Leader–Follower Navigation in Obstacle Environments While Preserving Connectivity Without Data Transmission

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

***Abstract*— In this paper, we propose a control method for leader–follower navigation in obstacle environments while preserving sensing network connectivity without data transmission between robots. Unlike most connectivity-preserving algorithms, the control input is determined in such a way as to not only guarantee connectivity preservation and collision avoidance, but also to ensure that input constraints are not violated at each time step. We also introduce a simple rule for changing network topology depending on environments such that some sensing links are deactivated in order to pass through narrow spaces, while active links are increased in free spaces to keep the group as cohesive as possible. The effectiveness of the proposed method is demonstrated in simulations and experiments.**

***Index Terms*— Collision avoidance, connectivity maintenance, leader–follower navigation, line of sight (LOS), obstacle environment.**

I. INTRODUCTION

THE cooperative control of multiple mobile robots has been intensively studied for potential applications, such as exploration, surveillance, mapping of unknown environments, and the transport of large objects (see [1]–[3] for an overview). This paper focuses on the fundamental problem of how to move a group of robots as a whole to a target area. Specifically, we assume that only one of the robots, called the leader, knows the path to the target area. Thus, since each robot in the group has a limited sensing and communication range, the connectivity of the sensing/communication network must be preserved in order to avoid leaving some robots behind. Furthermore, we aim to derive an algorithm without relying on data transmission between robots, in order to deal with environments where there is no wireless network available for such information exchange.

Various methods for controlling multiagent systems while preserving network connectivity have been proposed [4]–[24], as detailed in Section II (see also [25] for an overview). However, most of the previous methods have had at least one of the following limitations.

1) An obstacle-free environment is considered [4]–[20].
2) Data transmission between robots through a wireless network is required to estimate network connectivity [4]–[12], [22].
3) It is difficult to explicitly consider input constraints [4]–[17], [22]–[24].
4) A fixed network topology must be preserved or only links can be added [13]–[17], [23], [24].
5) Interrobot collision is not considered [18]–[21].

In this paper, we propose a control method for leader–follower navigation in obstacle environments while preserving sensing network connectivity without data transmission between robots. Unlike most connectivity-preserving algorithms, the control input is determined so as not only to guarantee connectivity preservation and collision avoidance, but also to ensure that a given input constraint is not violated at each time step. Although an input constraint is considered in the connectivity-preservation algorithm for the leader–follower navigation proposed in [20], it has limitations in that an obstacle-free environment is assumed and that collision avoidance is not guaranteed. In obstacle environments, the proposed method manages control input so as to preserve line-of-sight (LOS) visibility between neighbors as well as a maximum distance constraint. We also derive conditions for collision avoidance not only with robots that are visible at the current sampling step, but also with those that are not visible, e.g., due to an obstacle. Another key issue for leader–follower navigation in obstacle environments is how to move through narrow spaces without getting stuck. We introduce a simple rule to change network topology depending on environments in such a way that some sensing links are deactivated in order to pass through narrow spaces, while active links are increased in free spaces to keep the group as cohesive as possible. Furthermore, unlike many other studies including [20], the effectiveness of the algorithm is demonstrated not only with simulations, but also in real robot experiments.

II. RELATED WORKS

In this section, we review previous studies on network connectivity preservation in multiagent systems. Although many control methods have been developed by assuming network

Manuscript received December 25, 2016; accepted April 17, 2017. Date of publication May 26, 2017; date of current version June 11, 2018. Manuscript received in final form May 8, 2017. This work was supported by JSPS KAKENHI under Grant 15K06137. Recommended by Associate Editor A. G. Aghdam. *(Corresponding author: Hiroaki Fukushima.)*

D. Sakai is with Hitachi Construction Machinery Co., Ltd., Tsuchiura 300-0013, Japan (e-mail: sakaidaito@gmail.com).

H. Fukushima and F. Matsuno are with the Department of Mechanical Engineering and Science, Graduate School of Engineering, Kyoto University, Kyoto 615-8540, Japan (e-mail: fuku@me.kyoto-u.ac.jp; matsuno@me.kyoto-u.ac.jp).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCST.2017.2705121

connectivity (see [26]–[38]), we do not focus on these in this paper.

Many studies use the Fiedler value [39], which is the second smallest eigenvalue of the Laplacian matrix of the graph describing the network, as a metric for overall network connectivity. If the Fiedler value of a graph is positive, the connectivity of the graph is guaranteed. Some early studies presented centralized algorithms to increase the Fiedler value of graphs. Kim and Mesbahi [4] proposed an iterative semidefinite programming-based approach to maximizing the Fiedler value of the graph, while Zavlanos and Papas [5] reported an artificial potential function-based approach to keeping the Fiedler value positive.

One way to decentralize such algorithms is to make use of a decentralized estimation method of the connectivity [6]–[8], in which each agent estimates the eigenvalues (or eigenvectors) of the Laplacian matrix using information received from neighbors. Based on the connectivity estimation, a gradient controller to maximize the Fiedler value was designed in [6] and [7], while an artificial potential function to keep the Fiedler value positive was used in [9], [10], and [22].

Although these algorithms based on the Fiedler value are theoretically sophisticated, it is difficult to apply them to the control problem described in Section I, since they require data transmission between robots (or between a central computer and each robot in the centralized algorithms [4], [5]), through wireless communication to exchange information on the Laplacian matrix. Another limitation is the difficulty in explicitly considering an input constraint. Furthermore, poor estimation of connectivity could lead to violation of network connectivity. To address this issue, Sabattini *et al.* [9] proved the boundedness of estimation errors of the Fiedler value, and suggested taking an estimation error bound into account in the design of the artificial potential function. However, it is not clear how to obtain such an error bound that is not too conservative to apply to the artificial potential design.

In [11], an auction algorithm was implemented to decide cooperatively whether or not a link could be deactivated without violating network connectivity. Zavlanos *et al.* [12] proposed a flocking algorithm that achieved velocity synchronization of agents in a network while preserving connectivity using a method similar to that in [11]. However, the auction algorithm required data transmission between robots to share information on bids from other robots.

On the other hand, network connectivity preservation algorithms without data transmission between robots have also been studied. In [13]–[17] and [23], artificial potential functions were used to preserve the initial network topology, which was assumed to be connected. In particular, connectivity preservation in obstacle environments was considered in [23]. Since these methods do not allow for the deactivation of any sensing link, it is difficult for robots to pass through narrow spaces if their initial network topology has many redundant links to make the group cohesive. Another limitation of these methods is that it is difficult to explicitly consider an input constraint. On the other hand, in [18]–[21], a given input constraint can be explicitly considered in the control design.

In [18], [19], and [21], multirobot rendezvous algorithms while preserving connectivity are proposed. In particular, obstacle environments are considered in [21]. With these algorithms, each robot computes at each time a convex region, called a constraint set, in which the robot is constrained to move within this region to preserve sensing links with neighbors. The robots then move toward the circumcenter of their constraint set. The amount of movement is determined such that no given input constraint is violated. However, moving toward the circumcenter of the constraint set in order to gather at the same location is not necessarily appropriate in the leader–follower navigation problem considered here. Another difference between our proposed control method and these rendezvous algorithms is that we consider interrobot collisions. Thus, we need to consider the problem of how to deactivate sensing links in order for robots to pass through narrow spaces without getting stuck; this does not arise if interrobot collisions are ignored. Furthermore, our method does not need to compute a constrained set to determine control inputs, which results in decreased computation time.

The leader–follower navigation algorithm in [20] decides the direction of movement using an artificial potential function, then the amount of movement is determined taking into account the input constraint and network connectivity. We also use this basic procedure in our proposed method. However, [20] did not consider obstacle environments and interrobot collisions. To overcome this limitation, we derive additional constraints on the amount of movement so as to achieve LOS visibility preservation, obstacle avoidance, and interrobot collision avoidance. Furthermore, as already mentioned, we introduce a link deactivation rule in order for robots to pass through narrow spaces without getting stuck.

Panagou and Kumar [24] proposed a leader–follower navigation method in obstacle environments where the network topology was fixed to a chain formation. As the distance between robots was controlled to a given constant value, their multirobot system can be regarded as a tractor–trailer system. A limitation of this method is that as the number of robots grows, the turning radius of the leader must be increased, and a wider path is required. Furthermore, tracking error from a target relative position is not guaranteed to converge to zero, and the estimation of the error bound is difficult especially in the case of multiple followers. If the tracking error is large, connectivity maintenance and collision avoidance might not be achieved.

III. Problem Setting

We consider N robots in a 2-D work space with obstacles. The movement of the ith robot ($i = 1, 2, \ldots, N$) is described as the following discrete-time system:

$$x_i(k+1) = x_i(k) + u_i(k), \quad \|u_i(k)\| \le u_{\max} \tag{1}$$

where $x_i(k)$ and $u_i(k)$ are the position and control input of robot i, respectively, at time step k $(=0, 1, \ldots)$. The input limit $u_{\max}$ in (1) is given by taking into account the hardware limitations and length of the sampling interval. We assume that a sufficient condition for collision avoidance between robots i and j is given as follows:

$$\|x_i - x_j\| \geq d_c \quad \forall j \in \mathcal{V} \setminus \{i\} \tag{2}$$

where $\mathcal{V} := \{1, 2, \ldots, N\}$ is the set of indices of all the robots. Although a robot is modeled as a point in (1), d_c in (2) should be determined by taking into account the size of the actual robots. We also assume that a sufficient condition for obstacle avoidance is given as

$$\|x_i - x_o\| \geq d_o \quad \forall x_o \in \mathcal{O} \tag{3}$$

where $\mathcal{O}$ is a set of all points on obstacles in the workspace.

In order to describe the sensing model, we first define the line segment joining p and q as

$$\mathcal{L}(p, q) := \{(1 - \lambda)p + \lambda q, \forall \lambda \in [0, 1]\}. \tag{4}$$

Furthermore, we define

$$\mathcal{L}_{ij}(k) := \mathcal{L}(x_i(k), x_j(k)). \tag{5}$$

Then, we assume that robot i is able to sense the relative position of robot $j \in \mathcal{V} \setminus \{i\}$

$$x_{ji}(k) := x_j(k) - x_i(k) \tag{6}$$

if the following conditions are satisfied:

$$\|x_j - x_i\| \leq d_s \tag{7}$$

$$\|q - x_o\| \geq d_l \quad \forall q \in \mathcal{L}_{ij} \quad \forall x_o \in \mathcal{O}. \tag{8}$$

The condition in (7) implies that the maximum sensing range is given by a positive number d_s. The condition in (8) implies that the distance from $\mathcal{L}_{ij}$ to each obstacle is not less than a minimum clearance d_l, so that the LOS between robots i and j is not interrupted by obstacles.

It is also assumed that robot i is able to detect a point on an obstacle $x_o \in \mathcal{O}$, if

$$\|x_o - x_i\| \leq d_s \tag{9}$$

$$\|x_o - x_i\| \leq \|q - x_i\| \quad \forall q \in \bar{\mathcal{L}}(x_i, x_o) \cap \mathcal{O} \tag{10}$$

where $\bar{\mathcal{L}}(x_i, x_o) := \{(1 - \lambda)x_i + \lambda x_o, \forall \lambda \geq 0\}$. While the set $\mathcal{L}(x_i, x_o)$ only includes points between x_i and x_o, the set $\bar{\mathcal{L}}(x_i, x_o)$ includes points behind x_o along the line from x_i to x_o, in addition to the points in $\mathcal{L}(x_i, x_o)$. Thus, the condition in (10) implies that there is no other obstacle point closer to x_i than x_o on $\mathcal{L}(x_i, x_o)$. We denote $\mathcal{O}_i(k)$ as the set of points on obstacles detected by robot i at time k.

In terms of the sensing mentioned above, we represent the network topology of the multirobot system using a graph $\mathcal{G}_s(x(k)) = (\mathcal{V}, \mathcal{E}_s(x(k)))$ where $x = [x_1, x_2, \ldots, x_N]$. We denote $\mathcal{V}$ and $\mathcal{E}_s(x(k))$ as the node set and edge set, respectively. The elements of $\mathcal{E}_s(x(k))$ are pairs of robot indices that are able to sense each other's position at time k. The graph $\mathcal{G}_s$ is said to be connected, if for every pair of nodes, there exists a path from one node to the other.

We assume that a target path is given to only one of the N robots, called the leader, whose index is set as N without loss of generality. Other robots i $(= 1, 2, \ldots, N-1)$ are called followers. It is assumed that each follower is not able to recognize whether or not another robot is the leader. If $\mathcal{G}_s$ is connected, there is a path between the leader and each follower. Thus, since the maximum length of each link of $\mathcal{G}_s$ is kept to no more than a given finite value d_s, each follower is forced to follow the leader at a certain distance [at most $(N-1)d_s$]. Furthermore, if the length of the link is kept to less than d_s, the distance between the leader and a follower can be decreased. Thus in this paper, we aim to preserve the connectivity of the following subgraph of $\mathcal{G}_s(x(k))$:

$$\begin{aligned} &\mathcal{G}_n(x(k)) := (\mathcal{V}, \mathcal{E}_n(x(k))) \\ &\mathcal{E}_n(x(k)) := \{(i, j) \in \mathcal{E}_s(x(k)) \mid \|x_i(k) - x_j(k)\| \leq d_n\} \end{aligned} \tag{11}$$

where $d_c < d_n < d_s$. Then, the set of neighbors of robot i is defined as

$$\mathcal{N}_i(x(k)) := \{j \mid (i, j) \in \mathcal{E}_n(x(k))\}. \tag{12}$$

From the definition of $\mathcal{G}_n(x(k))$, the connectivity of $\mathcal{G}_s$ is preserved if that of $\mathcal{G}_n(x(k))$ is preserved. As clarified in Theorem 2 in Section IV-C, it is required that $(d_o^2 + d_n^2)^{1/2} \leq d_s$ holds to guarantee the preservation of $\mathcal{G}_n$. In other words, if $d_n = d_s$, i.e., $\mathcal{G}_n = \mathcal{G}_s$, it is difficult to guarantee the preservation of $\mathcal{G}_s$, which is another reason why we introduce $\mathcal{G}_n$ in addition to $\mathcal{G}_s$.

We assume that $\mathcal{G}_n$ is connected at the initial time $k = 0$. Thus, the simplest way to preserve connectivity is to control the robots such that the edges of $\mathcal{G}_n$ at $k = 0$ are not lost. However, in obstacle environments, it is often necessary to change the network topology appropriately to navigate through a narrow space. Thus, it is necessary to select edges to be maintained at each time step, such that the connectivity of $\mathcal{G}_n$ is preserved at the next time step. To describe the edges to be preserved, we define the symmetric indicator function $\sigma_{ij}(k) = \sigma_{ji}(k) \in \{0, 1\}$. If $\sigma_{ij} = 1$, there will be an effort to preserve the edge (i, j). In other words, robot i aims to preserve the link to robot j in the following set:

$$\mathcal{N}_i^{\sigma}(x(k)) := \{j \in \mathcal{N}_i(k) \mid \sigma_{ij}(k) = 1\}. \tag{13}$$

We also define the following subgraph of $\mathcal{G}_n(x(k))$:

$$\mathcal{G}_{\sigma}(x(k)) := (\mathcal{V}, \mathcal{E}_{\sigma}(x(k))) \tag{14}$$

$$\mathcal{E}_{\sigma}(x(k)) := \{(i, j) \in \mathcal{E}_n(x(k)) \mid \sigma_{ij}(k) = 1\}. \tag{15}$$

If σ_{ij} is determined such that $\mathcal{G}_{\sigma}$ is connected, the connectivity of $\mathcal{G}_n$ is preserved by moving the robots in such a way that no edge of $\mathcal{G}_{\sigma}$ is lost.

In this paper, we propose an algorithm to determine u_i so as to preserve the connectivity of $\mathcal{G}_n$, while satisfying the collision avoidance conditions in (2) and (3). The proposed method aims to preserve these properties not only at $x_i(k)$ $(k = 0, 1, \ldots)$ but also at each point on the line segments $\mathcal{L}(x_i(k), x_i(k+1))$. It is reasonable to consider a path composed of line segments in the sense that a robot described by a commonly used continuous time model $\dot{x}_i(t) = u_i(t)$ follows such a path when the control input $u_i(t)$ is fixed between discrete time steps. Furthermore, the proposed algorithm aims to determine σ_{ij} in order to avoid deadlock in narrow spaces, while increasing active links to keep the group as cohesive as possible if there are no obstacles around the robots.

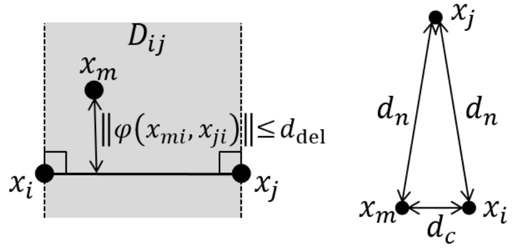

Fig. 1. Conditions for $(i, j, m) \in \mathcal{T}$ (left) and $(i, j, m) \in \overline{\mathcal{T}}$ (right).

Remark 1: In order to keep the notation as simple as possible, the problem setting in this section and control algorithm in the next section are described by the global coordinates. However, for implementation, each robot determines the control input vector using local coordinates. Global coordinates, such as x_i, x_j, and x_o, are replaced by local coordinates ${}^i x_i$, ${}^i x_j$, and ${}^i x_o$ in the control algorithm, where ${}^i x_i = 0$, since the origin of robot i's local frame is located at x_i. The control input ${}^i u_i$ with respect to the local frame is obtained by the control algorithm using the local coordinates. By applying ${}^i u_i$ to the robot, the global input vector $u_i = {}^0 R_i {}^i u_i$ is applied in (1), where ${}^0 R_i \in SO(2)$ denotes the orientation of robot i's local frame with respect to the global frame. Therefore, the proposed algorithm can be implemented using only local information.

IV. Control Algorithm

The outline of the control algorithm for robot i at time k is described as follows.

Step 1: According to the sensing information on the relative position of robot $j \in \mathcal{N}_i$, the indicator function $\sigma_{ij}(k)$ is determined, such that $\mathcal{G}_\sigma(x(k))$ is connected.

Step 2: The direction of the control input vector $u_i(k)$ is determined based on an artificial potential function.

Step 3: The magnitude of $u_i(k)$, which guarantees the edge preservation of $\mathcal{G}_\sigma(x(k))$ and collision avoidance at each point on $\mathcal{L}(x_i(k), x_i(k+1))$, is determined by taking into account the given input constraint in (1).

In Sections IV-A–IV-C, we describe the details of each step.

A. Link Deactivation for Navigation in Narrow Spaces

By link deactivation, we mean deciding the edges of $\mathcal{G}_n$ that will not be preserved. More precisely, we obtain the set $\mathcal{E}_\sigma$ of preserved edges by removing some edges from $\mathcal{E}_n$. There are two cases where an edge in $\mathcal{E}_n$ is not included in $\mathcal{E}_\sigma$. To describe the first case, the following region is defined for given x_i, x_j, and x_{ji} in (6):

$$D_{ij} := \{q \mid (q - x_i)^T x_{ji} > 0,\ (q - x_j)^T x_{ji} < 0\} \quad (16)$$

as shown in Fig. 1 (left). We also define

$$\varphi(p, q) := \frac{p^T H q}{\|Hq\|^2} Hq, \quad H := \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \quad (17)$$

which is the projection of a vector p to the line orthogonal to q. Thus, as shown in Fig. 1 (left), $\|\varphi(x_{mi}, x_{ji})\|$ is the distance from x_m to the line including x_i and x_j. Then, we define $\mathcal{T}$ as the set of triples of robots (i, j, m) that satisfies

$$\|\varphi(x_{mi}, x_{ji})\| \leq d_{\rm del}, \quad x_m \in D_{ij}, \quad \sin\alpha_m > 0 \quad (18)$$
$$(i, j) \in \mathcal{E}_n, \quad (j, m) \in \mathcal{E}_n, \quad (m, i) \in \mathcal{E}_n \quad (19)$$

where $d_{\rm del}$ is a positive constant satisfying

$$d_{\rm del} < d_c \sin\frac{\pi}{3}. \quad (20)$$

In (18), $\alpha_m \in (-\pi, \pi]$ denotes the angle from the vector x_{im} to x_{jm} measured in the counterclockwise direction. The condition $\sin\alpha_m > 0$ implies that the triple vertices in $\mathcal{T}$ are ordered in a counterclockwise direction, as shown in Fig. 1 (left). Then, robot i does not include the edge $(i, j) \in \mathcal{E}_n$ in $\mathcal{E}_\sigma$ if

$$(i, j, m) \in \mathcal{T} \ \text{ or } \ (j, i, m) \in \mathcal{T}, \quad \exists m \in \mathcal{V} \setminus \{i, j\}. \quad (21)$$

If the condition in (20) and the collision avoidance condition in (2) are satisfied, it is guaranteed that the links (j, m) and (m, i) of the robots (i, j, m) satisfying (21) will not be deactivated, as shown in Lemma 3 in Appendix A. Therefore, the network connectivity is preserved at least if $N = 3$, i.e., if there are no robots other than (i, j, m). Furthermore, Theorem 1 discusses connectivity in the case of $N > 3$. On the other hand, if (20) is violated, the preservation of connectivity is not guaranteed even if $N = 3$. For example, if $d_{\rm del} = d_c \sin(\pi/3)$, all the links among the robot triple (i, j, m) will be deactivated, when their positions constitute an equilateral triangle with edge length d_c.

To describe the other case where an edge is removed from $\mathcal{E}_\sigma$, we define $\overline{\mathcal{T}}$ as the set of robot triples (i, j, m) that satisfy

$$\|x_{ij}\| = \|x_{jm}\| = d_n, \quad \|x_{mi}\| = d_c, \quad \sin\alpha_m > 0 \quad (22)$$

and (19). As shown in Fig. 1 (right), a robot triple $(i, j, m) \in \overline{\mathcal{T}}$ forms an isosceles triangle with two edges of length d_n and one edge of length d_c. In the edge deletion rule in this paper, robot i does not include the edge $(i, j) \in \mathcal{E}_n$ in $\mathcal{E}_\sigma$ if

$$(i, j, m) \in \overline{\mathcal{T}} \ \text{ or } \ (j, i, m) \in \overline{\mathcal{T}}, \quad \exists m \in \mathcal{V} \setminus \{i, j\}. \quad (23)$$

In summary, the rule to decide σ_{ij} is described as follows:

$$\sigma_{ij}(k) = \begin{cases} 0, & \text{if (21) or (23)} \\ 1, & \text{otherwise.} \end{cases} \quad (24)$$

This rule is decentralized and does not require data transmission between robots. We obtain the following result on the connectivity of $\mathcal{G}_\sigma$ when multiple links are deleted at the same time by the rule in (24).

Theorem 1: Suppose that the following assumptions hold: 1) $\mathcal{G}_n(x(k))$ is connected; 2) all robots satisfy the collision avoidance conditions in (2) and (3) at time k; 3) the condition in (20) is satisfied; and 4) π is not an integer multiple of $\sin^{-1}(d_c/2d_n)$. Then, $\mathcal{G}_\sigma(x(k))$ obtained by the rule in (24) is connected.

Proof: See Appendix A. ∎

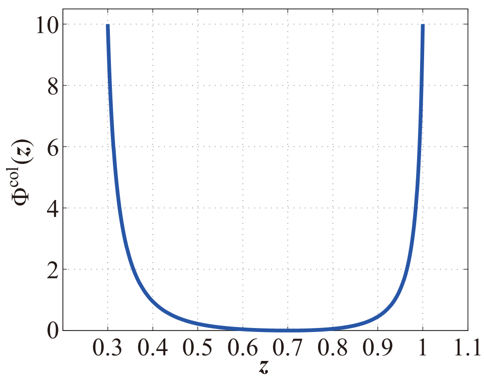

Fig. 2. Example of $\Phi^{\rm col}(z)$ ($d_c = 0.3, d_r = 0.7, and d_n = 1.0$).

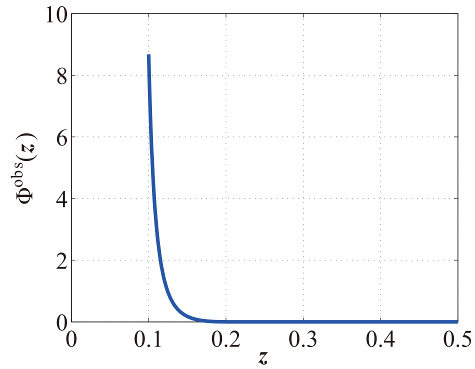

Fig. 3. Example of $\Phi^{\rm obs}(z)$ ($d_o = 0.1$ $and d_{\rm or} = 0.2$).

Remark 2: Since the equalities in (22) do not hold exactly in practice, we instead check if the following inequalities:

$$d_n - \epsilon_n \le \|x_{ij}\| \le d_n, \quad d_n - \epsilon_n \le \|x_{jm}\| \le d_n$$
$$d_c \le \|x_{mi}\| \le d_c + \epsilon_c$$

are satisfied for small positive constants ϵ_n and ϵ_c.

B. Direction of Control Input Vector

For the leader ($i = N$), the direction of $u_i(k)$, which is equivalent to the robot's direction of movement, is given from the target path in the same way as described in [20]. For the followers, on the other hand, the direction of the movement is decided by

$$v_i = -\nabla_{x_i}\Psi_i(x), \quad i \in \mathcal{V}\setminus\{N\} \tag{25}$$

where $\Psi_i(x)$ is the artificial potential function defined in the following. Note that v_i in (25) can be computed using only local information, since $\Psi_i(x)$ depends only on a part of the elements of x, i.e., x_i and x_j ($j \in \mathcal{N}_i^\sigma$).

The artificial potential function $\Psi_i(x)$ is a weighted sum of $\Psi_i^{\rm col}$, $\Psi_i^{\rm obs}$, $\Psi_i^{\rm los}$, and $\Psi_i^{\rm coh}$ as follows:

$$\Psi_i := c_1\Psi_i^{\rm col} + c_2\Psi_i^{\rm obs} + c_3\Psi_i^{\rm los} + c_4\Psi_i^{\rm coh}. \tag{26}$$

The first component $\Psi_i^{\rm col}$ takes into account the desired value and constraints on the relative distance to neighbors $j \in \mathcal{N}_i^\sigma$. Artificial potential functions for this purpose are available in the literature. In this paper, we adopt a similar function to that in [17], as follows:

$$\Psi_i^{\rm col}(x) = \sum_{j\in\mathcal{N}_i^\sigma} \Phi^{\rm col}(\|x_i - x_j\|)$$
$$\Phi^{\rm col}(z) := \frac{(z-d_r)^2(d_n-z)}{(d_n-d_c)^2(z-d_c)+(d_r-d_c)^2(d_n-z)/\kappa_1} + \frac{(z-d_c)(z-d_r)^2}{(d_n-d_c)^2(d_n-z)+(z-d_c)(d_n-d_r)^2/\kappa_2} \tag{27}$$

where κ_1 and κ_2 are design parameters whose values are equivalent to $\Phi^{\rm col}(z)$ at $z = d_c$ and $z = d_n$, respectively. Fig. 2 shows an example of $\Phi^{\rm col}(z)$ for $d_c = 0.3$, $d_r = 0.7$, $d_n = 1.0$, and $\kappa_1 = \kappa_2 = 10$. As shown in this example, $\Phi^{\rm col}(z)$ has the minimum value at the desired relative distance d_r, and monotonically increases as z goes to the maximum allowable distance d_n, or to the minimum allowable distance d_c.

The second component $\Psi_i^{\rm obs}(x)$ is introduced for obstacle avoidance. More precisely, $\Psi_i^{\rm obs}(x)$ is decided in such a way that $-\nabla_{x_i}\Psi_i^{\rm obs}(x)$ is in the direction away from the closest obstacle point detected at time k. Such an obstacle point is defined as

$$o_i^{\rm obs} = \mathop{\arg\min}_{x_o\in\mathcal{O}_i(k)} \|x_o - x_i(k)\|. \tag{28}$$

By using $o_i^{\rm obs}$ mentioned previously, $\Psi_i^{\rm obs}(x)$ is described as

$$\Psi_i^{\rm obs}(x) = \Phi^{\rm obs}(\|x_i - o_i^{\rm obs}\|)$$
$$\Phi^{\rm obs}(z) = \begin{cases} \frac{1}{2}\left(\left(\frac{z-d_o}{d_{\rm or}-d_o}+\delta\right)^{-1} - \frac{1}{1+\delta}\right)^2, & \text{if } z < d_{\rm or} \\ 0, & \text{otherwise} \end{cases}$$

where $d_{\rm or} > d_o$ is a design parameter, and δ is a small number to keep $\Phi^{\rm obs}(z)$ finite. Fig. 3 presents an example of $\Phi^{\rm obs}(z)$ for $d_o = 0.1$ and $d_{\rm or} = 0.2$. As shown in this example, $\Phi^{\rm obs}(z)$ monotonically increases as z goes from $d_{\rm or}$ to the minimum allowable distance d_o to obstacles.

The third component $\Psi_i^{\rm los}(x)$ is decided in such a way that $-\nabla_{x_i}\Psi_i^{\rm los}(x)$ is in the direction moving away from the closest point to $\mathcal{L}_{ij}(k)$ among the obstacle points in D_{ij} detected at time k. Such an obstacle point is defined as

$$o_{ij}^{\rm los} = \mathop{\arg\min}_{x_o\in\mathcal{O}_i\cap D_{ij}} \|\varphi(x_{oi}, x_{ji})\|. \tag{29}$$

We also define

$$j^* = \mathop{\arg\min}_{j\in\mathcal{N}_i^\sigma} \|o_{ij}^{\rm los}\|. \tag{30}$$

By using $o_{ij}^{\rm los}$ and j^* mentioned previously, $\Psi_i^{\rm los}(x)$ can be described as

$$\Psi_i^{\rm los}(x) = \Phi^{\rm los}\left(\left\|\varphi(x_i - o_{ij^*}^{\rm los}, x_i - x_{j^*})\right\|\right)$$
$$\Phi^{\rm los}(z) = \begin{cases} \frac{1}{2}\left(\left(\frac{z-d_l}{d_{\rm lr}-d_l}+\delta\right)^{-1} - \frac{1}{1+\delta}\right)^2, & \text{if } z < d_{\rm lr} \\ 0, & \text{otherwise.} \end{cases}$$

In the same way as $\Phi^{\text{obs}}(z)$, the value of $\Phi^{\text{los}}(z)$ monotonically increases as z goes from d_{lr} to the minimum allowable distance d_l from $\mathcal{L}_{ij}$ to obstacles. Thus, the component $-\nabla\Psi_i^{\text{los}}(x)$ of v_i in (25) has the effect of moving $\mathcal{L}_{ij}$ away from its closest obstacle point.

The fourth component $\Psi_i^{\text{coh}}(x)$ is introduced to make the group more cohesive. More precisely, $\Psi_i^{\text{coh}}(x)$ is decided such that $-\nabla_{x_i}\Psi_i^{\text{coh}}(x)$ is in the direction of moving closer to neighbors whose distance from robot i is more than d_n when there is no obstacle around, that is

$$\Psi_i^{\text{coh}}(x) = \sum_{j\in\mathcal{S}_i} \Phi^{\text{coh}}(\|x_i - x_j\|)$$

$$\Phi^{\text{coh}}(z) = \begin{cases} \frac{1}{2}\,(z-d_n)^2, & \text{if } \|z\| > d_n, \mathcal{O}_i = \emptyset \\ 0, & \text{otherwise.} \end{cases} \tag{31}$$

C. Magnitude of Control Input Vector

In this section, we determine $\|u_i(k)\|$ $(i = 1,\ldots,N)$ so as to achieve the preservation of $\mathcal{E}_\sigma(x(k))$ and collision avoidance. It should be noted that the leader and followers use the same algorithm to determine $\|u_i(k)\|$, unlike the direction of $u_i(k)$ in Section IV-B.

As mentioned in Section III, the proposed method aims to achieve connectivity preservation and collision avoidance not only at $x_i(k)$ for each discrete time k $(k = 0, 1, \ldots)$ but also at each point on the line segment connecting $x_i(k)$ and $x_i(k+1)$, that is $p_i \in \mathcal{P}_i(k) := \mathcal{L}(x_i(k), x_i(k+1))$. To this end, we first present an upper bound for $\|u_i(k)\|$ to satisfy the following conditions.

1) The maximum distance condition for each $j \in \mathcal{N}_i^\sigma(k)$
$$\|p_i - p_j\| \le d_n \quad \forall p_i \in \mathcal{P}_i(k) \quad \forall p_j \in \mathcal{P}_j(k). \tag{32}$$
2) The interrobot collision avoidance condition for each $j \in \mathcal{S}_i(k)$
$$\|p_i - p_j\| \ge d_c \quad \forall p_i \in \mathcal{P}_i(k) \quad \forall p_j \in \mathcal{P}_j(k). \tag{33}$$
3) The obstacle avoidance condition for each $x_o \in \mathcal{O}_i(k)$
$$\|p_i - x_o\| \ge d_o \quad \forall p_i \in \mathcal{P}_i(k). \tag{34}$$
4) The LOS preservation condition for each $j \in \mathcal{N}_i^\sigma(k)$
$$\begin{aligned} \|q - x_o\| \ge d_l \quad \forall x_o \in \mathcal{O} \quad \forall q \in \mathcal{L}(p_i, p_j) \\ \forall p_i \in \mathcal{P}_i(k) \quad \forall p_j \in \mathcal{P}_j(k). \end{aligned} \tag{35}$$

Note that the upper bound of $\|u_i(k)\|$ to satisfy 1) is derived in [20] without taking into account obstacles and collisions, while our goal here is to guarantee 1)–4) at the same time in obstacle environments. It also should be noted that conditions (33) and (34) must be satisfied for any $j \in \mathcal{V} \setminus \{i\}$ and $x_o \in \mathcal{O}$ in order to avoid collision with any robot or obstacle point. Thus, we further derive conditions where (33) and (34) are satisfied for $j \notin \mathcal{S}_i$ and $x_o \notin \mathcal{O}_i$.

The condition in [20] for $u_i(k)$ to satisfy 1) is described as follows:

$$\|u_i(k)\| \le \frac{1}{2}\Big(d_n - \max_{j\in\mathcal{N}_{ib}(k)} \|x_{ji}(k)\|\Big) =: \bar{u}_i^{\text{con1}}(k) \tag{36}$$

$$\|u_i(k)\|^2 \le \min_{j\in\mathcal{N}_{if}(k)} \big\{u_i^T(k)x_{ji}(k)\big\}. \tag{37}$$

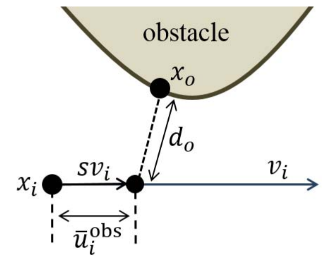

Fig. 4. Input bound $\bar{u}_i^{\text{obs}}$.

In these conditions, $\mathcal{N}_{ib}(k)$ denotes the set of robots $j \in \mathcal{N}_i^\sigma$ from which robot i will move away at time $k+1$, that is

$$\mathcal{N}_{ib}(k) = \big\{j \in \mathcal{N}_i^\sigma(k) | v_i^T(k)x_{ji}(k) \le 0\big\} \tag{38}$$

and $\mathcal{N}_{if}(k)$ denotes the rest of the robots $j \in \mathcal{N}_i^\sigma(k)$, that is

$$\mathcal{N}_{if}(k) = \big\{j \in \mathcal{N}_i^\sigma | v_i^T(k)x_{ji}(k) > 0\big\}. \tag{39}$$

In other words, $\mathcal{N}_{ib}$ is the set of robots behind robot i with respect to the direction of the movement of robot i, while $\mathcal{N}_{if}$ is the set of robots in front of robot i. Since u_i is described as

$$u_i(k) = \|u_i(k)\|\frac{v_i(k)}{\|v_i(k)\|} \tag{40}$$

using the vector $v_i(k)$ given in Section IV-B, the condition in (37) can be rewritten as

$$\|u_i(k)\| \le \frac{1}{\|v_i(k)\|} \min_{j\in\mathcal{N}_{if}(k)} \big\{v_i^T(k)x_{ji}(k)\big\} =: \bar{u}_i^{\text{con2}}(k).$$

In order to describe a condition for 2), we first decompose the set of robots $\mathcal{S}_i(k)$ detected by robot i into the following two sets:

$$\mathcal{S}_{ib}(k) = \big\{j \in \mathcal{S}_i(k) |\ v_i^T(k)x_{ji}(k) \le 0\big\} \tag{41}$$
$$\mathcal{S}_{if}(k) = \big\{j \in \mathcal{S}_i(k) |\ v_i^T(k)x_{ji}(k) > 0\big\}. \tag{42}$$

Using $\mathcal{S}_{if}(k)$, an upper bound for $\|u_i(k)\|$ is given as

$$\bar{u}_i^{\text{col}}(k) = \frac{1}{2}\Big(\min_{j\in\mathcal{S}_{if}(k)} \|x_{ji}(k)\| - d_c\Big). \tag{43}$$

An upper bound of $\|u_i(k)\|$ to satisfy 3) is given as

$$\bar{u}_i^{\text{obs}} = \max\nolimits_{s\ge 0}\{\|sv_i\| \mid \|x_{oi} - sv_i\| \ge d_o,\ \forall x_o \in \mathcal{O}_i\} \tag{44}$$

where $x_{oi} := x_o - x_i$. This upper bound allows robot i to proceed in the direction of v_i, unless the distance to the closest obstacle point is less than d_o, as indicated in Fig. 4. It should be noted that exact maximization on the right-hand side of (44) is difficult for general obstacle environments. An approximate value of $\bar{u}_i^{\text{obs}}$ can be obtained by discretizing s and x_o in (44). Another way is to approximate obstacles by a simple shape such as circles, for which the maximum in (44) can be analytically obtained (see Appendix D).

In order to describe the input bound $\bar{u}_i^{\text{los}}$ for 4), we define the set of detected obstacle points in D_{ij}, toward which robot i will move closer at $k+1$, as follows:

$$\mathcal{O}_{ijf}^{\text{los}}(k) := \big\{x_o \in \mathcal{O}_i \cap D_{ij} \ \big|\ v_i^T\varphi(x_{oi}, x_{ji}) > 0\big\}. \tag{45}$$

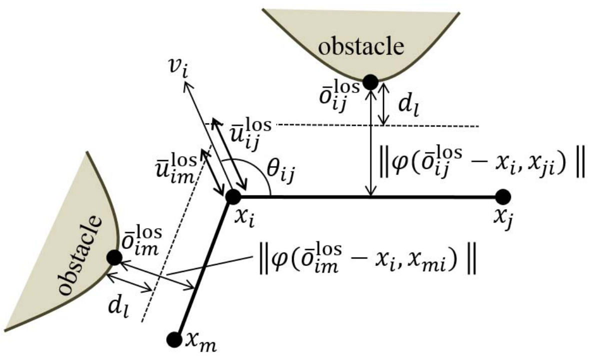

Fig. 5. Input bound $\bar{u}_{ij}^{\mathrm{los}}$.

If $\mathcal{O}_{ijf}^{\mathrm{los}}(k) = \emptyset$, no upper bound for $\|u_i(k)\|$ is given, i.e., $\bar{u}_i^{\mathrm{los}} = \infty$. If $\mathcal{O}_{ijf}^{\mathrm{los}}(k) \neq \emptyset$, we define the closest obstacle point in D_{ij} to $\mathcal{L}_{ij}(k)$ as

$$\bar{o}_{ij}^{\mathrm{los}} = \arg\min_{x_o \in \mathcal{O}_{ijf}^{\mathrm{los}}} \|\varphi(x_{oi}, x_{ji})\| \tag{46}$$

as shown in Fig. 5. Then, an upper bound for $\|u_i(k)\|$ is given as

$$\begin{aligned} \bar{u}_i^{\mathrm{los}} &= \min_{j \in \mathcal{N}_i^{\sigma}} \bar{u}_{ij}^{\mathrm{los}} \\ \bar{u}_{ij}^{\mathrm{los}} &= \max_{s \geq 0} \left\{ \|s v_i\| \;\middle|\; \left\|\varphi(\bar{o}_{ij}^{\mathrm{los}} - x_i - s v_i, x_{ji})\right\| \geq d_l \right\}. \end{aligned} \tag{47}$$

As shown in Fig. 5, $\bar{u}_{ij}^{\mathrm{los}}$ is the maximum allowable distance of movement, such that the distance from robot i to the obstacles in the direction orthogonal to $\mathcal{L}_{ij}(k)$ is not less than d_l at time $k+1$. Thus, $\bar{u}_{ij}^{\mathrm{los}}$ is obtained as

$$\bar{u}_{ij}^{\mathrm{los}} = \frac{\left\|\varphi(\bar{o}_{ij}^{\mathrm{los}} - x_i, x_{ji})\right\| - d_l}{\sin\theta_{ij}} \tag{48}$$

where $\theta_{ij} \in [0, \pi]$ is the angle between $\mathcal{L}_{ij}$ and v_i. An upper bound for $\|u_i(k)\|$ to satisfy 1)–4) is now given as

$$\bar{u}_i = \min\left\{\bar{u}_i^{\mathrm{con1}}, \bar{u}_i^{\mathrm{con2}}, \bar{u}_i^{\mathrm{col}}, \bar{u}_i^{\mathrm{obs}}, \bar{u}_i^{\mathrm{los}}, u_{\max}\right\}. \tag{49}$$

Therefore, we determine $u_i(k)$ as follows:

$$u_i(k) = \begin{cases} \bar{u}_i(k)\dfrac{v_i(k)}{\|v_i(k)\|}, & \text{if } \|v_i(k)\| > \bar{u}_i(k) \\ v_i(k), & \text{otherwise.} \end{cases} \tag{50}$$

Note that since u_i is computed using positions of detected robots and obstacles and since the sensing range of each robot is limited, the computation time for each robot does not grow as the total number of robots and obstacles increases.

Theorem 2: Suppose that collision avoidance constraints in (2) and (3) are satisfied for all robots at time k. Then, $u_i(k)$ in (50) satisfies 1)–4) for each robot $i \in \mathcal{V}$, if

$$u_{\max} \leq d_o - d_l, \quad \sqrt{d_o^2 + d_n^2} \leq d_s \tag{51}$$

in addition to (20).

Proof: See Appendix B. ■

It should be noted that the control algorithm satisfying 1)–4) does not necessarily guarantee that (33) and (34) are satisfied for $j \notin \mathcal{S}_i$ and $x_o \notin \mathcal{O}_i$. Thus, a collision with a robot hiding behind an obstacle at time k could possibly arise at time $k+1$. To guarantee collision avoidance with robots $j \notin \mathcal{S}_i(k)$ and obstacle points $x_o \notin \mathcal{O}_i$, additional conditions are required as shown in the following theorem.

Theorem 3: In addition to the assumptions in Theorem 2, we assume

$$u_{\max} \leq \min\left\{\frac{d_s}{2}, \sqrt{d_o^2 - d_l^2}\right\} - \frac{d_c}{2}. \tag{52}$$

Then, the interrobot collision avoidance condition in (33) is guaranteed for each $j \in \mathcal{V} \setminus \mathcal{S}_i(k)$. Furthermore, if

$$u_{\max} \leq d_s - d_o \tag{53}$$

the obstacle avoidance condition in (34) is guaranteed for each $x_o \in \mathcal{O} \setminus \mathcal{O}_i(k)$.

Proof: See Appendix C. ■

Remark 3: For the leader, $u_{\max}$ in (49) might be replaced by a smaller value $u_{\max}^l$, since it improves the cohesion of the group, i.e., more edges are generated in $\mathcal{E}_n$. In other words, if the leader moves at the maximum speed, i.e., $\|u_i\| = u_{\max}$, it is difficult for pairs of robots $(i, j) \notin \mathcal{E}_n$ to decrease the interrobot distance and to generate an edge in $\mathcal{E}_n$.

While conditions for connectivity preservation and collision avoidance are derived in Theorems 1–3, it is difficult to clarify conditions to ensure that no robots get stuck in narrow spaces. In Section V, we demonstrate the effectiveness and limitations of the proposed method by simulations under various conditions.

V. Simulation

In this section, we run various simulations to demonstrate the effectiveness of the proposed method. The values of the parameters defined in Section III are $N = 10$, $d_s = 2$ m, $d_n = 1$ m, $d_c = 0.3$ m, $d_o = 0.1$ m, $d_l = 0.05$ m, and $u_{\max} = 0.01$ m. The coefficients in (26) are set as $c_1 = 0.5$, $c_2 = c_3 = 0.01$, and $c_4 = 5$. The parameters in the artificial potential functions in Section IV-B are set as $\kappa_1 = \kappa_2 = 10$, $d_r = 0.7$, $d_{\mathrm{or}} = 0.2$, $d_{\mathrm{lr}} = 0.1$, and $\delta = 0.02$. We set $d_{\mathrm{del}} = 0.25$, $\epsilon_n = 0.1\ d_n$, and $\epsilon_c = 0.1\ d_c$ for link deletion, and $u_{\max}^l = (2/3)u_{\max}$ to improve the cohesion of the group, as mentioned in Remark 3.

In order to measure the connectivity of the graph $\mathcal{G}_\sigma$, we compute the Fiedler value [39], which is the second smallest eigenvalue of the Laplacian matrix L. The Laplacian matrix L is defined as

$$L = \Delta - \mathcal{A} \tag{54}$$

by using the adjacency matrix $\mathcal{A}$, whose elements $\mathcal{A}_{ij}$ are 1 if $(i, j) \in \mathcal{E}_\sigma$ and 0 otherwise, and a diagonal matrix Δ, whose diagonal elements are $\Delta_i = \sum_{j=1}^{N} \mathcal{A}_{ij}$ $(i = 1, \ldots, N)$.

Obstacle environments for simulations are shown in Fig. 6. The target path of the leader is a series of connected line segments starting at the origin, as shown in the thick solid lines. Obstacles are placed on both the sides of the path. The relative angle of the ith line segment from $i-1$th line segment is defined as $\phi_i \neq 0$ for $i = 1, \ldots, M$, if $M \geq 1$. Thus, M represents the number of corners of the target path.

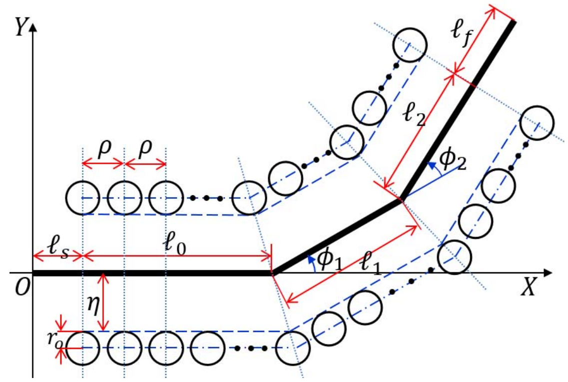

Fig. 6. Obstacle environment in simulations for $M = 2$.

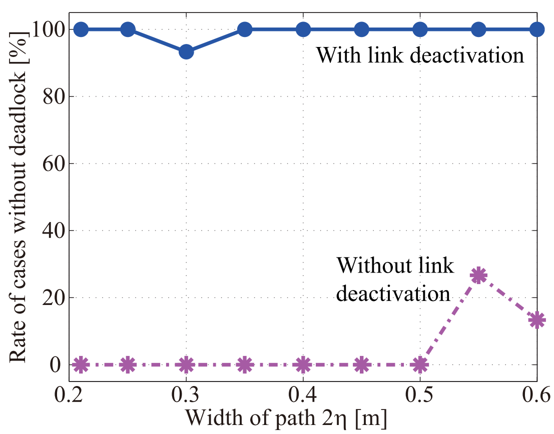

Fig. 7. Rate of cases without deadlock for $M = 0$.

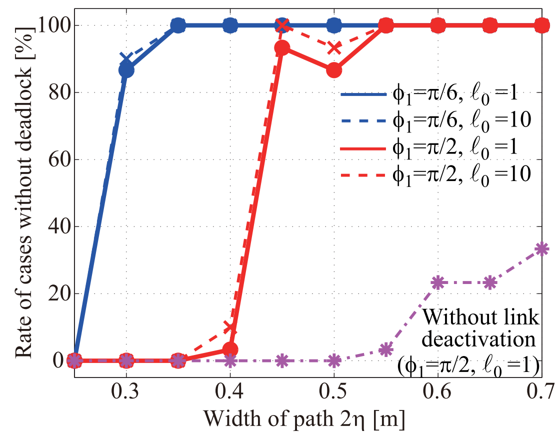

Fig. 8. Rate of cases without deadlock for $M = 1$ and $\phi_1 = (\pi/6), (\pi/2)$.

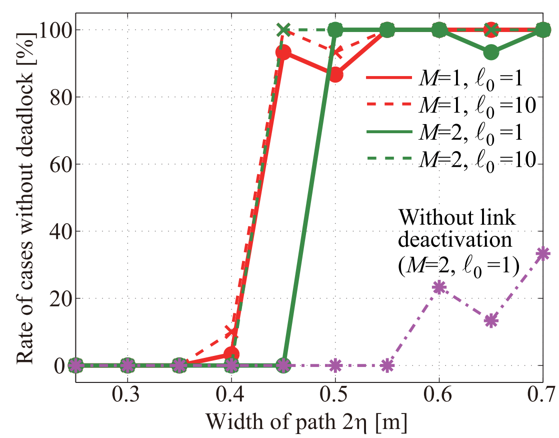

Fig. 9. Rate of cases without deadlock for $M = 1, 2$ and $\phi_1 = (\pi/2)$.

The first and last line segments can be divided into parts inside and outside the obstacle area. We define ℓ_s and ℓ_f as the lengths of the parts outside the obstacle area on the first and last line segments, respectively. Furthermore, the length of the ith line segment inside the obstacle area is defined as ℓ_i for $i = 0, 1, \ldots, M$.

The minimum distance between an obstacle and the leader's target path is defined as η, which implies that obstacles should not be in the area between two dashed lines. Subject to this minimum distance constraint, we place circular obstacles so that obstacles are as dense as possible to make the problem challenging. To this end, each obstacle is placed so that the distance from an obstacle to the leader's target path is equal to the allowable value η. In other words, the centers of circular obstacles with radius r_o are placed on a dashed dotted line in Fig. 6, which is at a distance of $\eta + r_o$ from the target path. We also set distance between the centers of neighboring obstacles, ρ, as a small value so that the density of obstacles is high. Furthermore, we located additional obstacles with a radius smaller than r_o at the inside corners, making the problem even more difficult. Specifically, we set $r_o = 0.5$ m and $\rho = 0.2$ m, while the radii of small obstacles at the inside corners were 0.1 and 0.01 m.

We first tested the proposed method in the case where the target path of the leader was a straight line, i.e., $M = 0$. This was to examine the effectiveness of the link deactivation rule in Section IV-A. If the target path has corners, it is possible for robots to get stuck for a reason not necessarily related to the proposed link deactivation rule, as will be shown later. Thus, simulations with a straight path are suitable for investigating the performance of the proposed link deactivation rule. We performed simulations for $\ell_0 = 1$ m, $\ell_s = 1$ m, and $\ell_f = 500$ m and various values of η, which is equivalent to various widths of passages, 2η. Since the allowable minimum distance between a robot and an obstacle is $d_o = 0.1$ m, we could not choose an η less than 0.1 m. We therefore tested the proposed method in cases where η ranged from 0.105 to 0.3 m. For each value of η, we performed 30 simulations for randomly selected initial robot positions. The solid line in Fig. 7 shows the percentage of cases in which all robots passed through the obstacle area without getting stuck using the proposed link deactivation rule. As may be seen in Fig. 7, the proposed link-deactivation rule was effective in most cases, in contrast to the results without link deactivation shown by the dashed dotted line. The minimum Fiedler value of $\mathcal{G}_\sigma$ in all the cases was 9.8×10^{-2}, which implied that connectivity was preserved in all the cases, since the Fiedler value was positive. Furthermore, the conditions for interrobot collision avoidance and obstacle avoidance were satisfied in all simulations, where minimum interrobot distance and minimum

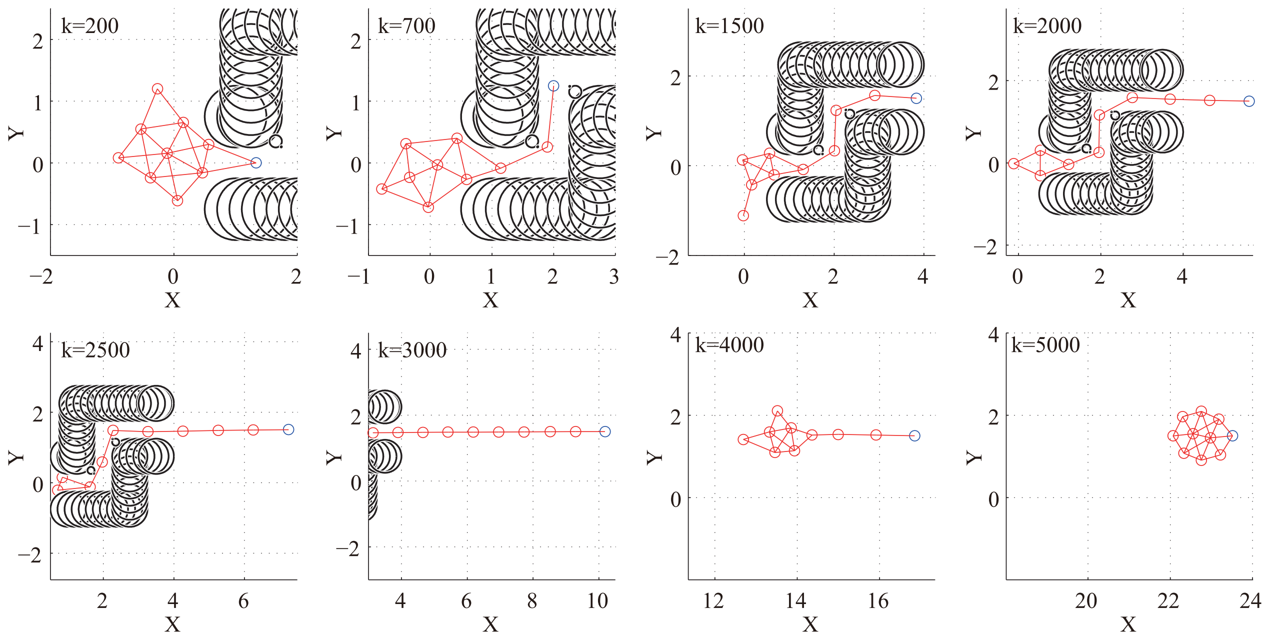

Fig. 10. Snapshots of a simulation.

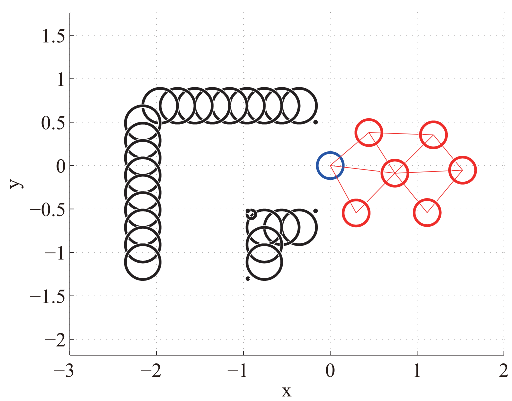

Fig. 11. Initial positions of robots (experiment).

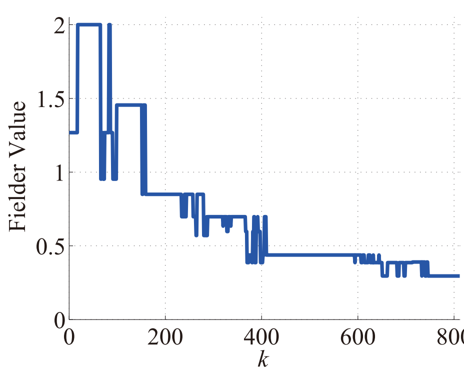

Fig. 12. Fiedler value of the graph (experiment).

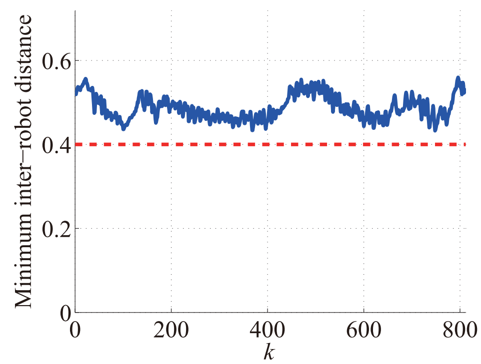

Fig. 13. Minimum interrobot distance (experiment).

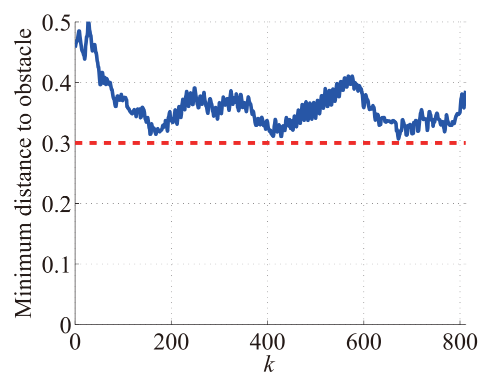

Fig. 14. Minimum distance to obstacles (experiment).

robot–obstacle distance were almost same as their allowable minimum value, d_c and d_o, respectively.

We next performed simulations in the case where the target path of the leader had a corner, i.e., $M = 1$. In this case, even if sensing links were properly deactivated to go through a path, robots could get stuck at the corner. To illustrate this problem, we show simulation results in the case of $\ell_0 = 10$ m, where the first straight path in the obstacle area is long. The dashed lines in Fig. 8 show that the percentage of cases where all robots passed through the obstacle area for $\ell_1 = 1.5$, $\ell_s = 1$ m,

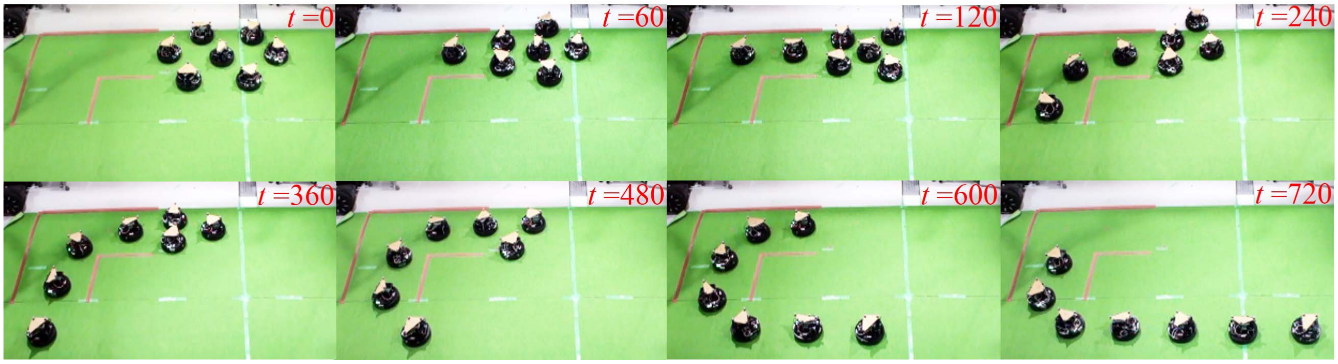

Fig. 15. Snapshots of an experiment.

and $\ell_f = 500$ m. It can be seen from this result that robots got stuck in many cases of $2\eta = 0.25$ and $2\eta \leq 0.4$ for $\phi_1 = \pi/6$ rad and $\phi_1 = \pi/2$ rad, respectively. This is in contrast that all robots passed through the obstacle area in most cases of $2.1 \leq 2\eta \leq 0.4$ for a straight path, as shown in Fig. 7. Note that since we set $N = 10$ and $d_n = 1$ m, the distance between the first and last robots is no more than 9 m. Thus, when the first robot reaches the corner in the case of $\ell_0 = 10$, all the robots are in the narrow path after the link deactivation has been successfully completed at the entrance of the narrow space. Therefore, this difference of results for $2\eta \leq 0.4$ in Fig. 7 compared with Fig. 8 illustrates that robots got stuck at a corner although the link deactivation was successfully completed. A reason why this problem arises is that the distance between two neighboring robots is too long to make a turn while preserving the LOS. Furthermore, the solid lines in Fig. 8 show that the possibility of getting stuck is increased for $\ell_0 = 1$ m, where the link deactivation has not been completed when some robots reach the corner. One reason for this is that the velocity of robots around an entrance of the narrow space is decreased during link deactivation. This makes it difficult for the robots in front to move forward. As a result, the distance between robots at the corner becomes longer than that in the case of $\ell_0 = 10$. Similarly, the rate of getting stuck is increased as the number of corners increases. Fig. 9 compares the cases of $M = 1$ and $M = 2$ where we set $\phi_1 = \pi/2$, $\phi_2 = -\pi/2$, and $\ell_1 = \ell_2 = 1.5$ m. As shown in Fig. 9, the rate of getting stuck is higher for $M = 2$. The limitation in making turns at narrow corners might be improved by modifying the direction of movement in Section IV-B so that the distance between robots is decreased at corners, although that is beyond the focus of this paper. Despite this limitation, the simulation results show that the robots can successfully go through paths in most cases of $2\eta \geq 0.5$ where it is still difficult to navigate without link deactivation. It should also be noted that the conditions for connectivity preservation, interrobot collision avoidance, and obstacle avoidance were satisfied in all examples shown in this section, regardless of whether the paths have corners or not. Fig. 10 shows snap shots of the case where $M = 2$, which illustrate that the robots are able to pass through narrow spaces by decreasing the number of active links, and then can increase the active links in free space. However, there is a limitation that regrouping into a cohesive formation takes quite a long time, since it is difficult to start regrouping before the last robot clears the obstacle area. One reason for this is that the leader (the first robot) does not consider group cohesion in the control law. Thus, in order to improve the cohesion of the group, followers need to catch up with the leader. However, the followers other than the last robot cannot easily get closer to the leader, since they need to keep the maximum allowable distance d_n with the neighbor behind as well as in front. Therefore, regrouping is not triggered until the last robot gets closer to the neighbor in front, after it clears the obstacle area.

VI. Experiment

The proposed method was applied to a group of seven robots. For each robot, we used a mobile robot platform (Kobuki, Yujin Robot). Since the robots were not omnidirectional, their orientation was controlled in the direction of the control input u_i before moving forward. The position and orientation of each robot were measured by a motion capture system (OptiTrack s250e) in a centralized way. Obstacle avoidance was simulated using virtual obstacles whose positions were known to the robots. However, only local information that could be obtained in the sensing model in Section III was used to compute the control input in order to simulate the decentralized algorithm. Experimental validation using onboard sensors is planned for future research.

The control algorithm was implemented every 0.5 s. The values of the parameters in Section III were $d_s = 1.6$ m, $d_n = 0.8$ m, $d_c = 0.4$ m, $d_o = 0.3$ m, $d_l = 0.2$ m, and $u_{\max} = 0.04$ m. The coefficients in (26) were set to $c_1 = 0.6$, $c_2 = 0.1$, and $c_3 = c_4 = 1$. Parameters in the artificial potential functions were set to $\kappa_1 = \kappa_2 = 100$, $d_r = 0.7$, $d_{\mathrm{or}} = d_{\mathrm{lr}} = 0.4$, and $\delta = 0.02$. We set $d_{\mathrm{del}} = 0.3$ for link deletion in (21) and $u^l_{\max} = \frac{2}{3}u_{\max}$ to improve the cohesion of the group.

As shown in Fig. 11, the virtual obstacles formed an L-shaped path. The initial positions of the leader and followers are indicated by blue and red circles, respectively. As shown in Fig. 12, the Fiedler value of $\mathcal{G}_\sigma$ was always positive, which implies that connectivity was preserved. Fig. 13 shows that interrobot distances (solid line) did not violate the minimum allowable value d_c (dashed line). Furthermore, as shown in Fig. 14, the distances to obstacles did not violate the minimum allowable value d_o. Snap shots of the experiment are shown in Fig. 15, where the red lines in each photograph represent the L-shaped path formed by the virtual obstacles. Fig. 15 shows that the robots traversed the L-shaped path by deactivating links to be preserved.

VII. Conclusion

This paper has presented a network connectivity preservation method for leader–follower navigation in obstacle environments that explicitly takes an input constraint into account. We derived conditions for the proposed method to guarantee connectivity preservation and collision avoidance in the presence of obstacles. A deactivation rule of sensing links, which uses only local sensing information to preserve global network connectivity, was introduced to allow the robots to navigate narrow spaces without getting stuck. The effectiveness of the proposed method was demonstrated by simulations and in experiments. Future research will address the problem that robots may get stuck in a corner even if sensing links are properly deactivated depending on the width of the path. The algorithm should also be improved so as to regroup robots into a cohesive formation more rapidly in free spaces.

Appendix A
Proof of Theorem 1

To prove Theorem 1 by contradiction, suppose that $\mathcal{G}_\sigma$ is not connected, under Assumptions 1)–4) in Theorem 1.

From Assumption 1), $\mathcal{G}_n$ is connected. Thus, there exists at least one pair of robots, i and j of $\mathcal{G}_\sigma$, such that all the paths connecting them are lost by applying the rule in (24), while i and j have an edge in $\mathcal{G}_n$, i.e., $(i, j) \in \mathcal{E}_n$. This implies that robots i and j and another one m satisfy (21) or (23), so that (i, j) is deactivated by the rule in (24). We denote this robot triple by $(A_{h_1}, B_{h_1}, C_{h_1})$, where $h_1 = 1$ in the case of $(A_{h_1}, B_{h_1}, C_{h_1}) \in \overline{\mathcal{T}}$ as in (23) while $h_1 = 2$ in the case of $(A_{h_1}, B_{h_1}, C_{h_1}) \in \mathcal{T}$ as in (21). Then, the edge (A_{h_1}, B_{h_1}) is deactivated by the rule in (24), regardless of $h_1 = 1$ or $h_1 = 2$. Therefore, if neither the edge (A_{h_1}, C_{h_1}) nor (B_{h_1}, C_{h_1}) is deactivated, there still exists a path $A_{h_1}C_{h_1}B_{h_1}$ between A_{h_1} and B_{h_1}, i.e., between robots i and j. From Lemma 1, there are three cases where (A_{h_1}, C_{h_1}) or (B_{h_1}, C_{h_1}) is deactivated, under Assumptions 2) and 3). We define $(A_{(h_1,h_2)}, B_{(h_1,h_2)}, C_{(h_1,h_2)})$ as the robot triple that causes such a deactivation, where $h_2 = 1$, 2, or 3 corresponding to 1)–3) in Lemma 1 for $(A, B, C) = (A_{h_1}, B_{h_1}, C_{h_1})$ and $(A', B', C') = (A_{(h_1,h_2)}, B_{(h_1,h_2)}, C_{(h_1,h_2)})$. We also define $\mathcal{I}(k) := (h_1, h_2, \ldots, h_k)$ in order to make the notation simple, which implies that

$$(A_{(h_1,h_2)}, B_{(h_1,h_2)}, C_{(h_1,h_2)}) = (A_{\mathcal{I}(2)}, B_{\mathcal{I}(2)}, C_{\mathcal{I}(2)}). \quad (55)$$

Although the edge $(A_{\mathcal{I}(2)}, B_{\mathcal{I}(2)})$, which is equivalent to (A_{h_1}, C_{h_1}) or (B_{h_1}, C_{h_1}), is deactivated in each case of $h_2 = 1, 2, 3$, there still exists a path $A_{\mathcal{I}(2)}C_{\mathcal{I}(2)}B_{\mathcal{I}(2)}$ between $A_{\mathcal{I}(2)}$ and $B_{\mathcal{I}(2)}$, if the edges $(A_{\mathcal{I}(2)}, C_{\mathcal{I}(2)})$ and $(B_{\mathcal{I}(2)}, C_{\mathcal{I}(2)})$ remain. However, these edges can be deactivated, if there exists a robot triple $(A_{\mathcal{I}(3)}, B_{\mathcal{I}(3)}, C_{\mathcal{I}(3)})$ that satisfies 1)–3) in Lemma 1 for $(A, B, C) = (A_{\mathcal{I}(2)}, B_{\mathcal{I}(2)}, C_{\mathcal{I}(2)})$ and $(A', B', C') = (A_{\mathcal{I}(3)}, B_{\mathcal{I}(3)}, C_{\mathcal{I}(3)})$, where $h_3 = 1, 2, 3$ corresponding to 1)–3) in Lemma 1.

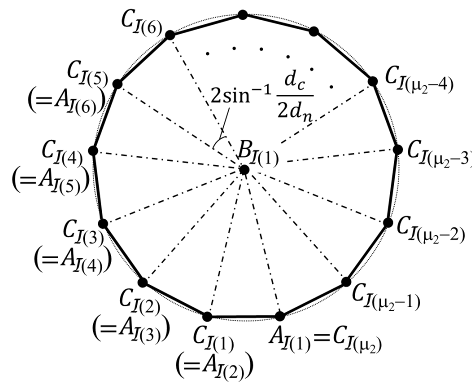

Fig. 16. Example of the case where all paths between two robots $A_{\mathcal{I}(1)}$ and $B_{\mathcal{I}(1)}$ are lost by deactivation rule in (24).

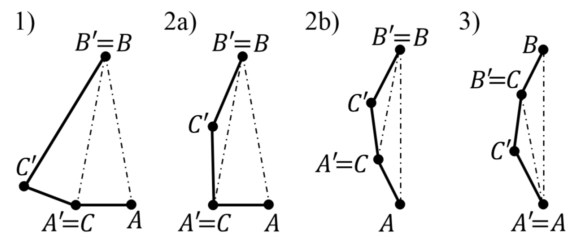

Fig. 17. Example of 1)–3) in Lemma 1. $(A, B, C) \in \overline{\mathcal{T}}$ in 1) and 2a), while $(A, B, C) \in \mathcal{T}$ in 2b) and 3).

Similarly, $(A_{\mathcal{I}(\mu)}, B_{\mathcal{I}(\mu)}, C_{\mathcal{I}(\mu)})$ $(h_\mu = 1, 2, 3)$ is defined for $\mu \geq 4$ as the robot triple that deactivates the edge $(A_{\mathcal{I}(\mu-1)}, B_{\mathcal{I}(\mu-1)})$ or $(B_{\mathcal{I}(\mu-1)}, C_{\mathcal{I}(\mu-1)})$. Since the number of robots is finite, the length of such a sequence of deactivations is finite. Thus, the path between A_{h_1} and C_{h_1} is not lost, unless a sequence of robot triples for deactivation form a loop; i.e., unless there exist two positive integers μ_1 and μ_2 $(\mu_1 < \mu_2)$ such that $(A_{\mathcal{I}(\mu_2)}, C_{\mathcal{I}(\mu_2)})$ or $(B_{\mathcal{I}(\mu_2)}, C_{\mathcal{I}(\mu_2)})$ is equivalent to $(A_{\mathcal{I}(\mu_1)}, B_{\mathcal{I}(\mu_1)})$. As shown in Lemma 2, such a loop is formed only if $(A_{\mathcal{I}(\mu)}, B_{\mathcal{I}(\mu)})$ $(\mu = 1, 2, \ldots, \mu_2)$ are each deactivated due to $(A_{\mathcal{I}(\mu)}, B_{\mathcal{I}(\mu)}, C_{\mathcal{I}(\mu)}) \in \overline{\mathcal{T}}$, i.e., $h_\mu = 1$ $(\mu = 1, 2, \ldots, \mu_2)$, until a loop is formed, as shown in Fig. 16. To form such a loop without collision, it is therefore necessary for 2π to be an integer multiple of $2\sin^{-1}(d_c/2d_n)$, which contradicts Assumption 4).

A. Lemmas to Prove Theorem 1

In this section, we denote the distance between robots A and B by $\overline{AB}$, in order to simplify the notation.

Lemma 1: In addition to Assumptions 2) and 3) in Theorem 1, we assume that three out of N robots satisfy $(A, B, C) \in \mathcal{T}$ or $\overline{\mathcal{T}}$ so that the edge (A, B) is deactivated by the rule in (24). Then, the edge (B, C) or (A, C) is also deactivated, if and only if at least one of the following three conditions is satisfied, as shown in Fig. 17.

1) There exists a robot triple $(A', B', C') \in \overline{\mathcal{T}}$ that deactivates the edge (B, C), which is equivalent to (A', B').
2) There exists a robot triple $(A', B', C') \in \mathcal{T}$ that deactivates the edge (B, C), which is equivalent to (A', B').

3) There exists a robot triple $(A', B', C') \in \mathcal{T}$ that deactivates the edge (A, C), which is equivalent to (A', B').

Proof: We only prove the necessity, since the sufficiency is obvious from the deactivation rule in (24).

From Lemma 3, we have $(C, A, B) \notin \mathcal{T}$ and $(B, C, A) \notin \mathcal{T}$, if $(A, B, C) \in \mathcal{T}$. Also, it is obvious from the definition of $\overline{\mathcal{T}}$ that $(C, A, B) \notin \overline{\mathcal{T}}$ and $(B, C, A) \notin \overline{\mathcal{T}}$ if $(A, B, C) \in \overline{\mathcal{T}}$. Thus, the three robots (A, B, C) do not deactivate the edge (A, C) or (B, C) by themselves. In other words, (A, C) or (B, C) is deactivated, only if a robot other than (A, B, C) constitutes a triple $(A', B', C') \in \mathcal{T}$ or $\overline{\mathcal{T}}$, such that (A', B') is equivalent to (A, C) or (B, C).

We first consider the case of $(A, B, C) \in \mathcal{T}$. Then, from Lemma 4, there is no robot triple $(A', B', C') \in \overline{\mathcal{T}}$ that deactivates the edge (B, C) or (A, C), which is equivalent to (A', B'). In other words, (B, C) or (A, C) is deactivated only if $(A', B', C') \in \mathcal{T}$, as shown in Fig. 17(2b) and (3).

In the case of $(A, B, C) \in \overline{\mathcal{T}}$, the edge (A, C) cannot be deactivated as shown in Lemma 5. In other words, only (B, C) can be deactivated due to $(A', B', C') \in \overline{\mathcal{T}}$ as in Fig.17(1) or $(A', B', C') \in \mathcal{T}$ as in Fig. 17(2a). ∎

Lemma 2: Under Assumptions 2) and 3) in Theorem 1, there exist two positive integers μ_1 and μ_2 $(\mu_1 < \mu_2)$ such that $(A_{\mathcal{I}(\mu_2)}, C_{\mathcal{I}(\mu_2)})$ or $(B_{\mathcal{I}(\mu_2)}, C_{\mathcal{I}(\mu_2)})$ is equivalent to $(A_{\mathcal{I}(\mu_1)}, B_{\mathcal{I}(\mu_1)})$, only if $h_\mu = 1$ $(\mu = 1, 2, \ldots, \mu_2)$.

Proof: To prove the contrapositive of the lemma, suppose that $h_\mu \neq 1$ for an integer μ $(1 \leq \mu \leq \mu_2)$. We define $\bar{\mu}$ $(\bar{\mu} \leq \mu_2)$ as the smallest positive integer μ satisfying $h_\mu \neq 1$. Then, it follows from Lemma 4 that $h_\mu \neq 1$, i.e., $(A_{\mathcal{I}(\mu)}, B_{\mathcal{I}(\mu)}, C_{\mathcal{I}(\mu)}) \in \mathcal{T}$, for $\bar{\mu} < \mu \leq \mu_2$. Thus, from Lemma 6, we have

$$\begin{aligned}\overline{A_{\mathcal{I}(\mu_2)}C_{\mathcal{I}(\mu_2)}} &< \overline{A_{\mathcal{I}(\mu_2)}B_{\mathcal{I}(\mu_2)}}\\ \overline{B_{\mathcal{I}(\mu_2)}C_{\mathcal{I}(\mu_2)}} &< \overline{A_{\mathcal{I}(\mu_2)}B_{\mathcal{I}(\mu_2)}}.\end{aligned} \tag{56}$$

Since the edge $(A_{\mathcal{I}(\mu_2-1)}, C_{\mathcal{I}(\mu_2-1)})$ or $(B_{\mathcal{I}(\mu_2-1)}, C_{\mathcal{I}(\mu_2-1)})$ is equivalent to $(A_{\mathcal{I}(\mu_2)}, B_{\mathcal{I}(\mu_2)})$, it follows from (56) and Lemma 6 that:

$$\begin{aligned}\overline{A_{\mathcal{I}(\mu_2)}C_{\mathcal{I}(\mu_2)}} &< \overline{A_{\mathcal{I}(\mu_2-1)}B_{\mathcal{I}(\mu_2-1)}}\\ \overline{B_{\mathcal{I}(\mu_2)}C_{\mathcal{I}(\mu_2)}} &< \overline{A_{\mathcal{I}(\mu_2-1)}B_{\mathcal{I}(\mu_2-1)}}.\end{aligned} \tag{57}$$

By repeating the same process, we have

$$\begin{aligned}\overline{A_{\mathcal{I}(\mu_2)}C_{\mathcal{I}(\mu_2)}} &< \overline{A_{\mathcal{I}(\mu)}B_{\mathcal{I}(\mu)}}\\ \overline{B_{\mathcal{I}(\mu_2)}C_{\mathcal{I}(\mu_2)}} &< \overline{A_{\mathcal{I}(\mu)}B_{\mathcal{I}(\mu)}}\end{aligned} \tag{58}$$

for $\bar{\mu} \leq \mu < \mu_2$. Thus, in the case of $\bar{\mu} \leq \mu_1$, we have

$$\begin{aligned}\overline{A_{\mathcal{I}(\mu_2)}C_{\mathcal{I}(\mu_2)}} &< \overline{A_{\mathcal{I}(\mu_1)}B_{\mathcal{I}(\mu_1)}}\\ \overline{B_{\mathcal{I}(\mu_2)}C_{\mathcal{I}(\mu_2)}} &< \overline{A_{\mathcal{I}(\mu_1)}B_{\mathcal{I}(\mu_1)}} \quad \forall \mu_1 < \mu_2.\end{aligned} \tag{59}$$

We next consider the case of $\mu_1 < \bar{\mu}$. Since $\bar{\mu} > 1$ in this case, we have $h_\mu = 1$, i.e., $(A_{\mathcal{I}(\mu)}, B_{\mathcal{I}(\mu)}, C_{\mathcal{I}(\mu)}) \in \overline{\mathcal{T}}$, for $1 \leq \mu < \bar{\mu}$. This implies from the definition of $\overline{\mathcal{T}}$ that

$$\overline{A_{\mathcal{I}(\mu)}B_{\mathcal{I}(\mu)}} = \overline{A_{\mathcal{I}(\bar{\mu})}B_{\mathcal{I}(\bar{\mu})}} = d_n \tag{60}$$

for $1 \leq \mu < \bar{\mu}$, since the edge $(B_{\mathcal{I}(\mu)}, C_{\mathcal{I}(\mu)})$ is equivalent to $(A_{\mathcal{I}(\mu+1)}, B_{\mathcal{I}(\mu+1)})$. Therefore, we have

$$\overline{A_{\mathcal{I}(\mu_1)}B_{\mathcal{I}(\mu_1)}} = \overline{A_{\mathcal{I}(\bar{\mu})}B_{\mathcal{I}(\bar{\mu})}} \tag{61}$$

which implies (59) from (58). Thus, neither $(A_{\mathcal{I}(\mu_2)}, C_{\mathcal{I}(\mu_2)})$ nor $(B_{\mathcal{I}(\mu_2)}, C_{\mathcal{I}(\mu_2)})$ is equivalent to $(A_{\mathcal{I}(\mu_1)}, B_{\mathcal{I}(\mu_1)})$ for any μ_1 and μ_2 $(\mu_1 < \mu_2)$, which completes the proof of the contrapositive of the lemma. ∎

Lemma 3: In addition to Assumptions 2) and 3) in Theorem 1, suppose that three out of N robots satisfy $(A, B, C) \in \mathcal{T}$. Then, we have $(C, A, B) \notin \mathcal{T}$ and $(B, C, A) \notin \mathcal{T}$.

Proof: Under the collision avoidance condition in Assumption 2), we have $\overline{AB} \geq d_c$, $\overline{BC} \geq d_c$, $\overline{CA} \geq d_c$. Let $\alpha_A \in (-\pi, \pi]$ denote the angle from x_{BA} to x_{CA} measured in the counterclockwise direction, and we define α_B and α_C in the same way. Then, it follows from a property of triangles that:

$$|\alpha_A| + |\alpha_B| + |\alpha_C| = \pi. \tag{62}$$

In order to prove $(C, A, B) \notin \mathcal{T}$ by contradiction, we assume $(C, A, B) \in \mathcal{T}$. Since (20) is assumed in Assumption 3), it holds from (18) and (20) that

$$\|\varphi(x_{BC}, x_{AC})\| = \overline{BC} \sin|\alpha_C| < d_c \sin\frac{\pi}{3} \tag{63}$$

which implies from $\overline{BC} \geq d_c$ that $|\alpha_C| < \frac{\pi}{3}$. Similarly, from $(A, B, C) \in \mathcal{T}$, it can be shown that $|\alpha_A| < \frac{\pi}{3}$ and $|\alpha_B| < \frac{\pi}{3}$, which contradicts (62). This concludes the proof of $(C, A, B) \notin \mathcal{T}$. It can be proved similarly that $(B, C, A) \notin \mathcal{T}$. ∎

Lemma 4: In addition to Assumptions 2) and 3) in Theorem 1, we assume $(A, B, C) \in \mathcal{T}$. Then, there is no robot triple $(A', B', C') \in \overline{\mathcal{T}}$ that deactivates the edge (A, C) or (B, C), which is equivalent to (A', B').

Proof: For $(A, B, C) \in \mathcal{T}$, we have

$$\overline{AC} < \overline{AB}, \quad \overline{BC} < \overline{AB} \tag{64}$$

as shown in Lemma 6. Since it follows from the definition of $\mathcal{T}$ that $(A, B) \in \mathcal{E}_n$, we have $\overline{AB} \leq d_n$. Therefore, it holds from (64) that

$$\overline{AC} < d_n, \quad \overline{BC} < d_n. \tag{65}$$

Thus, there is no robot triple $(A', B', C') \in \overline{\mathcal{T}}$ whose edge (A', B') corresponds to (A, C) or (B, C), since $\overline{A'B'} = d_n$ is required for $(A', B', C') \in \overline{\mathcal{T}}$ from the definition of $\overline{\mathcal{T}}$. ∎

Lemma 5: In addition to Assumptions 2) and 3) in Theorem 1, we assume $(A, B, C) \in \overline{\mathcal{T}}$. Then, it is not possible that the edge (A, C) is deactivated by the rule in (24).

Proof: The assumption $(A, B, C) \in \overline{\mathcal{T}}$ implies $\overline{AC} = d_c$. To prove by contradiction, we first assume that (A, C) is deactivated due to $(A', B', C') \in \mathcal{T}$ whose edge (A', B') is equivalent to (A, C). Then, from Lemma 6, we have

$$\overline{A'C'} < \overline{A'B'} = d_c, \quad \overline{B'C'} < \overline{A'B'} = d_c, \tag{66}$$

which implies that A' and B' collide with C'. This contradicts Assumption 2) that all robots satisfy the collision avoidance constraints.

We next assume that (A, C) is deactivated due to $(A', B', C') \in \overline{\mathcal{T}}$ whose edge (A', B') is equivalent to (A, C). Then, since $\overline{A'B'} = d_n$ from the definition of $\overline{\mathcal{T}}$, we have $\overline{AC} = d_n > d_c$, which contradicts $(A, B, C) \in \overline{\mathcal{T}}$. ∎

Lemma 6: In addition to Assumptions 2) and 3) in Theorem 1, we assume that $(A, B, C) \in \mathcal{T}$. Then, we have

$$\overline{AC} < \overline{AB}, \quad \overline{BC} < \overline{AB}. \tag{67}$$

Proof: To prove by contradiction, suppose that $\overline{BC} \geq \overline{AB}$ and $\overline{BC} \geq \overline{AC}$, without loss of generality.

We first show that $\cos\alpha_A \leq \frac{1}{2}$ in each case of $\overline{AC} \leq \overline{AB}$ and $\overline{AC} > \overline{AB}$. In the case of $\overline{AC} \leq \overline{AB}$, it holds from $\overline{BC} \geq \overline{AB}$ that

$$\cos\alpha_A = \frac{\overline{AB}^2 + \overline{AC}^2 - \overline{BC}^2}{2\overline{AB}\cdot\overline{AC}} \leq \frac{\overline{AC}}{2\overline{AB}} \leq \frac{1}{2}. \tag{68}$$

In the case of $\overline{AC} > \overline{AB}$, it holds from $\overline{BC} \geq \overline{AC}$ that

$$\cos\alpha_A = \frac{\overline{AB}^2 + \overline{AC}^2 - \overline{BC}^2}{2\overline{AB}\cdot\overline{AC}} \leq \frac{\overline{AB}}{2\overline{AC}} < \frac{1}{2}. \tag{69}$$

Thus, we have $\cos\alpha_A \leq \frac{1}{2}$, which implies $|\alpha_A| \geq \frac{\pi}{3}$. Furthermore, since $(A, B, C) \in \mathcal{T}$, it follows from (18) that $x_C \in D_{AB}$, which implies $|\alpha_A| < \frac{\pi}{2}$. Therefore, since $\overline{CA} \geq d_c$ due to the collision avoidance Assumption 2)

$$\varphi(x_{CA}, x_{BA}) = \overline{CA}\sin|\alpha_A| \geq d_c \sin\frac{\pi}{3} \tag{70}$$

which contradicts $(A, B, C) \in \mathcal{T}$, under Assumption 3) that (20) is satisfied. ∎

Appendix B
Proof of Theorem 2

If conditions 2)–4) are guaranteed, 1) can be proved in the same way as in [20]. Thus, in this section, we prove 2)–4).

In order to show that 2) is satisfied, it suffices to provide proof for the worst case scenario in which two robots move closer to each other. Thus, we show that 2) is satisfied for any pair of robots (i, j) that satisfies $j \in \mathcal{S}_{if}$, $i \in \mathcal{S}_{jf}$. Since $\|u_i\| \leq \bar{u}_i^{\mathrm{col}}$ for $\bar{u}_i^{\mathrm{col}}$ in (43), the following constraints are satisfied:

$$\min_{m\in\mathcal{S}_{if}} \|x_{mi}(k)\| - 2\|u_i(k)\| \geq d_c \tag{71}$$

$$\min_{m\in\mathcal{S}_{jf}} \|x_{mj}(k)\| - 2\|u_j(k)\| \geq d_c. \tag{72}$$

Therefore, from

$$\min_{m\in\mathcal{S}_{if}} \|x_{mi}(k)\| \leq \|x_{ji}(k)\| \tag{73}$$

$$\min_{m\in\mathcal{S}_{jf}} \|x_{mj}(k)\| \leq \|x_{ij}(k)\| \tag{74}$$

and $\|x_{ji}(k)\| = \|x_{ij}(k)\|$, we have

$$\|x_{ji}(k)\| - 2\|u_i(k)\| \geq d_c \tag{75}$$

$$\|x_{ji}(k)\| - 2\|u_j(k)\| \geq d_c. \tag{76}$$

By summing(75) and (76), we obtain

$$\|x_{ji}(k)\| - \|u_i(k)\| - \|u_j(k)\| \geq d_c \tag{77}$$

for each (i, j) that satisfies $j \in \mathcal{S}_{if}$ and $i \in \mathcal{S}_{jf}$. Thus, it holds for all $\lambda_i, \lambda_j \in [0, 1]$ that

$$\|x_{ji}(k)\| - \|\lambda_i u_i(k)\| - \|\lambda_j u_j(k)\| \geq d_c \tag{78}$$

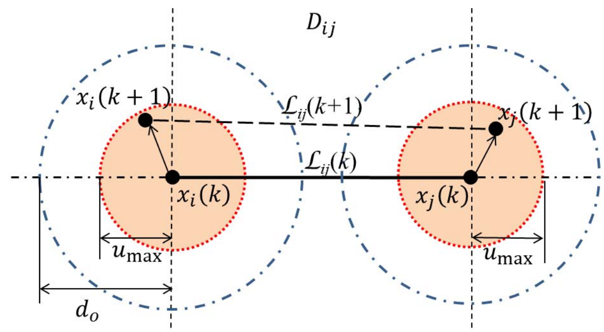

Fig. 18. Relations between $\mathcal{L}_{ij}$, d_o, and $u_{\max}$.

which implies

$$\|\lambda_i u_i(k) - x_{ji}(k) - \lambda_j u_j(k)\| \geq d_c. \tag{79}$$

Therefore, it holds from (1) that

$$\|\lambda_i x_i(k+1) + (1-\lambda_i)x_i(k) - \lambda_j x_j(k+1) - (1-\lambda_j)x_j(k)\| \geq d_c \tag{80}$$

which implies that 2) is satisfied.

We next prove 3). It holds from the definition of $\bar{u}_i^{\mathrm{obs}}$ in (44) that

$$\|x_{oi} - s v_i\| \geq d_o \quad \forall x_o \in \mathcal{O}_i \tag{81}$$

for each s such that $\|s v_i\| \leq \bar{u}_i^{\mathrm{obs}}$. Since v_i and u_i have the same direction, it follows from $\|u_i\| \leq \bar{u}_i^{\mathrm{obs}}$ that:

$$\|x_{oi} - \lambda_i u_i\| \geq d_o \quad \forall x_o \in \mathcal{O}_i \quad \forall \lambda_i \in [0, 1]. \tag{82}$$

Thus, it holds from (1) that

$$\|x_o - (1-\lambda_i)x_i(k) - \lambda_i x_i(k+1)\| \geq d_o \quad \forall x_o \in \mathcal{O}_i \tag{83}$$

for all $\lambda_i \in [0, 1]$, which implies that 3) is satisfied.

To prove 4), we first define $\mathcal{B}(p, r)$ to be a closed ball centered at p of radius r. Since $\|u_i(k)\| = \|x_i(k+1) - x_i(k)\| \leq u_{\max}$ from (1), the line segments $\mathcal{P}_i(k)$ and $\mathcal{P}_j(k)$ are included in $\mathcal{B}(x_i(k), u_{\max})$ and $\mathcal{B}(x_j(k), u_{\max})$, respectively, as shown by the colored circles in Fig. 18. This implies that $\mathcal{L}_{ij}(k+1)$ is included in $\mathrm{co}(\mathcal{B}(x_i(k), u_{\max}) \cup \mathcal{B}(x_j(k), u_{\max}))$ where $\mathrm{co}(X)$ denotes the convex hull of a set X. Thus, since $d_o - u_{\max} \geq d_l$ from the assumption in (51), the obstacle points outside $\mathrm{co}(\mathcal{B}(x_i(k), d_o) \cup \mathcal{B}(x_j(k), d_o))$ have a distance of more than d_l from any point in $\mathrm{co}(\mathcal{B}(x_i(k), u_{\max}) \cup \mathcal{B}(x_j(k), u_{\max}))$ including $\mathcal{L}_{ij}(k+1)$. Therefore, it is sufficient to consider the obstacle points in $\mathrm{co}(\mathcal{B}(x_i(k), d_o) \cup \mathcal{B}(x_j(k), d_o))$.

Furthermore, from the assumption that (3) is satisfied at time k, there is no obstacle point in $\mathcal{B}(x_i(k), d_o)$ and $\mathcal{B}(x_j(k), d_o)$. Thus, it is sufficient to consider the obstacle points in

$$\mathcal{A}_{ij}(k) := D_{ij} \cap \mathrm{co}(\mathcal{B}(x_i(k), d_o) \cup \mathcal{B}(x_j(k), d_o)). \tag{84}$$

Therefore, in order to prove 4), it suffices to show

$$\|\varphi(x_o - q, x_{ji}(k))\| \geq d_l \quad \forall x_o \in \mathcal{A}_{ij}(k) \quad \forall q \in \mathcal{L}(p_i, p_j) \quad \forall p_i \in \mathcal{P}_i(k) \quad \forall p_j \in \mathcal{P}_j(k) \tag{85}$$

for each $j \in \mathcal{N}_i^{\sigma}(k)$, since $\|x_o - q\| \geq \|\varphi(x_o - q, x_{ji}(k))\|$.

From the definition of φ in (17), we have

$$\|\varphi(x_o - p_i, x_{ji}(k))\| = \frac{|(x_o - p_i)^T H x_{ji}(k)|}{\|H x_{ji}(k)\|} = \frac{|(x_{oi}(k) - p_i + x_i(k))^T H x_{ji}(k)|}{\|H x_{ji}(k)\|}. \tag{86}$$

For $j \in \mathcal{N}_i^\sigma(k)$, we have $\|\varphi(x_{oi}(k), x_{ji}(k))\| \geq d_l > 0$ for any $x_o \in \mathcal{O}_i$, since the constraint in (8) is satisfied at time k. Thus, any obstacle point $x_o \in \mathcal{O}_i \cap D_{ij}$ belongs to one of the following two sets:

$$\mathcal{B}_{ij}(k) := \{x_o \in \mathcal{O}_i \cap D_{ij} \mid x_{oi}^T(k) H x_{ji}(k) > 0\}$$
$$\overline{\mathcal{B}}_{ij}(k) := \{x_o \in \mathcal{O}_i \cap D_{ij} \mid x_{oi}^T(k) H x_{ji}(k) < 0\}.$$

Without loss of generality, we assume $\upsilon_i^T H x_{ji} > 0$, which implies $\mathcal{O}_{ijf}^{\text{los}}(k) = \mathcal{B}_{ij}(k)$ in (45) from

$$\upsilon_i^T \varphi(x_{oi}, x_{ji}) = \frac{x_{oi}^T H x_{ji}}{\|H x_{ji}\|^2} \upsilon_i^T H x_{ji} > 0. \tag{87}$$

Thus, $\bar{o}_{ij}^{\text{los}}$ in (46) can be described as

$$\bar{o}_{ij}^{\text{los}} = \underset{x_o \in \mathcal{B}_{ij}}{\arg\min}\ x_{oi}^T H x_{ji}. \tag{88}$$

Therefore, it holds from (47) and (88) that

$$\frac{(x_{oi} - s\upsilon_i)^T H x_{ji}}{\|H x_{ji}\|} \geq d_l, \quad \forall x_o \in \mathcal{B}_{ij} \tag{89}$$

$$\frac{(x_{oi} - s\upsilon_i)^T H x_{ji}}{\|H x_{ji}\|} \leq -d_l, \quad \forall x_o \in \overline{\mathcal{B}}_{ij} \tag{90}$$

for each $s \geq 0$ such that $\|s\upsilon_i\| \leq \bar{u}_i^{\text{los}}$. This implies that

$$\frac{(x_{oi} - p_i + x_i)^T H x_{ji}}{\|H x_{ji}\|} \geq d_l, \quad \forall x_o \in \mathcal{B}_{ij}$$
$$\frac{(x_{oi} - p_i + x_i)^T H x_{ji}}{\|H x_{ji}\|} \leq -d_l, \quad \forall x_o \in \overline{\mathcal{B}}_{ij} \tag{91}$$

for each $p_i \in \mathcal{P}_i$, since $p_i = x_i + \lambda_i u_i$ and $\|\lambda_i u_i\| \leq \bar{u}_i^{\text{los}}$. Under the assumption that $(d_o^2 + d_n^2)^{1/2} \leq d_s$ in (51), any obstacle point in $\mathcal{A}_{ij}(k)$ is in the maximum sensor range of robot i at time k, which implies $\mathcal{A}_{ij} \subset \mathcal{O}_i \cap D_{ij}$, Thus, it follows from (91) that:

$$\frac{(x_{oi} - p_i + x_i)^T H x_{ji}}{\|H x_{ji}\|} \geq d_l, \quad \forall x_o \in \mathcal{B}_{ij}^{\mathcal{A}}$$
$$\frac{(x_{oi} - p_i + x_i)^T H x_{ji}}{\|H x_{ji}\|} \leq -d_l, \quad \forall x_o \in \overline{\mathcal{B}}_{ij}^{\mathcal{A}} \tag{92}$$

where

$$\mathcal{B}_{ij}^{\mathcal{A}}(k) := \{x_o \in \mathcal{A}_{ij} \mid x_{oi}^T(k) H x_{ji}(k) > 0\}$$
$$\overline{\mathcal{B}}_{ij}^{\mathcal{A}}(k) := \{x_o \in \mathcal{A}_{ij} \mid x_{oi}^T(k) H x_{ji}(k) < 0\}.$$

For p_j, as in (92), it can be proved that

$$\frac{(x_{oi} - p_j + x_i)^T H x_{ji}}{\|H x_{ji}\|} \geq d_l, \quad \forall x_o \in \mathcal{B}_{ij}^{\mathcal{A}}$$
$$\frac{(x_{oi} - p_j + x_i)^T H x_{ji}}{\|H x_{ji}\|} \leq -d_l, \quad \forall x_o \in \overline{\mathcal{B}}_{ij}^{\mathcal{A}} \tag{93}$$

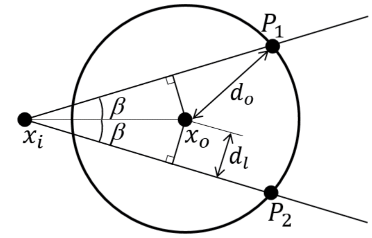

Fig. 19. Closest points to x_i behind an obstacle point x_o.

for each $p_j \in \mathcal{P}_j$, using $x_j^T H x_{ij} = -x_i^T H x_{ji}$. Thus, it holds from (92) and (93) that

$$\frac{(x_{oi} - q + x_i)^T H x_{ji}}{\|H x_{ji}\|} \geq d_l, \quad \forall x_o \in \mathcal{B}_{ij}^{\mathcal{A}}$$
$$\frac{(x_{oi} - q + x_i)^T H x_{ji}}{\|H x_{ji}\|} \leq -d_l, \quad \forall x_o \in \overline{\mathcal{B}}_{ij}^{\mathcal{A}} \tag{94}$$

for each $q \in \mathcal{L}(p_i, p_j)$. Therefore, we have (85), which completes the proof of 4).

Appendix C
Proof of Theorem 3

We first prove (33) for each robot $j \in \mathcal{V} \setminus \mathcal{S}_i$. From the definition of $\mathcal{S}_i$, robot i is not able to detect robot j (i.e., $j \in \mathcal{V} \setminus \mathcal{S}_i$), when at least one of the conditions in (7) and (8) is violated. If (7) is not satisfied at time k, we have $\|x_{ji}(k)\| > d_s$. Since $p_i \in \mathcal{P}_i(k)$ and $p_j \in \mathcal{P}_j(k)$ are described as

$$p_i = x_i(k) + \lambda_i u_i(k), \quad p_j = x_j(k) + \lambda_j u_j(k) \tag{95}$$

for $\lambda_i \in [0, 1]$ and $\lambda_j \in [0, 1]$, it holds that

$$\|p_j - p_i\| = \|x_{ji}(k) + \lambda_j u_j(k) - \lambda_i u_i(k)\| \geq \|x_{ji}(k)\| - 2u_{\max} > d_s - 2u_{\max} \tag{96}$$

from $\|\lambda_i u_i(k)\| \leq u_{\max}$ and $\|\lambda_j u_j(k)\| \leq u_{\max}$. Thus, since $d_s - 2u_{\max} \geq d_c$ under the assumption in (52), the condition in (33) is satisfied.

In the case where (8) is violated, a robot is behind an obstacle. Since it is assumed that the obstacle avoidance condition in (3) is satisfied at k, each robot is located on the boundary or the outside of $\mathcal{B}(x_o, d_o)$ for each obstacle point x_o. Thus, as shown in Fig. 19, robot i is not able to detect robot j behind an obstacle point x_o, if the angle between x_{ji} and x_{oi} is less than $\beta = \text{atan2}(d_l, (x_{oi}^2 - d_l^2)^{1/2})$. In this case, the distance between robots i and j satisfies

$$\|x_{ji}(k)\| > \sqrt{\|x_{oi}(k)\|^2 - d_l^2} + \sqrt{d_o^2 - d_l^2} \tag{97}$$

where the lower bound on the right-hand side is equal to the distance from x_i to point P_1 or P_2 in Fig. 19. Furthermore, it follows from (97) and $\|x_{oi}(k)\| \geq d_o$, that we obtain:

$$\|x_{ji}(k)\| > 2\sqrt{d_o^2 - d_l^2}. \tag{98}$$

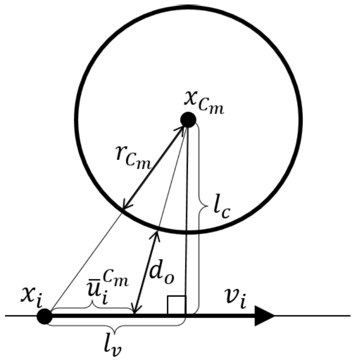

Fig. 20. Input bound $\bar{u}_i^{C_m}$ for a circle obstacle.

Since $2u_{\max} \leq 2(d_o^2 - d_l^2)^{1/2} - d_c$ under the assumption in (52), the inputs of robots i and j are constrained as follows:

$$2\sqrt{d_o^2 - d_l^2} - 2\|u_i(k)\| \geq d_c \tag{99}$$

$$2\sqrt{d_o^2 - d_l^2} - 2\|u_j(k)\| \geq d_c. \tag{100}$$

Thus, from (98), we have

$$\|x_{ji}(k)\| - 2\|u_i(k)\| \geq d_c \tag{101}$$

$$\|x_{ji}(k)\| - 2\|u_j(k)\| \geq d_c \tag{102}$$

which are the same inequalities as (75) and (76). The rest of the proof is then the same as the proof of 2) in Theorem 2.

We next prove (34) for each obstacle point $x_o \in \mathcal{O} \setminus \mathcal{O}_i(k)$. From the definition of $\mathcal{O}_i$, robot i is not able to detect an obstacle point x_o (i.e., $x_o \in \mathcal{O} \setminus \mathcal{O}_i$) when at least one of the conditions in (9) and (10) is violated. If (9) is not satisfied at time k, we have $\|x_{oi}(k)\| > d_s$. Then, in the same way as in (96), we have

$$\|x_o - p_i\| = \|x_{oi} - \lambda_i u_i\| > d_s - u_{\max} \tag{103}$$

for p_i in (95), since $\|\lambda_i u_i\| \leq u_{\max}$ for $\lambda_i \in [0, 1]$. Thus, since $d_s - u_{\max} \geq d_o$ under the assumption in (53), the condition in (34) is satisfied for each $x_o \in \mathcal{O} \setminus \mathcal{O}_i(k)$.

If an obstacle point x_o does not satisfy (10), there is a detected point $x_o' \in \mathcal{O}_i \cap \bar{\mathcal{L}}(x_i, x_o)$ which is located on the line segment connecting x_i and x_o. Since the obstacle point x_o behind x_o' obviously does not move unlike in the case where a robot is behind an obstacle point, (34) is satisfied for x_o if it is satisfied for x_o'. Thus, since Theorem 2 guarantees that (34) is satisfied for any detected obstacle point including x_o', (34) is satisfied for any obstacle point x_o violating (10).

APPENDIX D
COMPUTATION OF $\bar{u}_i^{\text{obs}}$ FOR CIRCULAR OBSTACLES

In this section, we describe how to compute $\bar{u}_i^{\text{obs}}$ in (44) when obstacles detected by robot i are approximated by circles. In other words, $\mathcal{O}_i$ in (44) is replaced by $\cup_{m=1}^{N_c} C_m$, where N_C is the number of circles, and C_m is the circumference of the circle centered at x_{C_m} of radius r_{C_m}. In this case, $\bar{u}_i^{\text{obs}}$ is described as

$$\bar{u}_i^{\text{obs}} = \min_{1 \leq m \leq N_c} \bar{u}_i^{C_m} \tag{104}$$

$$\bar{u}_i^{C_m} = \max_{s \geq 0}\{\|sv_i\| \mid \|x_{oi} - sv_i\| \geq d_o, \forall x_o \in C_m\}. \tag{105}$$

In order to describe how to obtain $\bar{u}_i^{C_m}$, we define

$$l_c := \|\varphi(x_{C_m} - x_i, v_i)\| \tag{106}$$

which is the distance from x_{C_m} to the line including x_i and $x_i + v_i$, as shown in Fig. 20. Then, a necessary and sufficient condition for $\bar{u}_i^{C_m}$ in (105) to be finite is that the following inequalities are satisfied:

$$(x_{C_m} - x_i)^T v_i > 0, \quad l_c - r_{C_m} < d_o. \tag{107}$$

The first inequality in (107) implies that the robot moves closer to C_m, while the second one implies that the distance from C_m to the line including x_i and $x_i + v_i$ is less than d_o. Using the condition in (107), $\bar{u}_i^{C_m}$ is obtained as

$$\bar{u}_i^{C_m} = \begin{cases} l_v - \sqrt{(d_o + r_{C_m})^2 - l_c^2}, & \text{if (107)} \\ \infty, & \text{otherwise} \end{cases} \tag{108}$$

where $l_v := (\|x_{C_m} - x_i\|^2 - l_c^2)^{1/2}$ as shown in Fig. 20.

REFERENCES

[1] Y. U. Cao, A. S. Fukunaga, and A. Kahng, "Cooperative mobile robotics: Antecedents and directions," *Auto. Robots*, vol. 4, no. 1, pp. 7–27, 1997.
[2] T. Arai, E. Pagello, and L. E. Parker, "Editorial: Advances in multi-robot systems," *IEEE Trans. Robot. Autom.*, vol. 18, no. 5, pp. 655–661, Oct. 2002.
[3] R. M. Murray, "Recent research in cooperative control of multi-vehicle systems," *J. Dyn. Syst., Meas., Control*, vol. 129, no. 5, pp. 571–583, 2007.
[4] Y. Kim and M. Mesbahi, "On maximizing the second smallest eigenvalue of a state-dependent graph Laplacian," *IEEE Trans. Autom. Control*, vol. 51, no. 1, pp. 116–120, Jan. 2006.
[5] M. M. Zavlanos and G. J. Pappas, "Potential fields for maintaining connectivity of mobile networks," *IEEE Trans. Robot.*, vol. 23, no. 4, pp. 812–816, Aug. 2007.
[6] M. C. De Gennaro and A. Jadbabaie, "Decentralized control of connectivity for multi-agent systems," in *Proc. 45th IEEE Int. Conf. Decision Control*, Dec. 2006, pp. 3628–3633.
[7] P. Yang, R. A. Freeman, G. J. Gordon, K. M. Lynch, S. S. Srinivasa, and R. Sukthankar, "Decentralized estimation and control of graph connectivity for mobile sensor networks," *Automatica*, vol. 46, no. 2, pp. 390–396, 2010.
[8] M. Franceschelli, A. Gasparri, A. Giua, and C. Seatzu, "Decentralized estimation of Laplacian eigenvalues in multi-agent systems," *Automatica*, vol. 49, no. 4, pp. 1031–1036, 2013.
[9] L. Sabattini, N. Chopra, and C. Secchi, "Decentralized connectivity maintenance for cooperative control of mobile robotic systems," *Int. J. Robot. Res.*, vol. 32, no. 12, pp. 1411–1423, Sep. 2013.
[10] L. Sabattini, C. Secchi, N. Chopra, and A. Gasparri, "Distributed control of multirobot systems with global connectivity maintenance," *IEEE Trans. Robot.*, vol. 29, no. 5, pp. 1326–1332, Oct. 2013.
[11] M. M. Zavlanos and G. J. Pappas, "Distributed connectivity control of mobile networks," *IEEE Trans. Robot.*, vol. 24, no. 6, pp. 1416–1428, Dec. 2008.
[12] M. M. Zavlanos, H. G. Tanner, A. Jadbabaie, and G. J. Pappas, "Hybrid control for connectivity preserving flocking," *IEEE Trans. Autom. Control*, vol. 54, no. 12, pp. 2869–2875, Dec. 2009.
[13] M. Ji and M. Egerstedt, "Distributed coordination control of multiagent systems while preserving connectedness," *IEEE Trans. Robot.*, vol. 23, no. 4, pp. 693–703, Aug. 2007.

[14] A. Ajorlow, A. Momeni, and A. G. Aghdam, "A class of bounded distributed control strategies for connectivity preservation in multi-agent systems," *IEEE Trans. Autom. Control*, vol. 55, no. 12, pp. 2828–2833, Dec. 2010.

[15] A. Ajorlou and A. G. Aghdam, "Connectivity preservation in nonholonomic multi-agent systems: A bounded distributed control strategy," *IEEE Trans. Autom. Control*, vol. 58, no. 9, pp. 2366–2371, Sep. 2013.

[16] H. Su, X. Wang, and G. Chen, "A connectivity-preserving flocking algorithm for multi-agent systems based only on position measurements," *Int. J. Control*, vol. 82, no. 7, pp. 1334–1343, Jul. 2009.

[17] G. Wen, Z. Duan, H. Su, G. Chen, and W. Yu, "A connectivity-preserving flocking algorithm for nonlinear multi-agent systems with bounded potential function," in *Proc. 30th Chin. Control Conf.*, Jul. 2011, pp. 6018–6024.

[18] H. Ando, Y. Oasa, I. Suzuki, and M. Yamashita, "Distributed memoryless point convergence algorithm for mobile robots with limited visibility," *IEEE Trans. Robot. Autom.*, vol. 15, no. 5, pp. 818–828, Oct. 1999.

[19] J. Cortés, S. Martínez, and F. Bullo, "Robust rendezvous for mobile autonomous agents via proximity graphs in arbitrary dimensions," *IEEE Trans. Autom. Control*, vol. 51, no. 8, pp. 1289–1298, Aug. 2006.

[20] A. Cezayirli and F. Kerestecioğlu, "Navigation of non-communicating autonomous mobile robots with guaranteed connectivity," *Robotica*, vol. 31, no. 5, pp. 767–776, 2013.

[21] A. Ganguli, J. Cortés, and F. Bullo, "Multirobot rendezvous with visibility sensors in nonconvex environments," *IEEE Trans. Robot.*, vol. 25, no. 2, pp. 340–352, Apr. 2009.

[22] P. R. Giordano, A. Franchi, C. Secchi, and H. H. Bülthoff, "A passivity-based decentralized strategy for generalized connectivity maintenance," *Int. J. Robot. Res.*, vol. 32, no. 3, pp. 299–323, 2013.

[23] X. Li, D. Sum, and J. Yang, "A bounded controller for multirobot navigation while maintaining network connectivity in the presence of obstacles," *Automatica*, vol. 49, no. 1, pp. 285–292, 2013.

[24] D. Panagou and V. Kumar, "Cooperative visibility maintenance for leader–follower formations in obstacle environments," *IEEE Trans. Robot.*, vol. 30, no. 4, pp. 831–844, Aug. 2014.

[25] M. M. Zavlanos, M. B. Egerstedt, and G. J. Pappas, "Graph-theoretic connectivity control of mobile robot networks," *Proc. IEEE*, vol. 99, no. 9, pp. 1525–1540, Sep. 2011.

[26] J. P. Desai, J. P. Ostrowski, and V. Kumar, "Modeling and control of formations of nonholonomic mobile robots," *IEEE Trans. Robot. Autom.*, vol. 17, no. 6, pp. 905–908, Dec. 2001.

[27] A. K. Das, R. Fierro, V. Kumar, J. P. Ostrowski, J. Spletzer, and C. J. Taylor, "A vision-based formation control framework," *IEEE Trans. Robot. Autom.*, vol. 18, no. 5, pp. 813–825, Oct. 2002.

[28] H. G. Tanner, G. J. Pappas, and V. Kumar, "Leader-to-formation stability," *IEEE Trans. Robot. Autom.*, vol. 20, no. 3, pp. 443–455, Jun. 2004.

[29] P. Ögren, M. Egerstedt, and X. Hu, "A control Lyapunov function approach to multiagent coordination," *IEEE Trans. Robot. Autom.*, vol. 18, no. 5, pp. 847–851, Oct. 2002.

[30] R. Olfati-Saber, "Flocking for multi-agent dynamic systems: Algorithms and theory," *IEEE Trans. Autom. Contr.*, vol. 51, no. 3, pp. 403–420, Mar. 2006.

[31] H. Su, X. Wang, and Z. Lin, "Flocking of multi-agents with a virtual leader," *IEEE Trans. Autom. Control*, vol. 54, no. 2, pp. 293–307, Feb. 2009.

[32] D. Sakai, H. Fukushima, and F. Matsuno, "Flocking for multirobots without distinguishing robots and obstacles," *IEEE Trans. Control Syst. Technol.*, vol. 25, no. 3, pp. 1019–1027, May 2017.

[33] H. Fukushima, K. Kon, and F. Matsuno, "Model predictive formation control using branch-and-bound compatible with collision avoidance problems," *IEEE Trans. Robot.*, vol. 29, no. 5, pp. 1308–1317, Oct. 2013.

[34] M. A. Lewis and K.-H. Tan, "High precision formation control of mobile robots using virtual structures," *Auto. Robots*, vol. 4, no. 4, pp. 387–403, 1997.

[35] W. Ren and R. W. Beard, "Decentralized scheme for spacecraft formation flying via the virtual structure approach," *J. Guid., Control, Dyn.*, vol. 27, no. 1, pp. 73–82, 2004.

[36] N. Michael and V. Kumar, "Planning and control of ensembles of robots with non-holonomic constraints," *Int. J. Robot. Res.*, vol. 28, no. 8, pp. 962–975, Aug. 2009.

[37] S. P. Hou and C. C. Cheah, "Dynamic compound shape control of robot swarm," *IET Control Theory Appl.*, vol. 6, no. 3, pp. 454–460, Feb. 2012.

[38] K. Yoshida, H. Fukushima, K. Kon, and F. Matsuno, "Control of a group of mobile robots based on formation abstraction and decentralized locational optimization," *IEEE Trans. Robot.*, vol. 30, no. 3, pp. 550–565, Jun. 2014.

[39] M. Fiedler, "Algebraic connectivity of graphs," *Czechoslovak Math. J.*, vol. 23, no. 2, pp. 298–305, 1973.

Daito Sakai received the B.S. degree in engineering and M.S. degrees in engineering, with a focus on formation control of multi-robot systems, from Kyoto University, Kyoto, Japan, in 2012 and 2014, respectively.

He is currently with Hitachi Construction Machinery Co., Ltd., Tsuchiura, Japan.

Hiroaki Fukushima (M'06) received the B.S. and M.S. degrees in engineering and the Ph.D. degree in informatics from Kyoto University, Kyoto, Japan, in 1995, 1998 and 2001, respectively.

From 2004 to 2009, he was a Research Associate and an Assistant Professor with the University of Electro-Communications, Chofu, Japan. He is currently an Assistant Professor with Kyoto University. His current research interests include the control design of mobile robots, predictive control, and system identification.

Dr. Fukushima was a Research Fellow of the Japan Society for the Promotion of Science from 1999 to 2004.

Fumitoshi Matsuno (M'94) received the Dr. Eng. degree from Osaka University, Suita, Japan, in 1986.

In 1986, he joined the Department of Control Engineering, Osaka University. Since 2009, he has been a Professor with the Department of Mechanical Engineering and Science, Kyoto University, Kyoto, Japan. He is also the Vice-President of the Institute of Systems, Control and Information Engineers and NPO International Rescue System Institute, Kobe, Japan. His current research interests include robotics, swarm intelligence, the control of distributed parameter system and nonlinear system, and rescue support system in disaster.

Dr. Matsuno received many awards, including the Outstanding Paper Award in 2001 and 2006, the Takeda Memorial Prize in 2001 from the Society of Instrument and Control Engineers (SICE), the Prize for Academic Achievement from Japan Society of Mechanical Engineers (JSME) in 2009, and the Best Paper Award in 2013 from the Information Processing Society of Japan. He served as a General Chair of the IEEE SSRR2011 and the IEEE/SICE SII2011, SWARM2015, and SWARM2017. He is a Fellow Member of the SICE, the JSME, and the Robotics Society of Japan.