Sensors & Transducers, Vol. 236, Issue 8, August 2019, pp. 16-27

Sensors & Transducers

Published by IFSA Publishing, S. L., 2019
http://www.sensorsportal.com

Distributed Range-based Localization and Control for Swarm Robot Systems

*** Daisuke INOUE, Daisuke MURAI, Yasuhiro IKUTA, Hiroaki YOSHIDA**
Toyota Central R&D Labs., Inc., Nagakute, Aichi, Japan
Tel.: +8190-6384-4542
* E-mail: daisuke-inoue@mosk.tytlabs.co.jp

Received: 18 June 2019 /Accepted: 22 July 2019 /Published: 31 August 2019

Abstract: Existing control methods for swarm robot systems typically assume that an individual robot knows its own position. Since the direct positional information is rarely available when the number of robots is large, estimation of the position based on indirect information is necessary. In this paper, we propose a new method to restore self-location from observed distance between each robot. An exteroceptive estimation method based on the measured distance is dynamically coupled with a simple proprioceptive estimation that uses a robot's own dynamical properties. The results of numerical simulations for several scenarios show that our proposed method is more accurate that conventional methods. We also evaluated the performance of decentralized control using the information estimated using the proposed method and confirmed that the performance was improved by 15 % compared to the existing methods.

Keywords: Network localization, Swarm robot systems, Distributed control, Multi-Agent systems.

1. Introduction

Swarm robot systems consist of large numbers of simple physical robots that cooperate with each other in order to perform complicated tasks [1-2]. Recently, such systems have attracted significant amounts of attention because of their flexible adaptability to various environments and robustness against failure [3-4].

Control methods for swarm robot systems are roughly categorized into a *centralized type* or a *distributed type*. In a centralized control method, a commanding robot gathers information from all other robots, calculates the inputs of each robot, and lets them propagate again. Such an approach makes possible relatively sophisticated tasks, thanks to the information of the entire group being available for calculating the control inputs. However, since the computational cost is concentrated on the commander, calculation time could be very long. On the other hand, in a distributed control method, each robot estimates its own state using only the information exchanged with the adjacent robots, and calculates the control inputs by itself using the estimated state. In view of the characteristics of the swarm robot systems, the latter approach is desirable for large scale systems.

When discussing the information which can be sensed by each robot and exchanged between robots, we can consider relative angles, relative distances, or both (i.e., relative positions). Of these, the use of relative distance is the most feasible because it can be easily transmitted and received simply by installing a single proximity sensor using infrared rays or ultrasonic waves on each robot body [5]. In fact, inexpensive small robots equipped only with actuators and a range sensor have already been developed [6]. On the other hand, in most of swarm robot control methods proposed so far, each robot is presumed to be capable of ascertaining its own position [7-9]. For this reason, it is first necessary to estimate the self-position from the observed distance to each other. Therefore, we focus on the problem of self-position estimation from the premise that each robot can only measure its distance from other robots. This type of problem is called *range-based localization*.

Different methods have been proposed to achieve range-based localization [10-13]. In [14], the geometric conditions for the robots to estimate their position from distance were given and a localization method that worked by minimizing an evaluation function was proposed. The author of [15] pointed out that the technique in [14] is vulnerable to observation noise and proposed a more robust method in which robots first estimate their positions in a cluster of robots close to each other, and then perform coordinate transformations between clusters. We note here that many methods including [14], [15] are centralized methods, that is, there is a commanding robot which collects and retransmits the information between other robots. To overcome this problem, in [16], a distributed method via which each robot could update its own estimated position simply by communicating with the neighboring robots, was proposed. Although this method enables scalable estimations, estimation accuracy is sacrificed compared with centralized methods.

In the present study, we propose a new distributed localization method which improves the accuracy of the method in [16]. In the proposed method, two kinds of estimated values are combined by using sensor fusion technique. One estimation is based on robot's dynamic characteristics, and the other estimation is based on observed distance. The weighted sum of both is adopted as a new estimated value, in which the weights are updated at each time to minimize the error variance of the estimated value. That is, the weights are automatically adjusted to reflect more reliable estimates.

Accuracy improvements using sensor fusion technique has been well studied. For example, several methods combining information from GPS and overhead cameras [17-20], or applying feedback terms based on relative distance to estimations using odometry [21], have been proposed. However, in either case, the communication and calculation costs increase as more robots are added, and scalability is sacrificed. In contrast, in our method, the locality of the algorithm is inherited from the method in [16] and the estimated value is based on the dynamical model.

After formulating the problem in the succeeding section, we detail the algorithm procedure in Sec. 3. In Sec. 4, we show several numerical simulation results for various situations in order to compare the estimation performances of the proposed method and the conventional methods. In all situations, we confirmed that the proposed method enables us to estimate positions more accurately than conventional methods. In Sec. 5, we evaluate the performance of a decentralized control when implemented with the proposed localization method. In this task, which is referred to as *coverage*, the arrangement of the entire robots is controlled to converge such that a user specified density distribution is realized. Coverage is the basis of various tasks such as environmental monitoring, target search, lifesaving act, etc. The algorithm of Cortés, *et al.* is known as a representative control method used for solving the coverage problem [22]. This method is a distributed type and the information required for each robot is only the self-position, the position of adjacent robots, and a given density distribution. The localization method proposed in this paper is thus compatible with the algorithm developed by Cortés, *et al.* We compared the performance of coverage control employing the proposed method and the one using two other existing estimation methods, to show that the proposed method is improved by 15 %.

2. Problem Formulation

In this section, we set up a problem for formulating the range-based localization. We begin by considering N mobile robots located on two-dimensional space $\mathbb{R}^2$. Each robot has a unique ID from 1 to N, and the set of these IDs is expressed as $\mathcal{N} = \{1,2,\ldots,N\}$. Let the states of the robot $i \in \mathcal{N}$ at time $k \in \mathbb{N}$ be $s_i(k) := [x_i(k)\ y_i(k)\ \theta_i(k)]^{\mathrm{T}}$. Here, $p_i(k) := [x_i(k)\ y_i(k)]^{\mathrm{T}} \in \mathbb{R}^2$ represents a position in $\mathbb{R}^2$, and $\theta_i(k) \in (-\pi,\pi]$ represents an angle.

The purpose of this study is to provide an algorithm for each robot to estimate the self-position $\hat{s}_i(k) = [\hat{x}_i(k)\ \hat{y}_i(k)\ \hat{\theta}_i(k)]^{\mathrm{T}}$, which should be as close as possible to its true position $s_i(k)$. To accomplish this, we make the following assumption, which is the premise of localization.

1) The dynamics of each robot is described by the following typical two-wheel vehicle model:

$$\begin{aligned} x_i(k+1) &= x_i(k) + v_i(k)\cos(\theta_i(k)), \\ y_i(k+1) &= y_i(k) + v_i(k)\sin(\theta_i(k)), \\ \theta_i(k+1) &= \theta_i(k) + \omega_i(k), \end{aligned} \tag{1}$$

where $v_i(k) \in \mathbb{R}$ and $\omega_i(k) \in (-\pi,\pi]$ represent the translation and rotation speed of the robot, respectively. Defining input $u_i(k) \in \mathbb{R}^2$ as $u_i(k) := [v_i(k)\ \omega_i(k)]^{\mathrm{T}}$, and $g: \mathbb{R}^3 \times \mathbb{R}^2 \to \mathbb{R}^3$ as

$$g(x,y) := \begin{bmatrix} x_1 + y_1\cos(x_3) \\ x_2 + y_1\cos(x_3) \\ x_3 + y_2 \end{bmatrix} \tag{2}$$

Eq. (1) can be written as:

$$s_i(k+1) = g(s_i(k), u_i(k)) \tag{3}$$

2) Since none of the robots are equipped with GPS sensors or wheel encoders, they are incapable of directly obtaining their true states $s_i(k)$. Hence, it is impossible for any robot to calculate the input $u_i(k)$ using state $s_i(k)$ (i.e., state feedback).

3) Each robot communicates with a neighboring robot. Let us define the adjacency set $\mathcal{N}_i(k)$, i.e., the set of number of robots j that can communicate with robot i at time k, as

$$\mathcal{N}_i(k) = \{ j \in \mathcal{N} \mid \| p_i(k) - p_j(k) \| < D \}, \quad (4)$$

where $D \in \mathbb{R}_+$ ($\mathbb{R}_+ := \{r \mid r \in \mathbb{R},\ r > 0\}$) represents the communicable distance of the robot, and $\|\cdot\|$ represents *Euclidean norm*: for any $p \in \mathbb{R}^2, \| p \| := \sqrt{p^{\mathrm{T}} p}$. Robot i can receive the following two pieces of information from robot $j \in \mathcal{N}_i(k)$:
$d_{ij}(k) := \| p_i(k) - p_j(k) \|$; Distance of robots i and j,
$\hat{p}_j(k) \in \mathbb{R}^3$; Estimated position of robot j.

4) In [14], the geometric conditions to be satisfied in order to uniquely determine a position from the distance are shown. A network only has a unique localization if and only if its underlying graph is *generically globally rigid*. We assume that this condition is satisfied at an arbitrary time $k \in \mathbb{N}$.

3. Localization Algorithm

In this section, we present our proposed localization method based on sensor fusion technique. The description of the method is also found in [23].

3.1. Proprioceptive Estimation

First, we describe a localization method based on the dynamical characteristics of each robot shown in Eq. (1), Let $\tilde{s}_i(k) = [\tilde{x}_i(k)\ \tilde{y}_i(k)\ \tilde{\theta}_i(k)]^{\mathrm{T}}$ be the estimated value of the state of robot i at time k and $u_i(k)$ be the input, and let these be known. Then, the value of the state at the next time $\tilde{s}_i(k+1)$ is estimated by using Eq. (1) as follows:

$$\tilde{s}_i(k+1) \quad = g(\tilde{s}_i(k), u_i(k)) \quad (5)$$

Hereafter, the method based on Eq. (5) is called the *proprioceptive method*. Unlike the exteroceptive method detailed in the next subsection, since the information necessary for the estimation is closed in each robot, estimation accuracy does not depend on communication quality with the neighboring robots. Another advantage is that it allows the robot to estimate the angle, as well as its position. However, since Eq. (5) excludes a feedback term for the observed information, errors accumulate over time and the deviation from the true value eventually becomes large.

3.2. Exteroceptive Estimation

Next, we present a localization method based on communication with neighboring robots. Suppose that each robot i receives an observed distance from neighboring robot d_{ij} and an estimated position of neighboring robot $\hat{p}_j$. First, we define an evaluation function $f: \mathbb{R}^{2N} \to \mathbb{R}$ that shows the consistency of the estimated position with respect to the observed distance:

$$f(\hat{p}(k)) = \sum_{i \in \mathcal{N}} \sum_{j \in \mathcal{N}_i} (\| \hat{p}_i(k) - \hat{p}_j(k) \| - d_{ij}(k))^2,$$
$$\hat{p}(k) := [\hat{p}_1(k)^{\mathrm{T}}\ \hat{p}_2(k)^{\mathrm{T}} \cdots \hat{p}_N(k)^{\mathrm{T}}]^{\mathrm{T}} \quad (6)$$

The localization is then formulated as an optimization problem that minimizes the function f.

One of the most common methods for solving the optimization problem above is the gradient method [14]. This method first estimates an appropriate initial position and updates the estimated position via:

$$\hat{p}(k+1) = \hat{p}(k) - \alpha(k) \nabla f(\hat{p}(k)), \quad (7)$$

where $\alpha(k) \in \mathbb{R}_+$ is the design parameter called a *step length*. As $\alpha(k)$ becomes larger, the convergence of the $\hat{p}(k)$ becomes faster. However, if the value of $\alpha(k)$ is too large, then $\hat{p}(k)$ diverges. As a condition for the convergence of $\hat{p}(k)$, what is called a *Walfe condition* is known. When $\alpha(k)$ satisfies this condition, Eq. (7) converges to the local optimal solution [24].

In Eq. (7), the gradient ∇f depends on the estimated position of all robots $\hat{p}(k)$, but it consists of only the sum of the pair terms of the neighboring robots estimates. Using this property, we can decompose and rewrite Eq. (7) for each robot i as follows:

$$\hat{p}_i(k+1) \quad = \hat{p}_i(k) - \alpha_i(k) \nabla_i f_i(\hat{p}(k)), \quad (8)$$

where we defined f_i as

$$f_i(\hat{p}(k)) = \sum_{j \in \mathcal{N}_i} (\| \hat{p}_i(k) - \hat{p}_j(k) \| - d_{ij}(k))^2,$$

and $\nabla_i f$ denotes the i^{th} 1×2 block in the gradient ∇f in Eq. (7). We generalized the step length as $\alpha_i(k)$. By applying Eq. (8), each robot independently updates its estimated position with a calculation of the local evaluation function f_i. This method type is often called a *distributed gradient method* [16]. In this study, it is adopted as the *exteroceptive method*. In order to satisfy the Walfe condition, the author of [16] proposed a method of sequentially updating the weights $\alpha_i(k)$ so that the robots could communicate. For simplicity, however, we assume that weight $\alpha_i(k)$ is invariant over time, and set α as the common weight for all robots.

Since the exteroceptive method uses distances between robots, the estimation performance degrades if the distance measurement is not accurate. In addition, since only the robot position is estimated with this method, it is not possible to estimate the state including the angle of the robot. To overcome these challenges, we propose a position estimation method that works by fusing both the proprioceptive and exteroceptive methods, as will be detailed in the following subsection.

3.3. Proposed Estimation Method

In the previous subsections, we have shown the state estimation algorithms using proprioceptive and exteroceptive information. The drawback of the former is in that it is weak against modeling errors and input noises, while the latter is vulnerable to distance measurement errors. We will now propose a new method that improves performance by obtaining the weighted sum from the error variances of the two estimated values.

First, we describe *error variance minimization*. Suppose that a variable z is estimated by two different methods as $\tilde{z}, \hat{z}$ and that the errors of the methods are defined as $\tilde{\delta} = z - \tilde{z}, \hat{\delta} = z - \hat{z}$. Then, assuming that the variance of $\tilde{\delta}$ and $\hat{\delta}$ is calculated as $\tilde{V}, \hat{V}$, the weighted sum of $\tilde{z}$ and $\hat{z}$,

$$\overline{z} = \tilde{z} + (\tilde{V}^{-1} + \hat{V}^{-1})^{-1}\hat{V}^{-1}(\hat{z} - \tilde{z}), \qquad (9)$$

is known to minimize the error variance of estimated value [25]. In the followings, we evaluate the error variance of estimated values of both the proprioceptive and exteroceptive methods and then combine these values using the Eq. (9).

Error Variance of Proprioceptive Method

First, we derive an approximate expression for estimating the error distribution of the states in the proprioceptive method. The input command value for robot i at time k is denoted as $u_i'(k) \coloneqq [v_i(k)\ \omega_i(k)]^{\mathrm{T}}$, and the input actually applied is denoted as $u_i(k)$. We then write the input error as $\delta_{u_i}(k) := u_i(k) - u_i'(k)$. Additionally, we denote the true and estimated value of the state of robot i as $s_i(k)$ and $\tilde{s}_i(k)$ respectively, and the estimation error as $\delta_{s_i}(k) := s_i(k) - \tilde{s}_i(k)$. Then, the state at time $k+1$ is represented as

$$\begin{aligned} s_i(k+1) &= g(s_i(k), u_i(k)) \\ &= g(\tilde{s}_i(k) + \delta_{s_i}(k), u_i'(k) + \delta_{u_i}(k)) \end{aligned} \qquad (10)$$

By linearizing the right-hand side of Eq. (10), we obtain

$$\begin{aligned} &s_i(k+1) \\ &\approx g(\tilde{s}_i(k), u_i'(k)) + J_{s_i(k)}\delta_{s_i}(k) + J_{u_i(k)}\delta_{u_i}(k), \end{aligned} \qquad (11)$$

where we have defined

$$J_{s_i(k)} := \begin{bmatrix} 1 & 0 & -\sin(\tilde{\theta}(k))v_i'(k) \\ 0 & 1 & \cos(\tilde{\theta}(k))v_i'(k) \\ 0 & 0 & 1 \end{bmatrix},$$
$$J_{u_i(k)} := \begin{bmatrix} \cos(\tilde{\theta}(k)) & 0 \\ \sin(\tilde{\theta}(k)) & 0 \\ 0 & 1 \end{bmatrix}.$$

Therefore, the estimation error $\delta_{s_i}(k+1)$ caused by Eq. (5) is written as the following expression:

$$\begin{aligned} \delta_{s_i}(k+1) &:= s_i(k+1) - \tilde{s}_i(k+1) \\ &= s_i(k+1) - g(\tilde{s}_i(k), u_i'(k)) \\ &\approx J_{s_i(k)}\delta_{s_i}(k) + J_{u_i(k)}\delta_{u_i}(k). \end{aligned}$$

We then assume the following in order to obtain an approximate representation of the error variance using the above equation:

1) Estimated error of state $\delta_{s_i}(k)$ is a random variable, and its distribution is a Gaussian distribution with an average of 0 and a variance of $\tilde{V}_{s_i}(k)$.

2) Similarly, let $\delta_{u_i}(k)$ be a random variable that follows the Gaussian distribution $N(0, V_{u_i})$ at all times k.

3) $\delta_{s_i}(k)$ and $\delta_{u_i}(k)$ are independent. Then, if we know $\tilde{V}_{s_i}(0)$ and V_{u_i} for all $i \in \mathcal{N}$, the error variance of state at time $k+1$ is approximated using information at time k as

$$\tilde{V}_{s_i}(k+1) = J_{s_i(k)}\tilde{V}_{s_i}(k)J_{s_i(k)}^{\mathrm{T}} + J_{u_i(k)}V_{u_i}J_{u_i(k)}^{\mathrm{T}} \qquad (12)$$

Error Variance of Exteroceptive Method

Next, we derive the error variance of state in the exteroceptive method. Let $d_{ij}(k)$ be the true distance between robots i and j at time k, let $d'_{ij}(k)$ be the observed distance, and define the error as $\delta_{d_{ij}}(k) := d_{ij}(k) - d'_{ij}(k)$. We further assume that the true and estimated position of robot i is $p_i(k)$ and $\hat{p}_i(k)$ respectively, and the estimation error is $\delta_{p_i}(k) := p_i(k) - \hat{p}_i(k)$. In Eq. (8), if $d_{ij}(k)$ is correctly observed, the estimated value $\hat{p}_i(k)$ is expected to converge to the true value $p_i(k)$ as $k \to \infty$. In addition, if $\hat{p}_i(k) = p_i(k)$ holds, in the next iteration in Eq. (8), $\hat{p}_i(k+1) = p_i(k+1)$ is expected to hold as well. However, this is not the case if $d_{ij}(k)$ is *not* correctly observed. In this situation, the true position $p_i(k)$ is written as

$$\begin{aligned} &p_i(k+1) \\ &= p_i(k) - \alpha\nabla_i \sum_{j\in\mathcal{N}_i} (\parallel p_i(k) - p_j(k) \parallel - d_{ij}(k))^2 \\ &= \hat{p}_i(k) + \delta_{p_i}(k) \\ &-\alpha\nabla_i \sum_{j\in\mathcal{N}_i} \left(\parallel \left(\hat{p}_i(k) + \delta_{p_i}(k)\right) - \left(\hat{p}_j(k) + \delta_{p_j}(k)\right) \parallel \right. \\ &\left. -(d'_{ij}(k) + \delta_{d_{ij}}(k)\right)^2 \end{aligned} \qquad (13)$$

By linearizing the right-hand side of Eq (13), we obtain

$$
\begin{aligned}
&p_i(k+1)\\
&= \hat{p}_i(k) - \alpha \nabla_i \sum_{j\in\mathcal{N}_i} (\parallel \hat{p}_i(k) - \hat{p}_j(k) \parallel - d'_{ij}(k))^2\\
&+ J_{p_i(k)}\delta_{p_i}(k) + \sum_{j\in\mathcal{N}_i} J_{p_{ij}(k)}\,\delta_{p_j}(k) + \sum_{j\in\mathcal{N}_i} J_{d_{ij}(k)}\,\delta_{d_{ij}}(k),
\end{aligned}
\tag{14}
$$

where we defined

$$
\begin{aligned}
&J_{p_i(k)} := \begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix} - \alpha \sum_{j\in\mathcal{N}_i} P_{ij},\\
&J_{p_{ij}(k)} := \alpha P_{ij},\\
&P_{ij} := \begin{bmatrix} 1 - \frac{d_{ij}(k)\hat{y}_{ij}(k)^2}{\parallel \hat{p}_{ij}(k) \parallel^3} & \frac{d_{ij}(k)\hat{x}_{ij}(k)\hat{y}_{ij}(k)}{\parallel \hat{p}_{ij}(k) \parallel^3}\\ \frac{d_{ij}(k)\hat{x}_{ij}(k)\hat{y}_{ij}(k)}{\parallel \hat{p}_{ij}(k) \parallel^3} & 1 - \frac{d_{ij}(k)\hat{x}_{ij}(k)^2}{\parallel \hat{p}_{ij}(k) \parallel^3}\end{bmatrix},\\
&J_{d_{ij}(k)} := \alpha \begin{bmatrix} \frac{\hat{x}_{ij}(k)}{\parallel \hat{p}_{ij}(k) \parallel}\\ \frac{\hat{y}_{ij}(k)}{\parallel \hat{p}_{ij}(k) \parallel}\end{bmatrix},\\
&\hat{x}_{ij}(k) := \hat{x}_i(k) - \hat{x}_j(k),\\
&\hat{y}_{ij}(k) := \hat{y}_i(k) - \hat{y}_j(k),\\
&\hat{p}_{ij}(k) := \hat{p}_i(k) - \hat{p}_j(k)
\end{aligned}
$$

Therefore, the estimation error $\delta_{p_i}(k+1)$ resulting from Eq. (8) is written as:

$$
\begin{aligned}
&\delta_{p_i}(k+1)\\
&:= p_i(k+1) - \hat{p}_i(k+1)\\
&\approx J_{p_i(k)}\delta_{p_i}(k) + \sum_{j\in\mathcal{N}_i} J_{p_{ij}(k)}\,\delta_{p_j}(k) + \sum_{j\in\mathcal{N}_i} J_{d_{ij}(k)}\,\delta_{d_{ij}}(k)
\end{aligned}
\tag{15}
$$

As in the proprioceptive method, the following assumptions are made in order to obtain an approximate representation of error variance:

1) For all $i \in \mathcal{N}$, the estimated error of position $\delta_{p_i}(k)$ is a random variable, and its distribution is Gaussian with an average of 0 and a variance of $\hat{V}_{p_i}(k)$.

2) $\delta_{d_{ij}}(k)$ is also a random variable that follows the Gaussian distribution $N(0, V_{d_{ij}})$.

3) $\delta_{p_i}(k), \delta_{d_{ij}}(k)$ are independent. Then, if we know $\hat{V}_{p_i}(0)$ and $V_{d_{ij}}$ for all $i, j \in \mathcal{N}$, the error variance of estimated position at time $k+1$ is approximated as:

$$
\begin{aligned}
&\hat{V}_{p_i}(k+1) = J_{p_i(k)}\hat{V}_{p_i}(k)J_{p_i(k)}^{\mathrm{T}}\\
&+ \sum_{j\in\mathcal{N}_i} J_{p_{ij}}\,\hat{V}_{p_j}(k)J_{p_{ij}}^{\mathrm{T}} + \sum_{j\in\mathcal{N}_i} J_{d_{ij}(k)}\,V_{d_{ij}}J_{d_{ij}(k)}^{\mathrm{T}}
\end{aligned}
\tag{16}
$$

Fusion of Estimated Values

We use Eq. (9) to combine the variances in the estimation methods calculated above. Note that the exteroceptive method only estimates the position and does not have angle information. Therefore, let the error variance of estimated angle be infinite and define the expanded variance $\hat{V}_{s_i}(k)$ as

$$
\hat{V}_{s_i}(k) := \begin{bmatrix}\hat{V}_{p_i}(k) & 0\\ 0 & \infty\end{bmatrix} \tag{17}
$$

Using this, the merged estimated value $\bar{s}(k)$ is given as the following expression:

$$
\begin{aligned}
&\bar{s}_i(k) = \tilde{s}_i(k)\\
&+ (\tilde{V}_{s_i}(k)^{-1} + \hat{V}_{s_i}(k)^{-1})^{-1}\hat{V}_{s_i}(k)^{-1}(\hat{s}_i(k) - \tilde{s}_i(k))
\end{aligned}
\tag{18}
$$

4. Numerical Examples

In this section, we show several numerical results for various practical situations in order to compare the estimation performances of the proprioceptive method, exteroceptive method, and a method of “fusing" the two values.

When the position of robot i at time k is written as $p_i(k)$ and the estimated position is written as $\hat{p}_i(k)$, the following two indices are considered for evaluating the estimation quality:

- Function J_1 that represents the distance between the estimated value and the true value:

$$
J_1 = \frac{1}{N}\sum_{i\in\mathcal{N}} \parallel p_i(k) - \hat{p}_j(k) \parallel \tag{19}
$$

- Function J_2 representing the consistency with respect to observed distance:

$$
J_2 = \frac{1}{N}\sum_{i\in\mathcal{N}, j\in\mathcal{N}_i} | \parallel \hat{p}_i(k) - \hat{p}_j(k) \parallel - d_{ij}(k)| \tag{20}
$$

We also consider the mean over the calculation time,

$$
\bar{J}_1 = \frac{1}{K}\sum_{k=0}^{K} J_1, \tag{21}
$$

$$
\bar{J}_2 = \frac{1}{K}\sum_{k=0}^{K} J_2, \tag{22}
$$

where $K \in \mathbb{N}$ represents the termination time of the simulation.

The parameters commonly set in this section are as follows: We first assume the communicable distance D is equal to 5.0 m. In the exteroceptive method, the step length of the gradient method is $\alpha = 0.1$, which is a reasonable choice because the typical length-scale of our system is the communicable distance D. Unless otherwise stated, the following values are used for the error variance of signal used in the fusion of the weights in the proposed method:

$$
\begin{aligned}
\tilde{V}_{s_i}(0) &= \mathrm{diag}(0.09, 0.09, 0.09),\\
\hat{V}_{p_i}(0) &= \mathrm{diag}(0.09, 0.09),\\
V_{u_i} &= \mathrm{diag}(0.002, 0.002),\\
V_{d_{ij}} &= 0.09,
\end{aligned}
\tag{23}
$$

Here it should be noted that, in the numerical calculations, the white noise following the Gaussian distribution whose average is 0 and whose variance is represented by Eq. (23) is supposed to be applied additively.

4.1. Performance in the Stationary Case

In this subsection, we report on a comparison of position estimation performance for a swarm consisting of nine robots. The initial positions were set in a lattice pattern, that is, $x_i(0) \in \{-2.0,\ 0.0,\ 2.0\}$, $y_i(0) \in \{-2.0,\ 0.0,\ 2.0\}$. The input of all robots was set to 0:

$$u_i(k) = [0.0 \quad 0.0]^{\mathrm{T}}$$

Fig. 1 shows the trajectory of each estimated robot position using all three estimation methods. In the proprioceptive method, since the initial estimation state is not updated, an estimation error appears and increases steadily. The exteroceptive method estimates a value deviating from the true value due to the influence of the observed distance error. In contrast, the proposed method estimates values very close to the true positions.

In Fig. 2, the time responses of the two evaluation functions J_1, J_2 defined by Eq. (19), Eq. (20) are shown. Fig. 2 shows that the proposed method estimates positions closer to their true values than both the proprioceptive and exteroceptive methods. In particular, in Fig. 2(a), it can be seen that the error of the exteroceptive method is large, whereas the error is suppressed in the proposed method. This reflects the proprioceptive information that the robots are at rest at their initial positions.

We have also checked the computational time for each estimation method. Table 1 compares the elapsed time of 1000 step simulation. It shows that the time taken in the proposed method is almost equal to the sum of those taken in the other two methods, confirming that the combination of the two methods does not cause any unreasonable overhead.

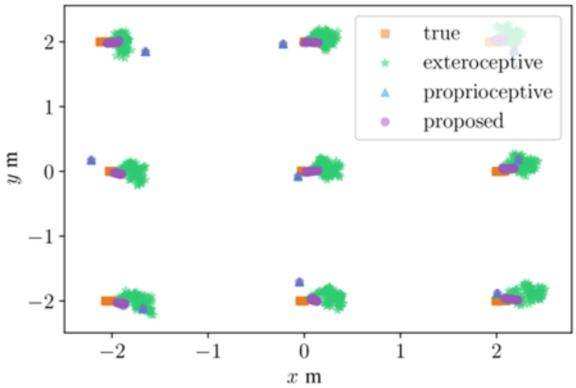

Fig. 1. Trajectories of nine robots. All robots are stopped at their initial position. The positions estimated by the proposed method (purple circle) are compared to the actual location (orange square). The positions estimated by the exteroceptive method only (green star) and the proprioceptive method only (blue triangle) are also shown for comparison.

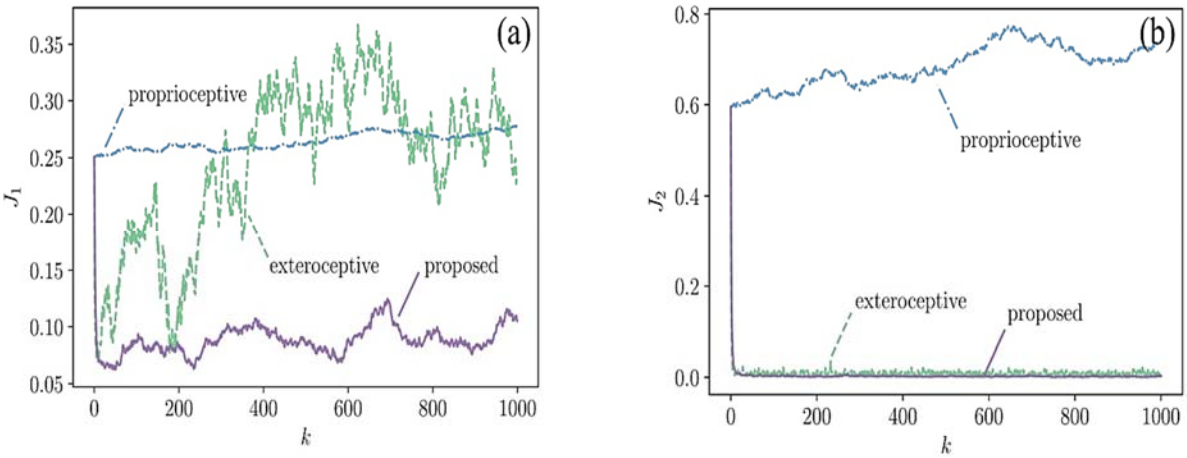

Fig. 2. Histories of two evaluation functions: (a) J_1, (b) J_2. The value of the present method (purple solid) is compared to that of the exteroceptive method (green dashed) and of the proprioceptive method (blue dash-dot).

Table 1. Elapsed time of 1000 step simulation in the stationary case.

Method	Elapsed time [s]
Proprioceptive	5.66
Exteroceptive	7.23
Proposed	13.84

4.2. Performance in the Dynamic Case

As in Sec. 4.1, nine robots are arranged in a grid pattern. Three units of $y_i(0) = -2$ turn to the left, three units of $y_i(0) = 2$ turn to the right, and three units $y_i(0) = 0$ remain at rest on their initial positions.

Specifically, the input $u_i(k)$ to each robot is represented by

$$u_i(k) = \begin{cases} [0.0 \quad 0.0]^{\mathrm{T}} & \text{for } y_i(0) = 0, \\ [0.05 \quad 0.05]^{\mathrm{T}} & \text{for } y_i(0) = -2, \\ [0.05 \quad -0.05]^{\mathrm{T}} & \text{for } y_i(0) = 2 \end{cases}$$

Fig. 3 shows the trajectories of each estimated robot position obtained by the three estimation methods. Here, it can be seen that the proposed method position estimates are somewhat more accurate than the other two methods. The estimated value of the proprioceptive method is biased over the entire time due to the influence of the initial error and input noise. The exteroceptive method estimates a value deviating from the true value because of the observed distance error influence.

Fig. 4 plots the time responses of two evaluation functions J_1, J_2 defined by Eq. (19), Eq. (20), which again confirm the accurate estimation of the proposed method compared to both the proprioceptive and exteroceptive methods.

We have also checked the computational time for this case. Table 2 shows the elapsed time of 1000 step simulation for each estimation method. Similarly to the previous subsection, the time taken for the proposed method is about the same as the sum of the time taken for the component methods.

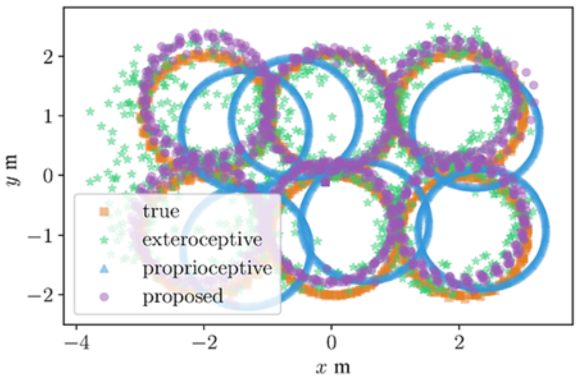

Fig. 3. Trajectories of nine robots. Three units of $y_i(0) = -2$ turn to the left, three units of $y_i(0) = 2$ turn to the right, and three units $y_i(0) = 0$ remain on their initial positions. The positions estimated by the proposed method (purple circle) are compared to the actual location (orange square). The positions estimated by the exteroceptive method only (green star) and the proprioceptive method only (blue triangle) are also shown for comparison.

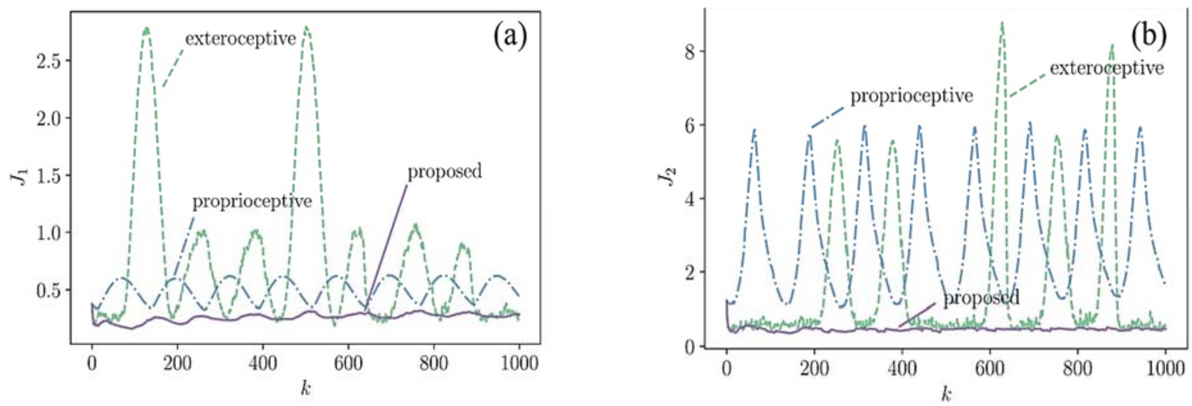

Fig. 4. Histories of two evaluation functions: (a) J_1, (b) J_2. The value of the present method (purple solid) is compared to that of the exteroceptive method (green dashed) and of the proprioceptive method (blue dash-dot).

Table 2. Elapsed time of 1000 step simulation in the dynamic case.

Method	Elapsed time [s]
Proprioceptive	5.99
Exteroceptive	6.67
Proposed	11.68

4.3. Input Noise Effect

Here we examine the performance while varying the magnitude of the standard deviation σ_u ($V_{u_i} = \mathrm{diag}(\sigma_u^2, \sigma_u^2)$) of the noise applied to input u_i ($\forall i \in \mathcal{N}$). The initial positions and inputs are the same as in Sec. 4.2.

In Fig. 5, the value of two evaluation functions $\bar{J}_1, \bar{J}_2$ defined as Eqs. (21), (22) with respect to σ_u are shown. The termination time is $K = 200$ unless otherwise is stated. Fig. 5(a) shows that, when σ_u is small in the proposed method, the value of Eq. (21) is also small. However, as σ_u gets larger, it approaches the value of Eq. (21) in the exteroceptive method. This is because of the weight adjustment function, which puts higher weight on the exteroceptive information than on the proprioceptive information, as the variance of the input noise increases. On the other hand, in Fig. 5(b), the proposed method makes the evaluation function J_2 small even when σ_u is large. Thus, we can conclude that our proposed method gives a value that is most consistent with the geometric relationship of all robots.

4.4. Observation Noise Effect

Next, we investigate the impact of the applied noise by examining the observed distance d_{ij} ($\forall i \in \mathcal{N}, j \in \mathcal{N}_i$) while changing the magnitude of the standard deviation σ_d ($V_{d_{ij}} = \sigma_d^2$). The initial positions and inputs are the same as in Sec. 4.2.

In Fig. 6, we show the value of two evaluation functions $\bar{J}_1, \bar{J}_2$ defined by Eqs. (21), (22) with respect to σ_d. The estimation accuracy degrades as the noise increases in the exteroceptive method. On the other hand, in the proprioceptive and proposed methods, no remarkable adverse effect is observed. In addition, in Fig. 6(a), (b), regardless of the value of σ_d, the proposed method takes a smaller value than the proprioceptive estimate. These results allow us to conclude that the proposed method is the most robust of the three methods against observed distance errors.

4.5. Effect of Number of Robots

Finally, we examine the effect of the number of robots N on the estimation performance. The initial configuration of the robots is generated as a uniform random number taking a value range in $(-2.5, 2.5)^2$. The input $u_i(k)$ to each robot is defined by

$$u_i(k) = \begin{cases} [0.0 \quad 0.0]^{\mathrm{T}} & \text{for } i = 3n \quad (n \in \mathbb{N}), \\ [0.05 \quad 0.05]^{\mathrm{T}} & \text{for } i = 3n+1 \quad (n \in \mathbb{N}), \\ [0.05 \quad -0.05]^{\mathrm{T}} & \text{for } i = 3n+2 \quad (n \in \mathbb{N}) \end{cases}$$

The values of two evaluation functions $\bar{J}_1, \bar{J}_2$ defined in Eqs. (21), (22) with respect to N are shown in Fig. 7. As in Fig. 7(a), the error of the proposed method decreases as N increases. This accuracy improvement stems from the fact that each robot has larger amounts of information for larger N. In Fig. 7(b), $\bar{J}_2$ increases as the value of N increases in all estimation methods. This is because the evaluation function consists of the sum of i and j, which increases J_2 with order $O(N)$. Note that this is simply the sum of errors over N while the accuracy is held fixed and does not indicate performance degradation. In Fig. 7, regardless of the value of N, the proposed method yields a smaller value than the other two methods. These results confirm the higher robustness of the proposed method against an increase in the number of robots.

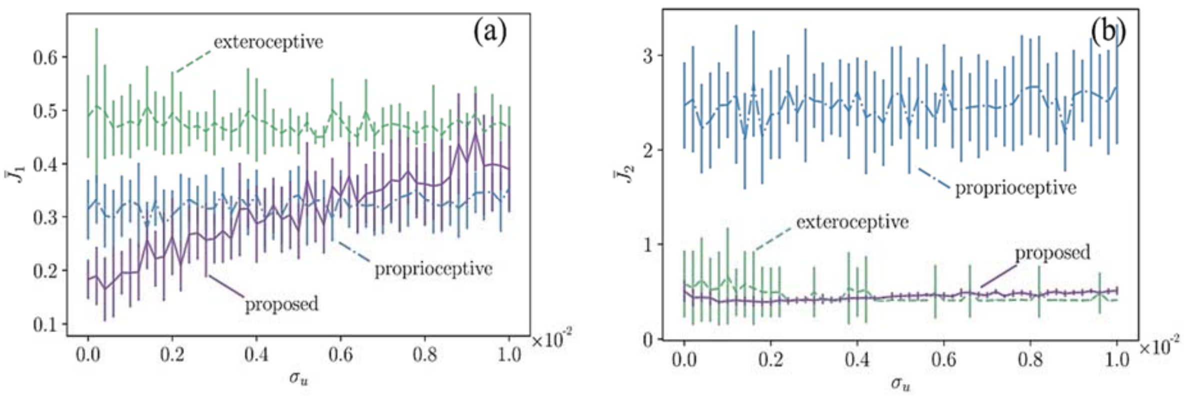

Fig. 5. Value of two evaluation functions: (a) J_1, (b) J_2, compared with the magnitude of input noise σ_u. Each line represents the average value obtained when the numerical calculation was performed 10 times, and the length of the vertical bar represents the standard deviation of 10 times. The value of the present method (purple solid) is compared to that of the exteroceptive method (green dashed) and of the proprioceptive method (blue dash-dot).

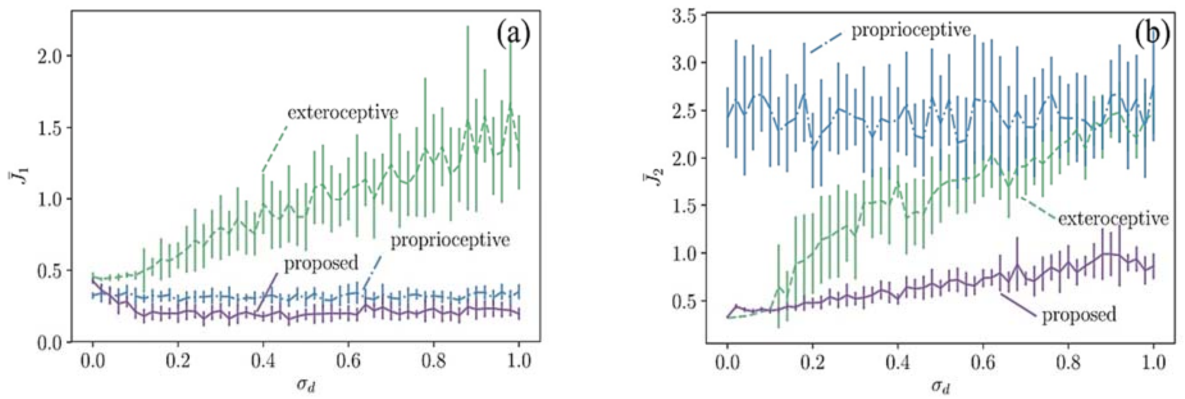

Fig. 6. Value of two evaluation functions: (a) J_1, (b) J_2, compared with the magnitude of observation noise σ_d. Each line represents the average value obtained when the numerical calculation was performed 10 times, and the length of the vertical bar represents the standard deviation of 10 times. The value of the present method (purple solid) is compared to that of the exteroceptive method (green dashed) and of the proprioceptive method (blue dash-dot).

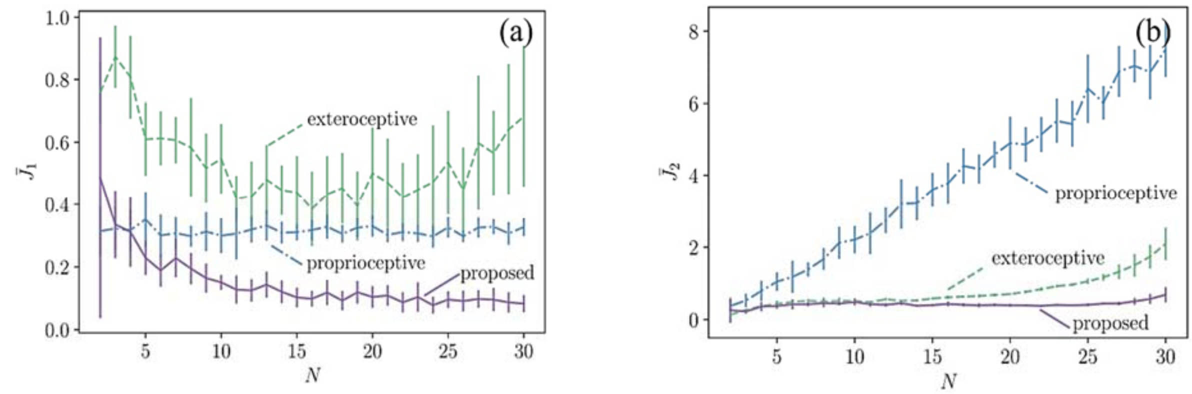

Fig. 7. Value of two evaluation functions: (a) J_1, (b) J_2, compared with the number of robots N. Each line represents the average value obtained when the numerical calculation was performed 10 times, and the length of the vertical bar represents the standard deviation of 10 times. The value of the present method (purple solid) is compared to that of the exteroceptive method (green dashed) and of the proprioceptive method (blue dash-dot).

5. Performance Evaluation when Combined with Distributed Control

In this section, we implement distributed control of the swarm robot using the proposed localization method and evaluate its performance. Here we employ the method proposed by Cortés, *et al.* as coverage control algorithm, which is one of the basic techniques of distributed control. We evaluate the performance of coverage control with using the proposed localization method, in comparison with the performance using the other localization methods.

5.1. Coverage Control

First, we describe the coverage control algorithm. Let $\phi: \mathbb{R}^2 \to \mathbb{R}$ be the density distribution representing the target arrangement of robots. To place each robot in accordance with this distribution ϕ, consider the following evaluation function:

$$H(p(k)) := \sum_{i=1}^{N} \int_{\mathcal{V}(p_i(k))} \| z - p_i(k) \|^2 \, \phi(z) dz, \quad (24)$$

where the integration domain $\mathcal{V}(p_i)$ is called a *Voronoi region* of the robot i and is defined as

$$\mathcal{V}(p_i) := \{ z \in \mathbb{R}^2 \mid \| z - p_i \| \le \| z - p_j \| \ \ \forall j \in \mathcal{N} \text{ s.t. } j \neq i \}$$

The Voronoi region of robot i is a set of points on $\mathbb{R}^2$ space, whose distance to p_i is not greater than their distance to the other robots position p_j. The integral term of Eq. (24) represents the distance between all points $z \in \mathcal{V}(p_i)$ and the robot position p_i, weighted by the function ϕ. Reducing Eq. (24) means to move each robot i to a place on its Voronoi region, where the density distribution ϕ takes a large value. As a result, the arrangement of the entire robot will reproduce the given density distribution ϕ.

Cortés proved that the evaluation function Eq. (24) is locally minimized by setting the target position $r_i(k) \in \mathbb{R}^2$ of the robot i as follows:

$$r_i(k) = \frac{\int_{\mathcal{V}(p_i(k))} z \, \phi(z) dz}{\int_{\mathcal{V}(p_i(k))} \phi(z) dz}, \quad (25)$$

and by letting each robot i move toward this position $r_i(k)$. This equation means that the target position of each robot is given by the center of gravity of each Voronoi region weighted by the density distribution ϕ. To calculate Eq. (25), each robot should be able to calculate the Voronoi region using the self-position and the positions of the other robots at each time, and perform numerical integration on each Voronoi region.

Next, we describe a method for each robot to move from its own position p_i to a target position r_i. Since robots represented by Eq. (1) are systems with nonholonomic constraints, they cannot perform simple position feedback control. Therefore, we consider the following two position control methods:

1) *Angle feedback control:* The input $v_i(k)$ of the robot to the forward direction has a constant value $v' \in \mathbb{R}_+$. The input $\omega_i(k)$ of the turning direction is provided as an input for feeding back an error between the target attitude and the current attitude, which is calculated from the current position and the target position:

$$\omega_i(k) = K(\mathrm{atan2}(r_i(k) - p_i(k)) - \theta_i(k)),$$

where $\mathrm{atan2}: \mathbb{R}^2 \to (-\pi, \pi]$ is a function for calculating an angle formed by a two-dimensional vector, and $K \in \mathbb{R}_+$ is a design parameter representing a feedback gain.

2) *Randomized control:* The input $v_i(k)$ of the robot to the forward direction has a constant value $v' \in \mathbb{R}_+$. The input of the turning direction $\omega_i(k)$ is determined as follows. The distance between the target

position and the current position is compared at the current time and the time one step past, and if the distance of current time is smaller, the robot goes straight, and if it is larger, the robot rotates at random angle. This is represented as follows:

$$\omega_i(k) = \begin{cases} 0 & \text{if } e_i(k) \le e_i(k-1), \\ \frac{2}{3}\pi & \text{if } e_i(k) > e_i(k-1),\ w_i(k) = 1, \\ -\frac{2}{3}\pi & \text{if } e_i(k) > e_i(k-1),\ w_i(k) = -1, \end{cases}$$

where we defined $e_i(k) := \| p_i(k) - r_i(k) \|$ and $w_i(k)$ represents the random variable following the Bernoulli distribution:

$$\begin{cases} \Pr(w_i(k) = 1) & = 0.5, \\ \Pr(w_i(k) = -1) & = 0.5 \end{cases}$$

5.2. Numerical Simulation

For the performance evaluation of coverage control, we compare the control results based on the each estimation method described above; proprioceptive method, exteroceptive method, and proposed method.

The information required for each robot to calculate Eq. (25) is only the self-position $p_i(k)$ and the position of the surrounding robot $p_j(k)$, and the density distribution ϕ. Therefore, the localization method proposed in the previous section makes it possible to calculate $\overline{r}_i(k)$, which is the estimated value of $r_i(k)$ given in Eq. (25).

In control using angle feedback, each robot needs to know its own angle $\theta_i(k)$ in order to calculate the inputs. For this reason, this control is possible only in the proprioceptive method and the proposed method, in which the angle is estimated, but is *not* possible in the exteroceptive method. On the other hand, the randomized control can be implemented using exteroceptive method because the required information is only the self-position $p_i(k)$ and the algorithm for random number generation.

Whereas the control using angle information allows each robot to go directly to the target position, the randomized control determines the direction by trial and error. Therefore, the latter generally takes longer to reach the target positions. In this reason, the exteroceptive estimation in which angle feedback cannot be performed is combined with randomized control, and the remaining two methods are combined with angle feedback control. That is, we consider combinations of the following three estimation methods and feasible control methods:

a) Proprioceptive estimation with angle feedback control;

b) Exteroceptive estimation with randomized control;

c) Proposed estimation with angle feedback control.

In the practical numerical simulation, we use the following function as target distribution $\phi: \mathbb{R}^2 \to \mathbb{R}$ for robot placement:

$$\phi(z) = \exp\left(30\sin\left(\frac{\| z \|}{2} + 1\right)\right) \tag{26}$$

This function is visualized in Fig. 8. The result of the coverage control is expected to exhibit the robots arranged on the perimeter of a circle centered on the origin. The number of robots used was $N = 10$, and the initial layout of each robot was generated from uniform random numbers in $[-2.5\ 2.5]\ \mathrm{m}^2$. The generated initial configuration is shown in Fig. 9.

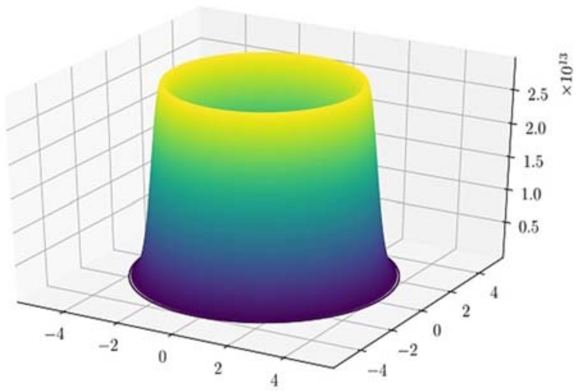

Fig. 8. Target density function ϕ.

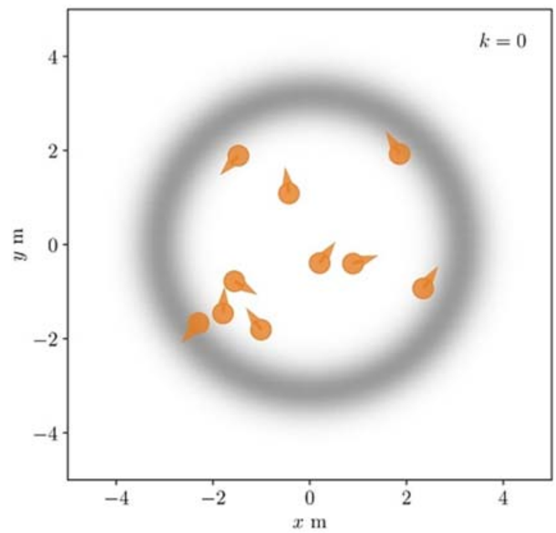

Fig. 9. Initial placement of 10 robots. Arrows indicate the attitude of each robot. Target density function ϕ is displayed as a mesh-plot.

The communication range of the robot is set as $D = 5.0$ m. In the two control methods, the forward input v' of the robot is commonly set as $v' = 0.01$ m/s. The gain K in angle feedback is designed as $K = 0.1$. In the exteroceptive estimation, the step length is set as $\alpha = 0.1$. For the noise of input, observation and intial estimation, we set for each as

$\sigma_u = 0.02, \sigma_d = 0.02, \sigma_s = 0.02$ and these values are used for the error variance of signal in the proposed method.

Under the above settings, Fig. 10 shows the transition of the placement of robots in the simulations using each combination of estimation and control method. We also plotted the history of the evaluation function Eq. (24) in Fig. 11. In calculating the value of H, the computational space is divided into 2500 to perform numerical integration.

Among the three methods (a–c), the value of the evaluation function in the proposed method (c) decreases most rapidly and the steady state value in (c) is also the smallest. Looking into more detail, the evaluation function of the proposed method (c) goes below the proprioceptive method (a). Comparing the steady-state values averaged over $k \in [1000{,}2000]$, (a) is 1.07×10^{16} and (c) is 9.14×10^{15}, that is the proposed method reduces the evaluation function value by 15 %. Moreover, in the method (a), the developed formation is broken again as time passes. This is because the Eq. (10) does not include a correction term based on the observation and thus the estimation errors accumulate over time. The randomized control (b) is slower in forming the distribution than the control using angle information (a, c). Compared with the angle feedback control which allows each robot to go directly to the target position, the randomized control determines the direction by trial and error, and hence, the latter takes longer to reach the target positions. The proposed method (c) can overcome the problems of the methods (a, b), and thus we conclude that it is the best among the three.

6. Conclusion

In this paper, we proposed a range-based localization method for swarm robot systems. We began by presuming that the robots were only capable of measuring their distances from neighboring robots and transmitting the estimated states of themselves. The proposed method linearly combines two estimated values, one obtained by means of the distributed gradient method $\hat{s}_i(k)$ and the other from the robot dynamics $\tilde{s}_i(k)$. The weight in combination is determined in a way that ensures the error variance is minimized, as shown in Eq. (18). That is, the weight is automatically adjusted to reflect more reliable estimates.

We then performed numerical simulations for various practical situations and verified the performance of our proposed algorithm. More specifically, we considered situations with

1) Robots at rest;
2) Moving robots;
3) Varying input noise;
4) Varying observation noise;
5) Different numbers of robots.

In all cases, we confirmed that the proposed method enables us to estimate positions more accurately than both the proprioceptive and exteroceptive methods.

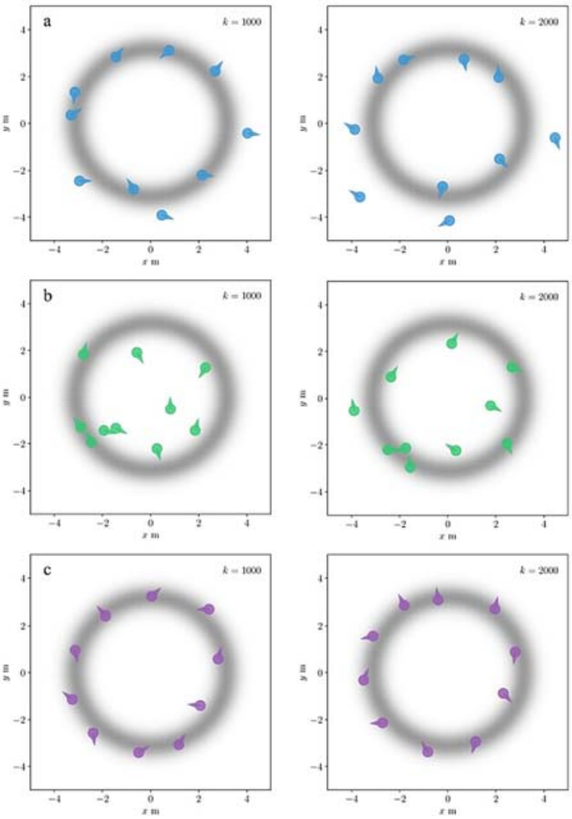

Fig. 10. Robots placement when using different combination of estimation method and control method.
(a): Proprioceptive estimation and Angle feedback control,
(b): Exteroceptive estimation and Randomized control,
(c): Proposed estimation and Angle feedback control.

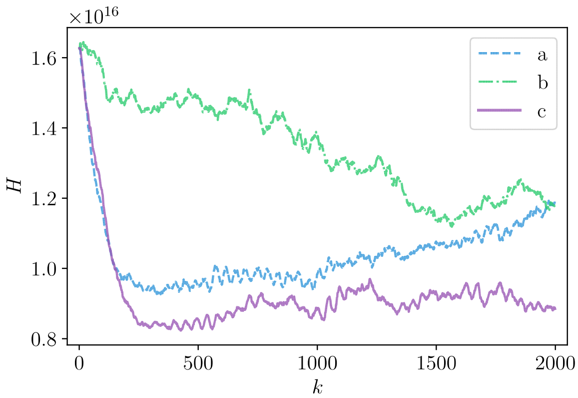

Fig. 11. Time evolution of evaluation function Eq. (26). The value of the present method (purple solid) is compared to that of the proprioceptive method (blue dashed) and of the exteroceptive method (green dash-dot).

We also executed simulations of coverage control proposed by Cortés, which is a representative algorithm of distributed control. We confirmed that the proposed method realizes more accurate position control than each of the proprioceptive and exteroceptive method, i.e., the control performance when using our method is 15 % superior to the other cases (see plot (c) in Fig. 11 in comparison with plot (a, b)).

One of the extensions of our investigation would be comparison with other recent methods such as one in [21]. In addition, implementation of the proposed method using real robots and experimental evaluation of the performance are also in the scope of our future studies.

References

[1]. J. C. Barca, Y. A. Sekercioglu, Swarm Robotics Reviewed, *Robotica*, Vol. 31, Issue 3, 2013, pp. 345–359.

[2]. M. Brambilla, E. Ferrante, M. Birattari, M. Dorigo, Swarm Robotics: A Review from the Swarm Engineering Perspective, *Swarm Intell.*, Vol. 7, Issue 1, 2013, pp. 1–41.

[3]. Y. Tan, Handbook of Research on Design, Control, and Modeling of Swarm Robotics, 1st ed. Hershey, PA, USA: IGI Global, 2015.

[4]. Y. U. Cao, A. S. Fukunaga, A. Kahng, Cooperative Mobile Robotics: Antecedents and Directions, *Auton. Robots*, Vol. 4, Issue 1, 1997, pp. 7–27.

[5]. F. Blais, Review of 20 Years of Range Sensor Development, *J. Electron. Imaging*, Vol. 13, Issue 1, 2004, pp. 231–244.

[6]. M. Rubenstein, C. Ahler, R. Nagpal, Kilobot: A Low Cost Scalable Robot System for Collective Behaviors, in *Proceedings of the IEEE International Conference on Robotics and Automation*, 2012, pp. 3293–3298.

[7]. H. Wang, Y. Guo, A Decentralized Control for Mobile Sensor Network Effective Coverage, in *Proceedings of the 7th World Congress on Intelligent Control and Automation (WCICA'08)*, 2008, pp. 473–478.

[8]. S. Bandyopadhyay, S.-J. Chung, F. Y. Hadaegh, Probabilistic and Distributed Control of a Large-Scale Swarm of Autonomous Agents, *IEEE Trans. Robot.*, Vol. 33, Issue 5, 2017, pp. 1103–1123.

[9]. M. Rubenstein, W. M. Shen, Scalable Self-Assembly and Self-Repair in a Collective of Robots, in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2009, pp. 1484–1489.

[10]. P. Biswas, T.-C. Liang, K.-C. Toh, Y. Ye, T.-C. Wang, Semidefinite Programming Approaches for Sensor Network Localization with Noisy Distance Measurements, *IEEE Trans. Autom. Sci. Eng.*, Vol. 3, Issue 4, 2006, pp. 360–371.

[11]. Y. Shang, W. Ruml, Improved MDS-based localization, in *Proceedings of the IEEE INFOCOM 2004 Conference*, Vol. 4, 2004, pp. 2640–2651.

[12]. B. Dil, S. Dulman, P. Havinga, Range-Based Localization in Mobile Sensor Networks, in *Proceedings of the 3rd European Workshop on Wireless Sensor Networks*, 2006, pp. 164–179.

[13]. C. Zhou, T. Xu, H. Dong, Distributed Locating Algorithm MDS-MAP (LF) Based on Low-Frequency Signal, *Comput. Sci. Inf. Syst.*, Vol. 12, Issue 4, 2015, pp. 1289–1305.

[14]. T. Eren, *et al.*, Rigidity, Computation, and Randomization in Network Localization, in *Proceedings of the IEEE INFOCOM 2004 Conference*, Vol. 4, 2004, pp. 2673–2684.

[15]. D. Moore, J. Leonard, D. Rus, S. Teller, Robust Distributed Network Localization with Noisy Range Measurements, in *Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems*, Baltimore, MD, USA, 2004, pp. 50–61.

[16]. G. C. Calafiore, L. Carlone, M. Wei, Distributed Optimization Techniques for Range Localization in Networked Systems, in *Proceedings of the 49th IEEE Conference on Decision and Control (CDC)*, 2010, pp. 2221–2226.

[17]. B. S. Choi, J. W. Lee, J. J. Lee, K. T. Park, A Hierarchical Algorithm for Indoor Mobile Robot Localization Using RFID Sensor Fusion, *IEEE Trans. Ind. Electron.*, Vol. 58, Issue 6, June 2011, pp. 2226–2235.

[18]. R. Deming, Sensor Fusion for Swarms of Unmanned Aerial Vehicles Using Modeling Field Theory, in *Proceedings of the International Conference on Integration of Knowledge Intensive Multi-Agent Systems,* 2005, pp. 122–127.

[19]. H. M. La, W. Sheng, Distributed Sensor Fusion for Scalar Field Mapping Using Mobile Sensor Networks, *IEEE Trans. Cybern.*, Vol. 43, Issue 2, April 2013, pp. 766–778.

[20]. G. Mao, S. Drake, B. D. O. Anderson, Design of an Extended Kalman Filter for UAV Localization, in *Proceedings of the Information, Decision and Control*, 2007, pp. 224–229.

[21]. A. Martinelli, F. Pont, R. Siegwart, Multi-Robot Localization Using Relative Observations, in *Proceedings of the IEEE International Conference on Robotics and Automation*, 2005, pp. 2797–2802.

[22]. J. Cortes, S. Martinez, T. Karatas, F. Bullo, Coverage Control for Mobile Sensing Networks, *IEEE Trans. Robot. Autom.*, Vol. 20, Issue 2, 2004, pp. 243–255.

[23]. D. Inoue, D. Murai, Y. Ikuta, Y. Horoaki, Distributed Range-based Localization for Swarm Robot Systems using Sensor-Fusion technique, in *SENSORNETS'2019*, Prague, Czech Republic, 26-27 February 2019, pp.13-22.

[24]. J. Nocedal, Numerical Optimization, *Springer*, 2006.

[25]. F. Gustafsson, Statistical Sensor Fusion, *Studentlitteratur,* 2010.