This paper reports some results of network connectivity control experiments using a multi-robot system which we developed. Although a lot of connectivity control algorithms for a multi-robot network are proposed, almost all of them are verified only on computer simulations or using experimental robots with centralized sensors and controllers. To execute experimental verifications of connectivity control algorithms on a distributed robotic system, we developed an experimental multi-robot system. Hardware installed on the robot and information flow from sensors to actuators are detailed. Some results of measurement experiments are shown to estimate accuracy to detect a neighbor position. Then, results of connectivity control experiments using the developed multi-robot system are discussed.

**Keywords:** multi-robot systems, experimental systems, connectivity preservation

1. Introduction

In recent years, multi-robot systems have been studied in several fields because they exhibit complex behaviors when in a swarm, where individual robots that exhibit simple behaviors interact each other [1, 2]. The interaction in real robots employs mechanical contacts, observations by sensors, and wireless communication. In particular, it is desirable to create a connected network structure, which is a model of information flow for the entire system, to perform a collaborative task. For this reason, several autonomous distributed control methods have been studied to preserve network connectivity [3–13], by assuming that a multi-robot system exchanges information through wireless communication.

However, experimental verification is important for proposing robot control methods because real environments contain numerous unmodeled uncertainties. In the field of multirobot systems or swarm robotics, experimental robotic systems are developed to examine the efficiency of the proposed algorithms or systems [14–20]. Additionally, a remotely accessible multirobot system was recently presented and widely used [20]. In most cases, a multirobot experimental system consists of expensive centralized devices when there is a demand for observations with high accuracy and communications with high frequency.

Several experimental verifications on the connectivity preservation methods have also been conducted for multirobot systems [3–6]. In these experimental verifications, the absolute positions of the robots provided by centralized measurement devices, such as a motion capture system (e.g., VICON [a]), were used to determine the control inputs of each robot. The control inputs in some of these experiments were determined using a centralized computer. However, this varies from the assumption of multirobot systems, such that each robot acts in an autonomous distributed manner. It is preferable that the experimental robot measures the environment and determines its control input using the robot itself. In many cases, data measured by sensors installed on a mobile robot exhibited poor accuracy compared to the absolute position data from static sensors. In verification experiments, it is important to confirm the robustness against such measurement errors.

The difficulty in implementing network connectivity preservation algorithms arises from the difference between network topologies of wireless communication and sensor observation. Majority of the connectivity preservation algorithms assume that the two topologies are similar, so that each robot exchanges information with its proper neighbors; however, it is not true. The differences between the two topologies occur owing to the differences between the communication range and sensing range; sensors such as a camera are available only within the line of sight, but wireless communication might be available even out of sight owing to diffraction or reflection. To exchange information with proper neighbors via a wireless communication network, each robot must observe the relative positions and network IDs of the neighbors simultaneously.

To experimentally verify the connectivity preservation control in a distributed manner, we developed a multirobot experimentation system in which the robot measures the positions of neighbor robots and the environment using sensors mounted on itself. Each developed robot is equipped with a two-dimensional LiDAR and an omnidirectional camera to observe the relative posi-
Hiasa, T. and Murayama, T.

Fig. 1. Developed experimental robot.

Fig. 2. Local coordinate system of robot and representative dimensions (plan view, unit: mm). Rotation axis of the two-dimensional LiDAR and optical axis of the omnidirectional camera are the same with the origin of the coordinate system.

Fig. 3. Information flow for robot control.

2. Developed System Configuration

In this section, the system configurations of the developed multirobot are outlined. First, we have introduced the hardware installed in the robot, and thereafter, we have described the internal information processing. The communication network structure considered in this study has been defined, and some notations related to a connectivity control algorithm have also been defined.

2.1. Hardware

In this study, we developed a two-wheeled robot that can move on a flat surface, as depicted in Fig. 1. We employed LEGO Mindstorms NXT as actuators and their controller and Raspberry Pi 4B (4 GB) as the computer to determine wheel speeds. YDLIDAR X2 [b] was equipped as a distance sensor, and a camera with a 220° wide-angle lens [c] was also equipped. The maximum ranging distance, relative measurement error, and angular resolution of YDLIDAR X2 are 8 m, 3.5%, and 0.013 rad, respectively, from the datasheet [b]. A 220° camera was mounted upwards to capture all directions in a single image. The robot body was enclosed with the boarding so that it could be observed using the LiDAR equipped with other robots. In addition, a full-color LED tape was attached as a marker to identify the robot such that it can be observed from all directions. The system recognizes the LEDs of the other robots through image processing.

The local coordinate system and the representative dimensions of the robot are depicted in Fig. 2. The LiDAR and 220° wide-angle camera were installed at the origin of the coordinates. Although the ground speed of the wheel was more than 0.25 m/s, we limited the maximum speed at 0.05 m/s for safety reasons. This is because the experiments detailed in subsequent sections were conducted in an arena limited to 3 m square.

2.2. Software

Figure 3 depicts the information flow from the sensor measurements for determining the wheel speeds in the developed robot. The Raspberry Pi 4B obtained the angle $\theta_j$ [rad] at which robot $j$ was located on the image via image processing. The position, $p_j$, of robot $j$ in the local coordinate system was calculated by integrating the information from the LiDAR. Thereafter, the wheel speeds of the robot were determined based on the positions of the neighbor robots. The pair of wheel speed information was sent from Raspberry Pi to Mindstorms NXT. The Mindstorms NXT executes the received wheel speed commands using a PI control.

The detailed process of the Raspberry Pi is as follows. At first, the LED mounted on neighbor robot $j$ was captured via image processing, and the direction angle $\theta_j$ of neighbor $j$ in the local coordinate system was computed. The omnidirectional image was projected onto $480 \times 480$ pixel and angle $\theta_j$ was detected via thresholding. $\theta_j$ denotes the direction angle to neighbor robot $j$ on the local coordinate system, as depicted in Fig. 2. Image processing was implemented using a Python program. Based on the angle $\theta_j$ of the neighbor robot $j$, the smallest distance $d_j$ was extracted from the LiDAR measurement data in the range $[\theta_j - \delta, \theta_j + \delta]$, where $\delta$
denotes the margin; here, we set $\delta = 0.15$ rad. Thus, $d_j$ denotes the distance to the nearest point of the vehicle body of neighbor robot $j$. The position $p_j$ of neighbor $j$ in the local coordinate system is calculated as $p_j = (d_j \cos \theta_j, d_j \sin \theta_j)$. Based on the position of the neighbor robot and LiDAR measurements, the ground speeds $v_r, v_l$ [m/s] of the robot's right and left wheels were determined and sent to Mindstorms via Bluetooth. The Mindstorms controlled the two motors using PI control algorithm. All the above processes, except for image recognition, were implemented using C++ programs.

In image processing, the robot was identified using the color and blinking pattern of the LED. Because the robot ID and position of neighbor robots could be obtained using the above mentioned methods, it is possible to realize the connectivity preservation methods. It is difficult for the robots to recognize the LED color of a neighbor robot at a distance greater than 1.5 m due to ambient lights. For this reason, we defined the maximum communication range in this study as 1.5 m.

The ground speeds $v_r$ and $v_l$ of both wheels were determined from the objective velocity $(v_r, v_l)$ on the local coordinate system, as depicted in Fig. 2. The calculation is as follows:

$$
\begin{pmatrix}
  v_r \\
  v_l
\end{pmatrix} = \begin{pmatrix}
  1 & \frac{w}{2c} \\
  1 & -\frac{w}{2c}
\end{pmatrix} \begin{pmatrix}
  v_r \\
  v_l
\end{pmatrix}, \quad \ldots \quad (1)
$$

where $w = 0.115$ m denotes the distance between the two wheels, and $c = 0.03$ m denotes the distance from center of two wheels to that of the body. The objective velocity $(v_r, v_l)$ was determined from the cooperative control algorithms detailed in the subsequent sections. The ground speeds $v_r$ and $v_l$ were forcibly limited as $|v_r| \leq 0.05$ m/s, $|v_l| \leq 0.05$ m/s.

### 2.3. Line-of-Sight Network

In this study, we considered a line-of-sight network in which a communication link was established if the two corresponding robots were visible to each other. We defined two robots that were visible to each other when the distance was less than 1.5 m owing to the limitation of ID reorganization. The connectivity of the line-of-sight network has been discussed in this study.

In this paper, we have defined the notations for the network. A set of neighbors that robot $i$ obtained at position $p_j$ is defined by $N_i$. From the neighbor set $N_i$, we defined an adjacency matrix $A$ whose $ij$-th element $a_{ij}$ is $a_{ij} > 0$ if $j \in N_i$ and $a_{ij} = 0$ otherwise. The element $a_{ij}$ (or the link weight between robots $i$ and $j$) can be expressed using the sigmoid function:

$$a_{ij} = 1 - \left(1 + \exp \left(-a \left(\frac{d_j}{R} - b\right)\right)\right)^{-1}, \quad \ldots \quad (2)$$

where $R$ denotes the maximum communication distance; and $a$ and $b$ are coefficients that shape the sigmoid curve. The Laplacian matrix $L$ was calculated from the adjacency matrix $A$. It is well known that the second smallest eigenvalue $\lambda_2$ of the Laplacian matrix $L$ implies connectivity of the communication network [21]. If $\lambda_2 > 0$ then the network is connected. If $\lambda_2 = 0$, then it is disconnected. For this reason, the second smallest eigenvalue $\lambda_2$ is also called the algebraic connectivity.

Although the algebraic connectivity $\lambda_2$ was estimated using a distributed algorithm [22], the data in the estimation algorithm were exchanged via a centralized server in this study. Because the algebraic connectivity estimation should be sufficiently fast compared to the control period, here, we employed a Wi-Fi and a centralized server as the information exchange pathway. The implementation of fully distributed information exchange is a future scope of our study. All the calculations for the measurement, motion control, and connectivity estimation were executed by each computer mounted on the robot.

### 3. Relative Position Measurement

To confirm the measurement accuracy of the robot, we conducted experiments for measuring the relative position of a neighbor robot using the LiDAR, a camera, and the method described in the previous section. Fig. 4 depicts a snapshot of the experiment. The details and results of the experiments are described below. In this section, we describe the distance to the neighbor robot as $d$, the angle of the neighbor as $\theta$, and the position of the neighbor as $p$, by omitting the subscript $j$. This is because in this section, the neighbor robot is the only one that we consider in this section.

First, we confirmed the distance measurement error, which is the discrepancy between the distances measured by the LiDAR and the actual distances. Fig. 5 depicts the average and maximum errors of eight measurements of the neighbor robots placed at distances $d$ [m] $\in \{0.50, 0.75, 1.00, 1.25, 1.50\}$. The error-bars indicate the standard deviations of the data, and the gray solid line indicates the average error (0.061 m) of all data. We can see
that the distance error does not depend on the distance. Because the average error 0.061 m approximately equals the radius of the robot body, we assumed an offset condition. We concluded that the accuracy of the distance measurement was sufficient for this study as the maximum error $\Delta d$ was less than 0.035 m when the offset was considered.

Subsequently, we conducted an experiment to obtain the direction (angle) of the neighbor robot using the image processing detailed in the previous section. In this experiment, the neighbor robot was set to the angle $\theta$ [rad] $\in \{0, \pi/4, \pi/2, 3\pi/4, 5\pi/4, 3\pi/2, 7\pi/4\}$ and the distance $d$ [m] $\in \{0.25, 0.50, 0.75, 1.00, 1.25\}$. Five measurement data were obtained at each point. Fig. 6 depicts the maximum, average, and minimum errors of the measurement, respectively. The error-bars indicate standard deviations of the obtained data. We confirmed that the maximum of the angle measurement error $\Delta \theta$ is approximately 0.07 rad = 4°.

From the above results of the distance measurement and angle measurement experiments, we can calculate the estimation error of the position of the neighbor robot. The position estimation error $\Delta p$ can be approximated by the first-order Taylor series:

$$
\Delta p = \begin{pmatrix}
\Delta d \cos \theta - \Delta \theta d \sin \theta \\
\Delta d \sin \theta + \Delta \theta d \cos \theta
\end{pmatrix}.
$$

Therefore, the estimation error distance $\|\Delta p\|$ [m] can be calculated as,

$$
\|\Delta p\| = \sqrt{\Delta d^2 + (\Delta \theta d)^2}.
$$

This indicates $\|\Delta p\| \leq 0.11$ m from $\Delta d \leq 0.035$, $\Delta \theta \leq 0.07$, and $d \leq 1.50$ as indicated in the above results.

The distance measured using the LiDAR and angle measured via image processing were integrated for estimating the position of the neighbor robot. Here, we reported the results of the position-estimation experiments. The experimental condition of the neighbor robot’s position was the same as that in the direction measurement experiment, that is, the position $(x, y) = (d \cos \theta, d \sin \theta)$. The estimated positions were plotted with blue $\circ$-markers as depicted in Fig. 7, and the actual positions are plotted with red $\times$-markers. Fig. 8 depicts the average and maximum errors of the estimated positions at each distance. Error bars indicate the standard deviations of the error. Although the error increases with respect to the distance to the neighbor robot, the position could be estimated with an error of less than 0.06 m. This result does not contradict the discussion derived from Eq. (4).

### 4. Connectivity Preservation Experiment

In this section, we have detailed the experimental results of connectivity preservation control proposed in a previous study [3]. The control algorithm is almost the same as that listed in [3], but the expressions are modified to match the local coordinate system of robot $i$.

Based on the results of the measurement experiments
Experimental Multi-Robot System for Network Connectivity Controls

4.1. Distance Preservation (Two Robots)

First, we conducted an experiment to preserve the connectivity of two robots using the control method described above. The initial distance of the robots was 0.5 m apart from each other. One robot was controlled using the algorithms detailed above, and the other robot was kept stationary in this experiment.

As a result, time evolution of the distance between the robots is depicted in Fig. 10, and the second smallest eigenvalue of the Laplacian is depicted in Fig. 11. The distance data were measured using a mobile robot. We can see that the distance gradually increased from 0.5 m due to the repulsive velocity (Eq. (6)); however, it does not exceed the predetermined maximum distance 1.35 m. We can confirm that the preservation control worked efficiently in this experiment.

4.2. Connectivity Preservation (Three Robots)

Later, we conducted an experiment to preserve connectivity using three robots. The initial positions of the robots were set as depicted in Fig. 12. Two robots (robot 2 and robot 3) were controlled using the algorithms detailed above, and one robot (robot 1) was kept stationary in this experiment.

The experimental results are presented in Figs. 13–15. The experimental data corresponding to robots 1, 2, and 3 are indicated using red, green, and blue markers, respectively. The trajectories of the mobile robots observed by robot 1 are depicted in Fig. 13. From the trajectories depicted in Fig. 13, we can see that the distances from robot 1 to 2, and that from robot 1 to 3 do not exceed the predetermined maximum distance $R = 1.35$ m. The time evolution of the distances observed by robot 2 is depicted in Fig. 14. We can see that the distance from robot 2 to robot 1 does not exceed the maximum distance $R = 1.35$ m, although it increases because of the repulsive
velocity in the beginning of the experiment. We can also see that the communication link between robots 2 and 3 is lost because the distance from robot 2 to 3 exceeds the maximum distance.

Although the communication link \( a_{23} \) is lost, the network is still connected because links \( a_{12} \) and \( a_{13} \) are alive. We can confirm this fact from Fig. 15, which illustrates the time evolutions of the algebraic connectivity of the network in this experiment.

5. Conclusion

In this study, we have presented a robot system developed for experiments on network connectivity controls and the results of the basic experiments. Experiments on the measurement of neighbor robots’ positions demonstrated the accuracy of the developed system. The position estimation error is less than 0.10 m, which is not superior when compared to the absolute position sensors, but sufficient for connectivity preservation. Experiments on network connectivity preservation demonstrated that a multi-robot system consisting of three robots can achieve the network preservation control.

The limitations and future works of the developed multirobot system are detailed below. The permitted number of robots in the developed system is approximately three to six. This depends on the color recognition because the IDs of the robot were detected by considering the color of the LED. This number can be improved by employing other recognition methods in future studies. The maximum communication range of the developed system is limited to 1.5 m due to ID recognition (although the maximum measurement distance of the LiDAR is 8 m). The capability of ID recognition distance remains a topic for the future work. Additionally, a mobile ad-hoc network module will be installed in the experimental system, and adequate sensors will be adopted to match the communication range of the wireless module in future works.

As an application of the developed multirobot, we assumed a system that expands the wireless communication range autonomously. In this system, the robots move to cover a designated area while preserving the mobile ad-hoc communication network. Our developed system will be helpful to experiment in a structured and complex space, compared to the centralized experimental systems introduced in Section 1.

Acknowledgments

This work was supported by JSPS KAKENHI (grant number 20K19902). We are grateful to the previous and present members of our laboratory for helping us in assembling the developed robot.
References:


Supporting Online Materials:
