text{sense}}^2 - L_{ij,kl}^2)^2}{\{L_{ij,kl}^2 - (2r)^2\}^2} \in \mathbb{R}. \tag{52}$$

3.4. Cost of Tracking

The tracking costs $J_{3,z}$ and $J_{3,v}$ are defined as follows:

$$J_{3,z} = \frac{c_{3,z}}{2} (\boldsymbol{z} - \boldsymbol{z}^d)^T (\boldsymbol{z} - \boldsymbol{z}^d), \tag{53}$$

$$J_{3,v} = \frac{c_{3,v}}{2} (\boldsymbol{v} - \boldsymbol{v}^d)^T (\boldsymbol{v} - \boldsymbol{v}^d), \tag{54}$$

where $\boldsymbol{z}^d = [(\boldsymbol{z}_1^d)^T, \ldots, (\boldsymbol{z}_n^d)^T]^T \in \mathbb{R}^{2\tilde{m}\times 1}$ and $\boldsymbol{v}^d = [(\boldsymbol{v}_1^d)^T, \ldots, (\boldsymbol{v}_n^d)^T]^T \in \mathbb{R}^{2\tilde{m}\times 1}$ are given desired trajectories, and $c_{3,z}$ and $c_{3,v}$ are positive weighting coefficients. The gradients of the tracking costs can be computed as follows:

$$\frac{\partial J_{3,z}}{\partial \boldsymbol{z}} = c_{3,z} (\boldsymbol{z} - \boldsymbol{z}^d), \qquad \frac{\partial J_{3,v}}{\partial \boldsymbol{z}} = c_{3,v} (\boldsymbol{v} - \boldsymbol{v}^d). \tag{55}$$

3.5. Control Input

To summarize, the control input can be written as

$$\begin{aligned} \boldsymbol{u} = {} & -c_{1,z} L \boldsymbol{D}^{sc} - c_{1,v} L \boldsymbol{v} - c_{2,z} \begin{bmatrix} -\sum_{(k,l)\in N_{11}^{z}} c_{11,kl}^{z} (\boldsymbol{z}_{11} - \boldsymbol{z}_{kl}) \\ \vdots \\ -\sum_{(k,l)\in N_{nm_n}^{z}} c_{nm_n,kl}^{z} (\boldsymbol{z}_{nm_n} - \boldsymbol{z}_{kl}) \end{bmatrix} - c_{2,v} \begin{bmatrix} \sum_{(k,l)\in \bar{N}_{11}^{v}} c_{11,kl}^{v} (\boldsymbol{v}_{11,kl} - \boldsymbol{v}_{11,kl}^{\text{pref}}) \\ \vdots \\ \sum_{(k,l)\in \bar{N}_{nm_n}^{v}} c_{nm_n,kl}^{v} (\boldsymbol{v}_{nm_n,kl} - \boldsymbol{v}_{nm_n,kl}^{\text{pref}}) \end{bmatrix} \\ & - c_{3,z} (\boldsymbol{z} - \boldsymbol{z}^d) - c_{3,v} (\boldsymbol{v} - \boldsymbol{v}^d) + \boldsymbol{u}^d. \end{aligned} \tag{56}$$

Here, the control input $\boldsymbol{u}_{ij}$ for robot *ij* becomes

$$\begin{aligned}\boldsymbol{u}_{ij} = &-\sum_{l\in N_{ij}}\{c_{1,z}(\boldsymbol{D}_{ij}^{SC}-\boldsymbol{D}_{il}^{SC})+c_{1,v}(\boldsymbol{v}_{ij}-\boldsymbol{v}_{il})\}-c_{2,z}\sum_{(k,l)\in N_{ij}^{z}}c_{ij,kl}^{z}(\boldsymbol{z}_{kl}-\boldsymbol{z}_{ij})-c_{2,v}\sum_{(k,l)\in \overline{N}_{ij}^{v}}c_{ij,kl}^{v}(\boldsymbol{v}_{ij,kl}-\boldsymbol{v}_{ij,kl}^{\text{pref}})\\ &-c_{3,z}(\boldsymbol{z}_{ij}-\boldsymbol{z}_{ij}^{d})-c_{3,v}(\boldsymbol{v}_{ij}-\boldsymbol{v}_{ij}^{d})+\boldsymbol{u}_{ij}^{d}.\end{aligned} \tag{57}$$

We find that this control input uses the information of the robots inside a distance, and of the robots specified by the graph Laplacian. Thus, this controller is the distributed controller.

4. Stability Analysis

The closed-loop system under the controller (56) can be written as

$$\begin{cases}\dot{\boldsymbol{z}}=\boldsymbol{v}\\ \dot{\boldsymbol{v}}=-c_{1,z}L\boldsymbol{D}^{sc}-c_{1,v}L\boldsymbol{v}-c_{2,z}\begin{bmatrix}-\sum_{(k,l)\in N_{11}^{z}}c_{11,kl}^{z}(\boldsymbol{z}_{11}-\boldsymbol{z}_{kl})\\ \vdots\\ -\sum_{(k,l)\in N_{nm_n}^{z}}c_{nm_n,kl}^{z}(\boldsymbol{z}_{nm_n}-\boldsymbol{z}_{kl})\end{bmatrix}-c_{2,v}\begin{bmatrix}\sum_{(k,l)\in \overline{N}_{11}^{v}}c_{11,kl}^{v}(\boldsymbol{v}_{11,kl}-\boldsymbol{v}_{11,kl}^{\text{pref}})\\ \vdots\\ \sum_{(k,l)\in \overline{N}_{nm_n}^{v}}c_{nm_n,kl}^{v}(\boldsymbol{v}_{nm_n,kl}-\boldsymbol{v}_{nm_n,kl}^{\text{pref}})\end{bmatrix}\\ \quad -c_{3,z}(\boldsymbol{z}-\boldsymbol{z}^{d})-c_{3,v}(\boldsymbol{v}-\boldsymbol{v}^{d})+\boldsymbol{u}^{d}\end{cases} \tag{58}$$

Now, we investigate if the closed-loop system (58) satisfies conditions (6)-(8).

4.1. Fulfill the Conditions about Formation Maintenance and Collision Avoidance

First, we set several assumptions.

Assumption 1: Conditions (6) and (7) are satisfied in the initial states at $t=0$.

Assumption 2: The values of $\|\boldsymbol{v}(0)\|$ and $\|\boldsymbol{z}(0)-\boldsymbol{z}^{d}(0)\|$ are bounded in the initial states at $t=0$.

Assumption 3: $N_{ij}^{z}\bigcup\overline{N}_{ij}^{v}\neq\text{ø}$ for all i and j $\Rightarrow$ $\dot{\boldsymbol{z}}^{d}=\boldsymbol{v}^{d}=\boldsymbol{0}$.

Here, ø is an empty set. Assumption 3 means that the desired trajectory is the static desired point in the case where some robots approach and avoid collision.

Theorem 1: Under Assumption 1, 2, and 3, the closed-loop system (58) satisfies conditions (6) and (7).

Proof: For condition (6) and formation maintenance cost $J_{1,z}$, and for condition (7) and collision avoidance cost $J_{2,z}$, the following hold: for all (i, j) and for all $(k, l)\neq(i, j)$,

$$\lim_{\|L(\boldsymbol{z}-\boldsymbol{\sigma})\|_{\infty}\to\varepsilon-0}J_{1,z}=\infty,\qquad \lim_{\|\boldsymbol{z}_{ij}-\boldsymbol{z}_{kl}\|\to 2r-0}J_{2,z}=\infty, \tag{59}$$

where $\lim_{x\to a-0}f(x)$ is the left-hand limit of $f(x)$ at $x=a$. Thus, if assumption 1 is satisfied, we obtain

$$J_{1,z}<M_1\Rightarrow\|L(\boldsymbol{z}-\boldsymbol{\sigma})\|_{\infty}<\varepsilon,\qquad J_{2,z}<M_2\Rightarrow\|\boldsymbol{z}_{ij}-\boldsymbol{z}_{kl}\|<2r, \tag{60}$$

for all (i, j) and for all $(k, l)\neq(i, j)$. Here, M_1 and M_2 are sufficiently large positive constants. That is,

conditions (6) and (7) are satisfied if we show that $J_{1,z}$ and $J_{2,z}$ are bounded.

Using the variable transformations $\hat{\boldsymbol{z}} = \boldsymbol{z} - \boldsymbol{z}^d$, $\hat{\boldsymbol{v}} = \boldsymbol{v} - \boldsymbol{v}^d$, $\hat{\boldsymbol{u}} = \boldsymbol{u} - \boldsymbol{u}^d$, $\hat{\boldsymbol{x}} = [\hat{\boldsymbol{z}}^T, \hat{\boldsymbol{v}}^T]^T$ in (56) yields

$$\hat{\boldsymbol{u}} = -c_{1,z} L \boldsymbol{D}^{sc}(\hat{\boldsymbol{z}}) - c_{1,v} L \hat{\boldsymbol{v}} - c_{2,z} \sum_{i=1}^{n} \sum_{j=1}^{m_i} \sum_{(k,l) \in N_{ij}^z} \frac{\partial G_{ij,kl}^z}{\partial \hat{\boldsymbol{z}}} - c_{2,v} \sum_{i=1}^{n} \sum_{j=1}^{m_i} \sum_{(k,l) \in \bar{N}_{ij}^v} \frac{\partial G_{ij,kl}^v}{\partial \hat{\boldsymbol{v}}} - c_{3,z} \hat{\boldsymbol{z}} - c_{3,v} \hat{\boldsymbol{v}}. \tag{61}$$

Now, let us define the Lyapunov function candidate as follows:

$$V(\hat{\boldsymbol{x}}) = \frac{1}{2} \hat{\boldsymbol{v}}^T \hat{\boldsymbol{v}} + J_{1,z} + J_{2,z} + J_{3,z} . \tag{62}$$

Here, note that $V(\hat{\boldsymbol{x}}) > 0$ for all $\hat{\boldsymbol{x}} \neq 0$, but we cannot say $V(\hat{\boldsymbol{x}}) = 0$ when $\hat{\boldsymbol{x}} = 0$. The time derivative of $V(\hat{\boldsymbol{x}})$ along the solution of (58) is given by

$$\begin{aligned}
\dot{V}(x) &= \hat{\boldsymbol{v}}^T \hat{\boldsymbol{u}} + c_{1,z} \hat{\boldsymbol{v}}^T L \boldsymbol{D}^{sc}(\hat{\boldsymbol{z}}) + c_{2,z} (\hat{\boldsymbol{v}} + \boldsymbol{v}^d)^T \sum_{i=1}^{n} \sum_{j=1}^{m_i} \sum_{(k,l) \in N_{ij}^z} \frac{\partial G_{ij,kl}^z}{\partial \hat{\boldsymbol{z}}} + c_{3,z} \hat{\boldsymbol{v}}^T \hat{\boldsymbol{z}} \\
&= -c_{1,v} \hat{\boldsymbol{v}}^T L \hat{\boldsymbol{v}} - c_{2,v} \hat{\boldsymbol{v}}^T \sum_{i=1}^{n} \sum_{j=1}^{m_i} \sum_{(k,l) \in \bar{N}_{ij}^v} \frac{\partial G_{ij,kl}^v}{\partial \hat{\boldsymbol{v}}} - c_{3,v} \hat{\boldsymbol{v}}^T \hat{\boldsymbol{v}} + c_{2,z} \sum_{i=1}^{n} \sum_{j=1}^{m_i} \sum_{(k,l) \in N_{ij}^z} c_{ij,kl}^z (\boldsymbol{v}_{ij}^d)^T (\boldsymbol{z}_{kl} - \boldsymbol{z}_{ij}) \\
&= -c_{1,v} \hat{\boldsymbol{v}}^T L \hat{\boldsymbol{v}} - c_{2,v} \sum_{i=1}^{n} \sum_{j=1}^{m_i} \sum_{(k,l) \in \bar{N}_{ij}^v} c_{ij,kl}^v \boldsymbol{v}_{ij}^T (\boldsymbol{v}_{ij,kl} - \boldsymbol{v}_{ij,kl}^{\text{pref}}) \\
&\quad - c_{3,v} \hat{\boldsymbol{v}}^T \hat{\boldsymbol{v}} + c_{2,v} \sum_{i=1}^{n} \sum_{j=1}^{m_i} \sum_{(k,l) \in \bar{N}_{ij}^v} c_{ij,kl}^v (\boldsymbol{v}_{ij}^d)^T (\boldsymbol{v}_{ij,kl} - \boldsymbol{v}_{ij,kl}^{\text{pref}}) + \frac{c_{2,z}}{2} \sum_{i=1}^{n} \sum_{j=1}^{m_i} \sum_{(k,l) \in N_{ij}^z} c_{ij,kl}^z (\boldsymbol{v}_{ij,kl}^d)^T \boldsymbol{z}_{kl,ij} \\
&= -c_{1,v} \hat{\boldsymbol{v}}^T L \hat{\boldsymbol{v}} - \frac{c_{2,v}}{2} \sum_{i=1}^{n} \sum_{j=1}^{m_i} \sum_{(k,l) \in \bar{N}_{ij}^v} c_{ij,kl}^v \boldsymbol{v}_{ij,kl}^T (\boldsymbol{v}_{ij,kl} - \boldsymbol{v}_{ij,kl}^{\text{pref}}) \\
&\quad - c_{3,v} \hat{\boldsymbol{v}}^T \hat{\boldsymbol{v}} + \frac{c_{2,v}}{2} \sum_{i=1}^{n} \sum_{j=1}^{m_i} \sum_{(k,l) \in \bar{N}_{ij}^v} c_{ij,kl}^v (\boldsymbol{v}_{ij,kl}^d)^T (\boldsymbol{v}_{ij,kl} - \boldsymbol{v}_{ij,kl}^{\text{pref}}) + \frac{c_{2,z}}{2} \sum_{i=1}^{n} \sum_{j=1}^{m_i} \sum_{(k,l) \in N_{ij}^z} c_{ij,kl}^z (\boldsymbol{v}_{ij,kl}^d)^T \boldsymbol{z}_{kl,ij},
\end{aligned} \tag{63}$$

where $\boldsymbol{v}_{ij,kl}^d = \boldsymbol{v}_{ij}^d - \boldsymbol{v}_{kl}^d$, and $\boldsymbol{z}_{kl,\,ij} = \boldsymbol{z}_{kl} - \boldsymbol{z}_{ij}$. From assumption 3, we obtain $\boldsymbol{v}_{ij,kl}^d = \boldsymbol{0}$, and thus

$$\dot{V}(x) = -c_{1,v} \hat{\boldsymbol{v}}^T L \hat{\boldsymbol{v}} - c_{3,v} \hat{\boldsymbol{v}}^T \hat{\boldsymbol{v}} - \frac{c_{2,v}}{2} \sum_{i=1}^{n} \sum_{j=1}^{m_i} \sum_{(k,l) \in \bar{N}_{ij}^v} c_{ij,kl}^v \boldsymbol{v}_{ij,kl}^T (\boldsymbol{I}_2 - C_2(\theta_{ij,kl}^{\text{pref}})) \boldsymbol{v}_{ij,kl}. \tag{64}$$

Here, the graph Laplacian L is positive semi-definite, $c_{ij,kl}^v$ is a positive scalar, and $\boldsymbol{I}_2 - C_2(\theta_{ij,kl}^{\text{pref}})$ is positive semi-definite because of (44). Thus, we obtain

$$\dot{V}(x) \leq 0. \tag{65}$$

Further, $\dot{V}(x)$ becomes zero only if $\hat{\boldsymbol{v}} = 0$ and $\bar{N}^v = \text{ø}$. Here,

$$\bar{N}^v = \bigcup_{i=1}^{n} \bigcup_{j=1}^{m_i} \bar{N}_{ij}^v, \tag{66}$$

and $\bar{N}^v = \text{ø}$ means that $J_{2,v}$ is zero and the third term of the right-hand side (RHS) in (64) is zero. In addition, the initial value of V ($V(t = 0) = V_0$) is finite, and this leads to

$$\frac{1}{2} \hat{\boldsymbol{v}}^T \hat{\boldsymbol{v}} + J_{1,z} + J_{2,z} + J_{3,z} \leq V_0. \tag{67}$$

Here, all terms of the LHS of (67) are larger than or equal to zero, and the following holds:

$$J_{1,z} \le V_0, \quad J_{2,z} \le V_0. \tag{68}$$

Therefore, the closed-loop system (58) satisfies conditions (6) and (7). ■

4.2. Convergence

We investigated the convergence of the solution of the closed-loop system (58), and showed that condition (8) is satisfied. Now, we set several assumptions.

Assumption 4: $\| \boldsymbol{z}_{ij}^d - \boldsymbol{z}_{il}^d \| = \| \boldsymbol{\sigma}_{ij}^d - \boldsymbol{\sigma}_{il}^d \| > R_{\text{safe}}, \quad \forall (i, j), \ \forall l \neq j$.

Assumption 5: The parameter ε in (6) is sufficiently small, and collision avoidance behavior does not occur in robots of the same group.

Assumption 6: After a sufficiently large amount of time has passed, the crossing motions between robotic swarms are finished, and $\overline{N}^v = \text{ø}$ and $N^z = \text{ø}$ hold, where $N^z = \bigcup_{i=1}^{n} \bigcup_{j=1}^{m_i} N_{ij}^z$.

Assumption 4 indicates that, in the desired formation, all robots belonging to the same group keep more than R_{safe} away from their neighbors. Using this assumption leads to $V(\hat{\boldsymbol{x}}) = 0$ when $\hat{\boldsymbol{x}} = 0$. Assumption 5 means that the distance between robots in the same group is not less than R_{safe}, because all robots belonging to the same group keep more than R_{safe} away from their neighbors by virtue of Assumption 4, Theorem 1, (6), (10), and the assumption that ε is sufficiently small. Assumption 6 means that the robots do not have to consider collision avoidance with robots in other groups after a sufficient amount of time has passed. In this paper, we consider the well-organized crossing motion, like human mass games. The mass games have crossing motions, which lead to visually amusing motion, but there are few ending with crossing motion in the mass game program. Therefore, we consider Assumption 6 is not too strong for the mass game.

Theorem 2: Under Assumption 4, 5, and 6, the closed-loop system (58) satisfies condition (8).

Proof: From Assumption 6, the second and fourth terms in the RHS of (63) become zero, and the fifth term becomes the following:

$$\frac{c_{2,z}}{2} \sum_{i=1}^{n} \sum_{j=1}^{m_i} \sum_{(k,l) \in N_{ij}^z} c_{ij,kl}^z (\boldsymbol{v}_{ij,kl}^d)^T \boldsymbol{z}_{kl,ij} = \frac{c_{2,z}}{2} \sum_{i=1}^{n} \sum_{j=1}^{m_i} \sum_{l \in N_{ij,i}^z} c_{ij,il}^z (\boldsymbol{v}_{ij,il}^d)^T \boldsymbol{z}_{il,ij} = 0, \tag{69}$$

where $N_{ij,i}^z = \{ l : L_{ij,il} < R_{\text{safe}} \}$ (note that $\boldsymbol{v}_{ij,il}^d = \boldsymbol{0}$). Thus, (63) can be rewritten as

$$\dot{V}(x) = -c_{1,v} \hat{\boldsymbol{v}}^T L \hat{\boldsymbol{v}} - c_{3,v} \hat{\boldsymbol{v}}^T \hat{\boldsymbol{v}} \le 0 . \tag{70}$$

Here, $\dot{V}(x)$ becomes zero only if $\hat{\boldsymbol{v}} = 0$, and in that case, $\hat{\boldsymbol{u}} = 0$ because of $\dot{\hat{\boldsymbol{v}}} = 0$. Substituting these facts into (61) yields

$$\mathbf{0} = -c_{1,z} L\boldsymbol{D}^{sc}(\hat{\boldsymbol{z}}) - c_{2,z} \sum_{i=1}^{n} \sum_{j=1}^{m_i} \sum_{(k,l)\in N_{ij}^z} \frac{\partial G_{ij,kl}^z}{\partial \hat{\mathbf{z}}} - c_{3,z} \hat{\boldsymbol{z}} \,. \tag{71}$$

Using Assumption 5 in (71) leads to

$$\mathbf{0} = -c_{1,z} L\boldsymbol{D}^{sc}(\hat{\boldsymbol{z}}) - c_{3,z} \hat{\boldsymbol{z}} \,. \tag{72}$$

Now, we show that (72) is satisfied only if $\hat{\boldsymbol{z}} = \mathbf{0}$. Multiplying $\hat{\boldsymbol{z}}^T$ in (72) leads to

$$\mathbf{0} = -c_{1,z} \hat{\boldsymbol{z}}^T L\boldsymbol{D}^{sc}(\hat{\boldsymbol{z}}) - c_{3,z} \hat{\boldsymbol{z}}^T \hat{\boldsymbol{z}} \,. \tag{73}$$

The first term of the RHS in (73) can be expanded as

$$\hat{\boldsymbol{z}}^T L\boldsymbol{D}^{sc}(\hat{\boldsymbol{z}}) = (L\hat{\boldsymbol{z}})^T \boldsymbol{D}^{sc}(\hat{\boldsymbol{z}}) = \frac{2\varepsilon}{\pi} \boldsymbol{D}^T \boldsymbol{D}^{sc}(\hat{\boldsymbol{z}}) = \frac{2\varepsilon}{\pi} \sum_{i=1}^{n} \sum_{j=1}^{m_i} \left(D_{ij,x} \frac{\sin(D_{ij,x})}{\cos^2(D_{ij,x})} + D_{ij,y} \frac{\sin(D_{ij,y})}{\cos^2(D_{ij,y})} \right). \tag{74}$$

In the derivation of the second equality, we used (9) and (18). Since $\hat{\boldsymbol{z}}$ satisfies condition (6), we obtain $|D_{ij,x}| < \pi/2$ and $|D_{ij,y}| < \pi/2$. This means

$$D_{ij,x} \frac{\sin(D_{ij,x})}{\cos^2(D_{ij,x})} \geq 0, \quad D_{ij,y} \frac{\sin(D_{ij,y})}{\cos^2(D_{ij,y})} \geq 0, \tag{75}$$

and thus, $\hat{\boldsymbol{z}}^T L\boldsymbol{D}^{sc}(\hat{\boldsymbol{z}}) \geq 0$ if $\hat{\boldsymbol{z}} \neq \mathbf{0}$. Thus, if $\hat{\boldsymbol{z}} \neq \mathbf{0}$,

$$-c_{1,z} \hat{\boldsymbol{z}}^T L\boldsymbol{D}^{sc}(\hat{\boldsymbol{z}}) - c_{3,z} \hat{\boldsymbol{z}}^T \hat{\boldsymbol{z}} < 0 \,. \tag{76}$$

This is in contradiction to (72), and (72) is satisfied only if $\hat{\boldsymbol{z}} = \mathbf{0}$. Therefore, from the LaSalle's invariant principle, we found that condition (8) holds. ■

Therefore, we found that the distributed controller (56) satisfies conditions (6)-(8).

5. Experiment and Numerical Simulation

To investigate whether the well-organized crossing motion can be realized, we carried out an experiment. For the experiment, we used six mobile robots. As six was the maximum number we

Fig. 5. Omni-directional robot.

TABLE 1 PARAMETERS USED IN EXPERIMENTS

Parameter	Value
Control period T_s	0.1 s
Robot radius r	0.12 m
R_{safe}	0.36 m
R_{sense}	10 m
$c_{1,z}$	0.010
$c_{1,v}$	1.000
$c_{2,z}$	0.002×10^{-3}
$c_{2,v}$	0.002×10^{-1}
$c_{3,z}$	0.005
$c_{3,v}$	0.200

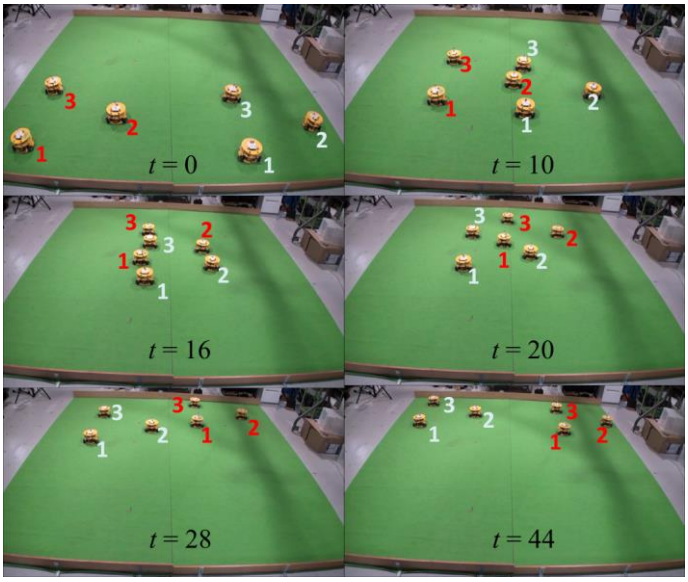
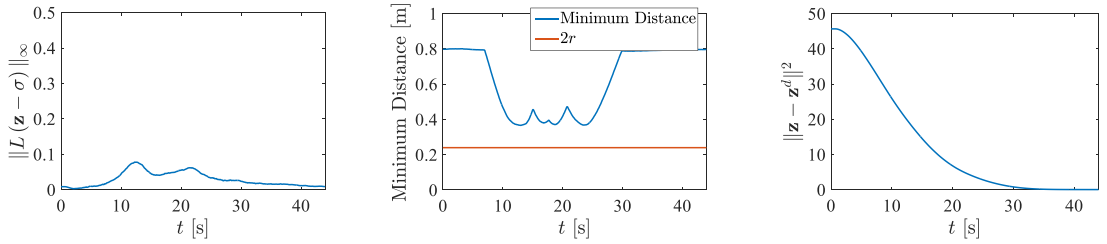


Fig. 6. Screenshots of experiment by proposed method.



(a) The index for formation mix-up (b) The minimum distance (c) The error norm of the trajectory tracking

Fig. 7. Experimental results by proposed method.

could prepare, to consider the motion for a larger number of robots, we also carried out numerical simulations.

5.1. Experiment

5.1.1. Experimental Environment

Fig. 5 shows the omni-directional mobile robot used in this experiment. Each robot had three omniwheels. To measure the positions of robots, we used a motion-capture system that included high-speed cameras and tracking software. Although we used a global sensor for the motion-capture system, the controller used only the local information of each robot, and thus we constructed a distributed controller in the experiment. The controller was implemented in the computer, and the velocity command for each robot was communicated via Bluetooth. The robot was then controlled by the received velocity command.The size of the field where the robots moved was 3.5 m × 3.5 m. Although we were able to realize a decentralized controller, the experimental system is a centralized system. We do, however, consider that a decentralized controller in a decentralized system could be

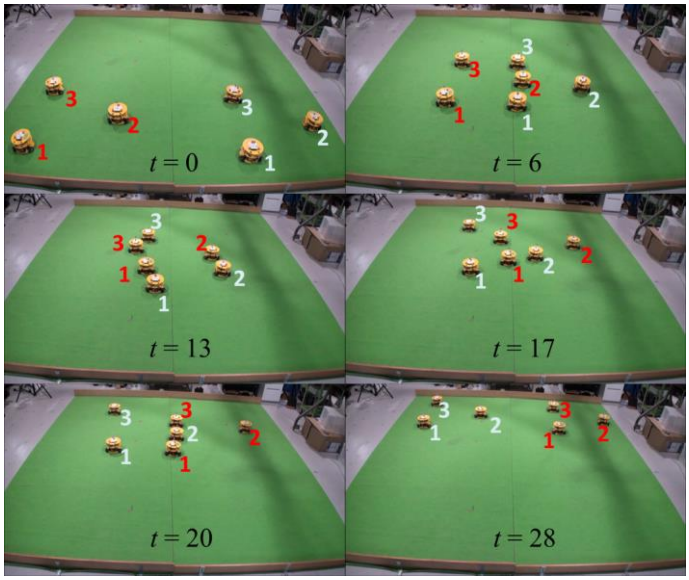
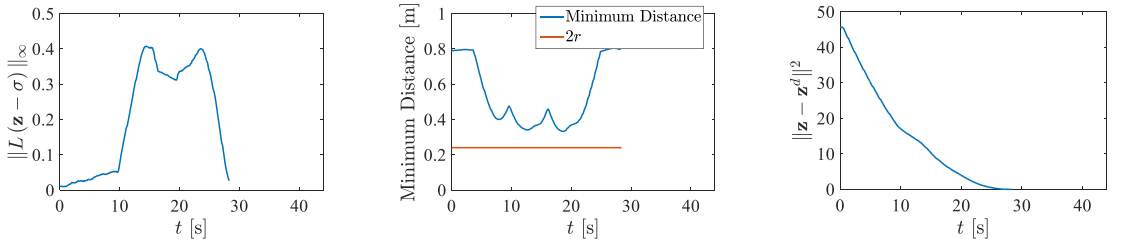


Fig. 8. Screenshots of experiment by RVO.



(a) The index for formation mix-up (b) The minimum distance (c) The error norm of the trajectory tracking

Fig. 9. Experimental results by RVO.

achieved as follows: A computer and a laser rangefinder are mounted on each robot, and we assume that each robot knows its own initial position. The position of the robot is then measured by odometry, and the positions of other robots are measured by the laser rangefinder. It would then be possibble to construct the controller (57) using the obtained data.

The parameters of the robot and controller are listed in Table 1. In the experiment, two robotic swarms, with each swarm consisting of three robots, were obliquely crossed from the given initial position to the given final positions. Here, the desired values were fixed values; that is, $\dot{\boldsymbol{z}}^d = \boldsymbol{v}^d = 0$. For comparison, we carried out two experiments: experiment 1 used the proposed controller, and experiment 2 used only the collision avoidance method by RVO [19].

RVO is the distributed algorithm that finds the input velocity at which collisions do not occur. For robot *ij*, the algorithm finds the velocity regions RVO^{ij} where robot *ij* would collide with its neighbors if robot *ij* were to move at its present speed. The algorithm then seeks the velocity input for robot *ij* from the region outside. That is, the velocity input $\boldsymbol{u}_{ij}$ is determined for robot *ij* as $\boldsymbol{u}_{ij} = \min_{\boldsymbol{v} \notin RVO^{ij}} \| \boldsymbol{v} - \boldsymbol{u}_{ij}^d \|$, where $\boldsymbol{u}_{ij}^d$ is the desired velocity. In this experiment, we set $\boldsymbol{u}_{ij}^d$ as

$\boldsymbol{u}_{ij}^d = -\boldsymbol{K}_p(\boldsymbol{z}_{ij} - \boldsymbol{z}_{ij}^d)$, where $\boldsymbol{K}_p$ is the feedback gain set as the identity matrix. Here, note that the desired trajectory of robot *ij*, $\boldsymbol{z}_{ij}^d$, is the constant in this section.

5.1.2. Experimental Results

We first show the results of experiment 1. Fig. 6 contains screenshots of the experiment, and Fig. 7 shows the experimental results by the proposed method for $\| L_i(\boldsymbol{z}_i - \boldsymbol{\sigma}_i^d) \|_\infty$, the minimum distance between robots, and $\| \boldsymbol{z} - \boldsymbol{z}^d \|^2$. In each figure, the horizontal axis shows the time, and the vertical axis shows the corresponding value. Here we set the parameter of the formation maintenance ε in (6) as $\varepsilon = 0.1$.

As shown in Fig. 6, we found that the robotic swarms maintained their formations, and collisions did not occur during the crossing motion. In fact, the index for formation mix-up, $\| L_i(\boldsymbol{z}_i - \boldsymbol{\sigma}_i^d) \|_\infty$, was less than the value of $\varepsilon = 0.1$; in addition, the minimum distance between robots was larger than 2*r* in Fig. 7(a) and (b). The error norm of the trajectory tracking converged to zero in Fig. 7(c). Thus, the robotic swarm could realize the crossing motion satisfying conditions (6)-(8) by the proposed controller (56).

Next, we show the results of experiment 2. Fig. 8 contains screenshots of the experiment and Fig. 9

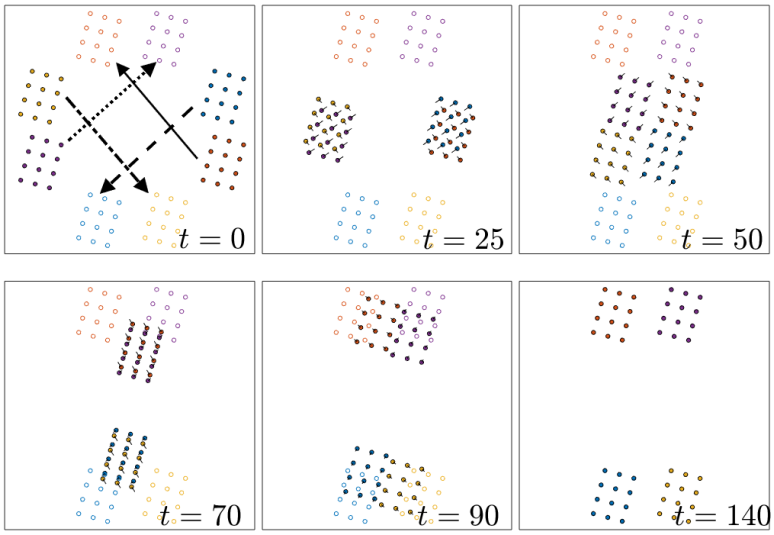
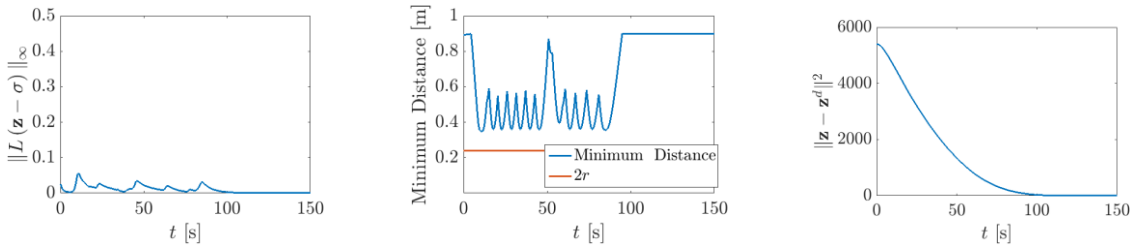


Fig. 10. Screenshots of simulation by proposed method.



(a) The index for formation mix-up (b) The minimum distance (c) The error norm of the trajectory tracking

Fig. 11. Simulation results by proposed method.

shows the experimental results using RVO. In this case, although collision avoidance was realized, we found that formation mix-up occurred in the crossing. As Fig. 9(a) shows, the index for formation mix-up was larger than the value of $\varepsilon = 0.1$. This index value was five times larger than the value of the proposed method, and the formation was not maintained. Through the experiment, we found that the proposed method works well, and realized the well-organized crossing motion satisfying the conditions (6)-(8).

5.2. Numerical Simulations

5.2.1. Numerical Simulation 1

To investigate whether the well-organized crossing motion can be realized for a larger number of robots, we carried out numerical simulations. In the simulation, we used the same parameters as the experiment in Table 1, with four robotic swarms in which each swarm consisted of twelve robots obliquely crossing from the given initial position to the given final positions. Here, the desired values were fixed values, that is, $\dot{\boldsymbol{z}}^d = \boldsymbol{v}^d = 0$.

Fig. 10 shows the screenshots of the simulation by the proposed method. In the figure, the colored

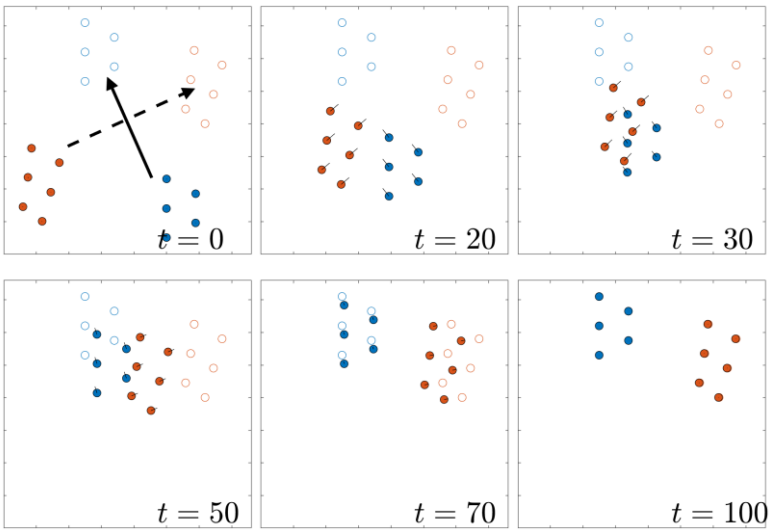
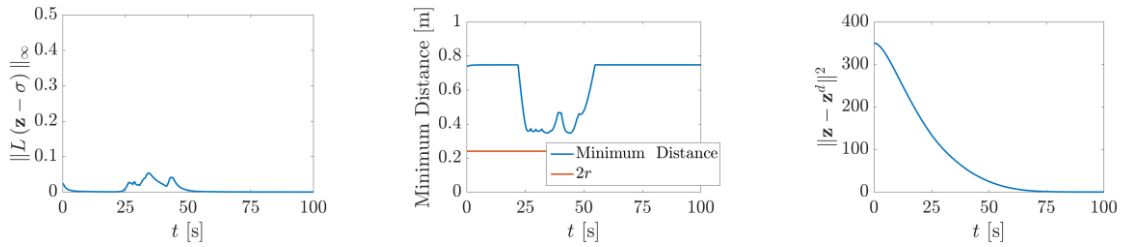


Fig. 12. Screenshots of simulation with asymmetric setup.



(a) The index for formation mix-up (b) The minimum distance (c) The error norm of the trajectory tracking

Fig. 13. Simulation results with asymmetric setup.

circles represent the robots, the open circles show the desired positions of the corresponding robots, and the black line segment at the robots shows the speed of the corresponding robot. In the figure at t = 0, the arrows show the moving direction of each group. In addition, Fig. 11 shows the simulation results for $\| L_i(\boldsymbol{z}_i - \boldsymbol{\sigma}_i^d) \|_\infty$, the minimum distance between robots, and $\| \boldsymbol{z} - \boldsymbol{z}^d \|^2$. From Figs. 10 and 11(a), we found that the robotic swarms maintained their formation, and the index for formation mix-up $\| L_i(\boldsymbol{z}_i - \boldsymbol{\sigma}_i^d) \|_\infty$ was less than the value of $\varepsilon = 0.1$. Further, we found that the minimum distance between robots was larger than 2r from Fig. 11(b), which means that a collision did not occur during the crossing motion. Finally, the error norm of the tracking converged to zero in Fig. 11(c). Thus, the proposed method could realize the crossing motion satisfying conditions (6)-(8), and the well-organized crossing motion under the proposed method can be realized even for a large number of robots.

5.2.2. Numerical Simulation 2

To investigate the validity of the proposed controller for asymmetric setups, including the initial configuration, number of agents, and direction of motion, we carried out a numerical simulation. The parameters used are shown in Table 1. This simulation had two robotic swarms, one with six robots and the other with five. The two swarms crossed from the initial position to the final position, and both the initial positions and motion directions of the swarms were set asymmetrically. Here, the desired values were fixed values; that is, $\dot{\boldsymbol{z}}^d = \boldsymbol{v}^d = 0$.

Fig. 12 shows screenshots of the simulation results. As in the case of Fig. 10, the colored circles indicate the robots, the open circles show the desired positions, and the black line segments show the speed of corresponding robots. In the figure at t = 0, the arrows represent the moving direction, and we can see the asymmetric setups about the initial positions, number of agents, and direction of motion. Fig. 13 shows the simulation results for $\| L_i(\boldsymbol{z}_i - \boldsymbol{\sigma}_i^d) \|_\infty$, the minimum distance between robots, and $\| \boldsymbol{z} - \boldsymbol{z}^d \|^2$. From these figures, we found that the proposed method could realize the crossing motion satisfying conditions (6)-(8); thus, well-organized crossing motion under the proposed method can be realized even for asymmetric setups in the initial positions, number of agents, and direction of motion.

6. Conclusions

This paper proposed a distributed formation controller to realize a well-organized crossing motion for a mass game by robotic swarms. To achieve such crossing motion, we imposed the following three conditions on the motion of the swarms: formation maintenance, collision avoidance, and tracking during motion. To satisfy these conditions, we defined the cost functions and proposed a distributed controller using the gradient of the cost functions. We then proved that the closed-loop system satisfied the three conditions under several assumptions. Finally, we carried out experiments and

numerical simulations. Results showed that the well-organized crossing motion of robotic swarms could be realized under the proposed controller.

When we set a wide distance between robots in the desired formation, as in the experiment and simulations, the deadlock problem of robotic swarms did not occur. However, it is possible that a deadlock could occur if the distance between robots in the desired formation were short. We have not yet investigated the conditions under which deadlock could occur. This will be the next problem we address. On the other hand, the individuals constituting a group in a human mass game have distinct differences. Therefore, it is an interesting problem to consider the formation control for heterogeneous robotic swarms with crossing motion. Here, heterogeneous robotic swarms mean that each agent has different abilities, including translational velocity, sensing distance, etc., as in our previous study [36, 37]. In addition, we did not consider the extension of the proposed controller to the case in which the underlying systems are in an uncertain or time-varying networked environment [38]. From a practical perspective, a distributed controller design for the underlying systems in an uncertain or time-varying networked environment is a useful and challenging topic, and this will be a focus of future research.

Appendix

The symbols used frequently in this paper are listed in Table 2.

REFERENCES

[1] W. Burgard, M. Moors, C. Stachniss, and F.E. Schneider, "Coordinated Multi-robot Exploration," *IEEE Trans. on Robotics*, Vol.21, No.3, pp.376-386, 2005.

[2] J. Clark and R. Fierro, "Cooperative Hybrid Control of Robotic Sensors for Perimeter Detection and Tracking," in *Proc. of American Control Conf.*, pp.3500-3505, 2005.

[3] N. Miyata, J. Ota, T. Arai, and H. Asama, "Cooperative Transport by Multiple Mobile Robots in Unknown Static Environments Associated with Real-time Task Assignment," *IEEE Trans. on Robotics and Automation*, Vol.18, No.5, pp.769-780, 2002.

[4] J. Ota, "Multi-agent robot systems as distributed autonomous systems," *Adv. Eng. Inform.*, Vol.20, No.1, pp. 59-70, 2006.

[5] I. Navarro and F. Matia, "A Survey of Collective Movement of Mobile Robots," *Int. J. Advanced Robotic Syst.*, Vol.10, No.73, pp. 1-9, 2013.

[6] M. Brambilla, E. Ferrante, M. Birattari and M. Dorigo, "Swarm robotics: a review from the swarm engineering perspective," *Swarm Intelligence*, Vol.7, No.1, pp. 1-41, 2013.

[7] L.E. Parker and F. Tang, "Building Multirobot Coalitions through Automated Task Solution Synthesis," *Proceedings of the IEEE*, Vol.94, No.7, pp.1289-1305, 2006.

TABLE 2 SYMBOLS USED IN THIS PAPER

Symbol	Description
m_i	Number of mobile robots in group i
n	Number of the group
$\bar{\boldsymbol{z}}_i$	$\bar{\boldsymbol{z}}_i=[x_i, y_i]^T \in \mathbb{R}^{2\times 1}$ is the representative position of group i
$\boldsymbol{z}_{ij}$	$\boldsymbol{z}_{ij}=[x_{ij}, y_{ij}]^T \in \mathbb{R}^{2\times 1}$ is the position of the j-th robot in group i (robot ij)
$\boldsymbol{v}_{ij}$	$\boldsymbol{v}_{ij}=[\dot{x}_{ij}, \dot{y}_{ij}]^T \in \mathbb{R}^{2\times 1}$ is the velocity of robot ij
$\boldsymbol{u}_{ij}$	$\boldsymbol{u}_{ij}=[u_{ijx}, u_{ijy}]^T \in \mathbb{R}^{2\times 1}$ is the acceleration input of robot ij
r	The size of robot ij fits inside a circle of radius r
$\boldsymbol{\sigma}_{ij}$	The relative position of robot ij from $\bar{\boldsymbol{z}}_i$, (3)
$\bar{\boldsymbol{z}}_i^d$	The desired trajectory of $\bar{\boldsymbol{z}}_i$
$\boldsymbol{z}_{ij}^d$	The desired trajectory of $\boldsymbol{z}_{ij}$
$\boldsymbol{\sigma}_{ij}^d$	The relative desired position of robot ij from $\bar{\boldsymbol{z}}_i^d$, (4)
N_{ij}	The neighbors of robot ij in group i
L_i	$L_i = L_i' \otimes \boldsymbol{I}_2 \in \mathbb{R}^{2m_i\times 2m_i}$, where $L_i' \in \mathbb{R}^{m_i\times m_i}$ is the graph Laplacian defined by (5)
L	$L=\mathrm{diag}(L_1,\cdots,L_n)\in\mathbb{R}^{2m_in\times 2m_in}$
$\boldsymbol{z}_i$, $\boldsymbol{v}_i$, $\boldsymbol{u}_i$	$\boldsymbol{z}_i=[\boldsymbol{z}_{i1}^T,\ldots,\boldsymbol{z}_{im_i}^T]^T\in\mathbb{R}^{2m_i\times 1}$, $\boldsymbol{v}_i=\dot{\boldsymbol{z}}_i\in\mathbb{R}^{2m_i\times 1}$, $\boldsymbol{u}_i=\dot{\boldsymbol{v}}_i\in\mathbb{R}^{2m_i\times 1}$
$\boldsymbol{z}_i^d$, $\boldsymbol{v}_i^d$, $\boldsymbol{u}_i^d$	$\boldsymbol{z}_i^d=[(\boldsymbol{z}_{i1}^d)^T,\ldots,(\boldsymbol{z}_{im_i}^d)^T]^T\in\mathbb{R}^{2m_i\times 1}$, $\boldsymbol{v}_i^d=\dot{\boldsymbol{z}}_i^d\in\mathbb{R}^{2m_i\times 1}$, $\boldsymbol{u}_i^d=\dot{\boldsymbol{v}}_i^d\in\mathbb{R}^{2m_i\times 1}$
$\boldsymbol{\sigma}_i^d$	$\boldsymbol{\sigma}_i^d=[(\boldsymbol{\sigma}_{i1}^d)^T,\ldots,(\boldsymbol{\sigma}_{im_i}^d)^T]^T\in\mathbb{R}^{2m_i\times 1}$
$\bar{m}$	$\bar{m}=\sum_{i=1}^{n}\sum_{j=1}^{m_i}1=m_in$ is the total number of the robots in all groups
$\boldsymbol{z}$, $\boldsymbol{v}$, $\boldsymbol{u}$, $\boldsymbol{x}$	$\boldsymbol{z}=[\boldsymbol{z}_1^T,\ldots,\boldsymbol{z}_n^T]^T\in\mathbb{R}^{2\bar{m}\times 1}$, $\boldsymbol{v}=\dot{\boldsymbol{z}}\in\mathbb{R}^{2\bar{m}\times 1}$, $\boldsymbol{u}=\dot{\boldsymbol{v}}\in\mathbb{R}^{2\bar{m}\times 1}$, $\boldsymbol{x}=[\boldsymbol{z}^T,\boldsymbol{v}^T]^T\in\mathbb{R}^{4\bar{m}\times 1}$
$\boldsymbol{z}^d$, $\boldsymbol{v}^d$, $\boldsymbol{u}^d$	$\boldsymbol{z}^d=[(\boldsymbol{z}_1^d)^T,\ldots,(\boldsymbol{z}_n^d)^T]^T\in\mathbb{R}^{2\bar{m}\times 1}$, $\boldsymbol{v}^d=[(\boldsymbol{v}_1^d)^T,\ldots,(\boldsymbol{v}_n^d)^T]^T\in\mathbb{R}^{2\bar{m}\times 1}$, $\boldsymbol{u}^d=[(\boldsymbol{u}_1^d)^T,\ldots,(\boldsymbol{u}_n^d)^T]^T\in\mathbb{R}^{2\bar{m}\times 1}$
$\hat{\boldsymbol{z}}$, $\hat{\boldsymbol{v}}$, $\hat{\boldsymbol{u}}$, $\hat{\boldsymbol{x}}$	$\hat{\boldsymbol{z}}=\boldsymbol{z}-\boldsymbol{z}^d\in\mathbb{R}^{2\bar{m}\times 1}$, $\hat{\boldsymbol{v}}=\boldsymbol{v}-\boldsymbol{v}^d\in\mathbb{R}^{2\bar{m}\times 1}$, $\hat{\boldsymbol{u}}=\boldsymbol{u}-\boldsymbol{u}^d\in\mathbb{R}^{2\bar{m}\times 1}$, $\hat{\boldsymbol{x}}=[\hat{\boldsymbol{z}}^T,\hat{\boldsymbol{v}}^T]^T\in\mathbb{R}^{4\bar{m}\times 1}$

[8] C.W. Reynolds, "Flocks, Herds, and Schools: Adistributed Behavioral Model," *Computer Graphics*, Vol.21, No.4, pp.25-34, 1987.

[9] A. Jadbabaie, J. Lin, and A.S. Morse, "Coordination of Groups of Mobile Autonomous Agents Using Nearest Neighbor Rules," *IEEE Trans. Autom. Control*, Vol.48, No.56, pp.988-1001, 2003.

[10] R. Olfati-Saber, "Flocking for Multi-agent Dynamic Systems: Algorithms and Theory," *IEEE Trans. on Automatic Control*, Vol.51, No.3, pp.401-420, 2006.

[11] F. Bullo, J. Cortes, and S. Martinez, *Distributed Control of Robotic Networks: A Mathematical Approach to Motion Coordination Algorithms*, Princeton Univ. Press, 2009.

[12] M. Mesbahi, and M. Egerstedt, *Graph Theoretic Methods in Multiagent Networks*, Princeton Univ. Press, 2010.

[13] W. Ren, R.W. Beard, and E.M. Atkins, “A survey of consensus problems in multi-agent coordination,” in *Proc. of American Control Conf.*, pp.1859-1864, 2005.

[14] B.A. Francis, and M. Maggiore, *Flocking and Rendezvous in Distributed Robotics*, Springer, 2016.

[15] S. Izumi, S. Azuma, and T. Sugie, “Distributed Hybrid Controllers for Multi-Agent Mass Games by a Variable Number of Player Agents,” *Asian J. Control*, Vol.17, No.3, pp.762-774, 2015.

[16] J. Alonso-Mora, A. Breitenmoser, M. Rufli, R. Siegwart, and P. Beardsley, “Image and Animation Display with Multiple Mobile Robot,” *The Int. Journal of Robotics Research*, Vol.31, No.6, pp.753-773, 2012.

[17] K. Yamane and J. Goerner, “Task Assignment and Trajectory Optimization for Displaying Stick Figure Animations with Multiple Mobile Robots,” in *Proc. of IROS*, pp.3806-3813, 2014.

[18] M. Rubenstein, A. Cornejo, and R. Nagpal, “Programmable Self-assembly in A Thousand-robot Swarm,” *Science*, Vol.345, No.6198, pp.795-799, 2014.

[19] J. van den Berg, M. Lin, and D. Manocha, “Reciprocal Velocity Obstacles for Real-time Multi-agent Navigation,” in *Proc. of ICRA*, pp.1928-1935, 2008.

[20] J. van den Berg, S. J. Guy, M. Lin, and D Manocha, “Reciprocal n-body Collision Avoidance,” in *Robotics Research: The 14th International Symposium ISRR*, C. Pradalier, R. Siegwart, and G. Hirzinger, eds., Springer-Verlag, pp.3-19, 2011.

[21] J. Alonso-Mora, T. Naegeli, R. Siegwart and P. Beardsley, “Collision Avoidance for Aerial Vehicles in Multi-agent Scenarios,” *Autonomous Robots*, Vol.39, No.1, pp.101-121, 2015.

[22] M. Xu, Y. Wu, Y. Ye, I. Farkas, H. Jiang, and Z. Deng, “Collective Crowd Formation Transform with Mutual Information-Based Runtime Feedback,” *Computer Graphics Forum*, Vol.34, No.1, pp.60-73, 2015.

[23] Y.Q. Chen, and Z. Wang, “Formation Control: A Review and A New Consideration”, in *Proc. of IEEE/RSJ IROS 2005*, pp.3181-3186, 2005.

[24] Y. Cao, W. Yu, W. Ren, and G. Chen, “An Overview of Recent Progress in the Study of Distributed Multi-Agent Coordination”, *IEEE Trans. on Industrial Informatics*, Vol.9, No.1, pp.427-438, 2013.

[25] K.-K. Oh, M.-C. Park, and H.-S. Ahn, “A survey of multi-agent formation control”, *Automatica*, Vol.53, pp.424-440, 2015.

[26] S. Knorn, Z. Chen, and R. H. Middleton, “Overview: Collective Control of Multiagent Systems”, *IEEE Trans. on Control of Network Systems*, Vol.3, No.4, pp.334-347, 2016.

[27] N. E. Leonard and E. Fiorelli, “Virtual leaders, artificial potentials and coordinated control of groups”, in *Proc. of IEEE CDC 2001*, pp.2968-2973, 2001.

[28] H. G. Tanner, A. Kumar, “Towards Decentralization of Multi-robot Navigation Functions”, in *Proc. of IEEE ICRA 2005*, pp.4132-4137, 2005.

[29] D. V. Dimarogonas, K. J. Kyriakopoulos, “Connectedness Preserving Distributed Swarm Aggregation for Multiple Kinematic Robots”, *IEEE Trans. on Robotics*, Vol.24, No.5, pp.1213-1223, 2008.

[30] E. G. Hernández-Martínez, and E. Aranda-Bricaire, “Convergence and Collision Avoidance in Formation Control: A Survey of the Artificial Potential Functions Approach”, In: F. Alkhateeb (Ed.), *Multi-Agent Systems - Modeling, Control, Programming, Simulations and Applications*, InTech, 2011.

[31] D. Panagou, “A Distributed Feedback Motion Planning Protocol for Multiple Unicycle Agents of Different Classes”, *IEEE Trans. on Automatic Control*, Vol.62, No.3, pp.1178-1193, 2017.

[32] Y. Kobayashi, M. Yoshimoto, K. Kon, H. Fukushima, F. Matsuno, T. Morii, M. Kitagawa, S. Tsuji, and K. Yoshikawa, “Formation Control Considering Well-organized Crossing of Robot Swarms,” in *Proc. of SWARM 2015*, pp.465-466, 2015.

[33] J. Wang, and M. Xin, “Integrated Optimal Formation Control of Multiple Unmanned Aerial Vehicles,” *IEEE Trans. on Control System Technology*, Vol.21, No.5, pp.1731-1744, 2013.

[34] R. Falconi, L. Sabattini, C. Secchi, C. Fantuzzi, and C. Melchiorri, “Edge-weighted Consensus-based Formation Control Strategy with Collision Avoidance,” *Robotica*, Vol.33, pp.332-347, 2014.

[35] C. Godsil, and G. Royle, *Algebraic Graph Theory*, Springer, 2001.

[36] R. Maeda, T. Endo, and F. Matsuno, “Decentralized Navigation for Heterogeneous Swarm Robots With Limited Field of View”, *IEEE Robotics and Automation Letters*, Vol.2, No.2, pp.904-911, 2017.

[37] M. Yoshimoto, T. Endo, R. Maeda, and F. Matsuno, “Decentralized navigation method for a robotic swarm with nonhomogeneous abilities”, *Auton Robot*, (to be published)

[38] L. Zhang, H. Gao, and O. Kaynak, “Network-Induced Constraints in Networked Control Systems—A Survey”, *IEEE Trans. on Industrial Informatics*, Vol.9, No.1, pp.403-416, 2013.