Distributed Range-based Localization for Swarm Robot Systems using Sensor-fusion Technique

Daisuke Inoue, Daisuke Murai, Yasuhiro Ikuta and Hiroaki Yoshida
Toyota Central R&D Labs., Inc., Nagakute, Aichi, Japan

Keywords: Network Localization, Swarm Robotics, Sensor Fusion.

Abstract: Herein, we present a localization method for swarm robot systems that relies solely on measured distances from surrounding robots. Using the sensor fusion technique, 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. Our method strictly preserves the locality of algorithm. The results of numerical simulations for several scenarios show that our proposed method is more accurate that conventional methods.

1 INTRODUCTION

Swarm robot systems consist of large numbers of simple physical robots that cooperate with each other in order to perform complicated tasks as a group (Barca and Sekercioglu, 2013; Brambilla et al., 2013; Sahin, 2004). Recently, because such systems have attracted significant amounts of attention due to positive features such as their flexible adaptability to various environments and robustness against failure (Tan, 2015; Cao et al., 1997), there has been a concurrent increase in the proposed task execution control techniques required for such robot systems (Cheah et al., 2009; Xie and Wang, 2007; Spears et al., 2004; Castillo et al., 2012). In most of those control methods, each robot is presumed to be capable of ascertaining its own position (Wang and Guo, 2008; Bandyopadhyay et al., 2017; Rubenstein and Shen, 2009) by means of GPS data or via image recognition using cameras installed overhead or the like. However, GPS is limited in terms of resolution and indoor use utility (Kourogi et al., 2006), and image recognition involves problems in that the amount of communication data increases along with the number of robots (Luo et al., 2014). Therefore, in order to maximize the utility of swarm robot systems, it is necessary for each robot to be capable of estimating its own position autonomously and locally without relying on external systems, such as information exchanges between swarm robots.

When discussing the behaviors of swarm robot systems, the abilities and functions of each robot must be fixed. It should also be possible to exchange various elements of information such as relative angles, relative distances, or both (i.e., relative positions) among the robots. 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 (Blais, 2004). In fact, inexpensive robots equipped only with actuators and a range sensor have already been developed (Rubenstein et al., 2012). For this reason, this study focuses on the problem of self-position estimation from the premise that each robot can only measure its distance from other swarm robots. This type of problem is called *range-based localization*.

Range-based localization has attracted attention in tandem with the rapid growth of interest in swarm robot systems and different methods have been proposed to achieve it (Biswas et al., 2006; Shang et al., 2004; Dil et al., 2006; Zhou et al., 2015; Shang and Ruml, 2004). In (Eren et al., 2004), the geometric conditions that a robot needs to satisfy in order to estimate its position from distance information were given and a position estimation method that worked by minimizing the evaluation function of the difference between observed and estimated positions was proposed. However, the author of (Moore et al., 2004) pointed out that the technique in (Eren et al., 2004) is vulnerable to observed distance 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 in many methods, including (Eren et al., 2004; Moore et al., 2004),

Inoue, D., Murai, D., Ikuta, Y. and Yoshida, H.
Distributed Range-based Localization for Swarm Robot Systems using Sensor-fusion Technique.
DOI: 10.5220/0007258800130022
In *Proceedings of the 8th International Conference on Sensor Networks (SENSORNETS 2019)*, pages 13-22
ISBN: 978-989-758-355-1

it is assumed that there is a so-called *anchor node* that collects and calculates information of the surrounding robots and then retransmits the calculated result to the surrounding devices. This means that as the number of robots increases, robot node computational costs and memory capacity requirements also increase, which results in communication route congestion. To overcome this problem, in (Calafiore et al., 2010), a distributed method via which a robot could update its own estimated position simply by communicating with the robots closest to it was proposed. This method enables scalable estimations to be performed independently of the number of robots.

In the present study, we propose a method of range-based localization in which each robot uses information obtained both from its own dynamical characteristics and relative distance measurements. More specifically, we aim to improve the self-position estimation method proposed in (Calafiore et al., 2010) by combining it with an estimation method based on dynamical characteristics.

Accuracy improvements using multiple information types has been well studied in the field of sensor fusion. For example, several methods combining information from GPS and overhead cameras (Choi et al., 2011; Deming, 2005; La and Sheng, 2013; Mao et al., 2007), or applying feedback terms based on relative distance to estimations using odometry (Martinelli et al., 2005), 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 (Calafiore et al., 2010) and the estimated value is based on the dynamical model. We begin by combining the two estimated values by regarding both estimation methods as dynamical characteristics, and then determine the values of weights coefficients in a way that minimizes error variances. Next, after problem formulation, we detail the algorithm procedure. We also show numerical results to confirm the improvements of our proposed method over conventional methods.

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 (2D) space. 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 point in 2D, and $\theta_i(k) \in (-\pi,\pi]$ represents an angle.

The purpose of this study is to provide an algorithm for each robot i to use when producing a self-position estimation $\hat{s}_i(k) = [\hat{x}_i(k)\ \hat{y}_i(k)\ \hat{\theta}_i(k)]^{\mathrm{T}}$, 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.

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

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

where $\varepsilon \in \mathbb{R}_+$ $(:= \{r; r \in \mathbb{R},\ r > 0\})$ is a parameter corresponding to the sampling period, while $v_i(k) \in \mathbb{R}$ and $\omega_i(k) \in \mathbb{R}$ represent the translation and rotation speed of the robot, respectively. Equation (1) represents a model in which each robot moves in 2D space according to the input of the translation and rotation speed. 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

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

Eq. (1) can be written as:

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

(ii) 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).

(iii) 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:

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

where $D \in \mathbb{R}_+$ represents the communicable distance of the robot, and $\|\cdot\|$ represents Euclidean norm: for any $p \in \mathbb{R}^2$,

$$\|p\| := \sqrt{p^{\mathrm{T}}p}. \tag{5}$$

Robot i can receive the following two pieces of information from robot $j \in \mathcal{N}_i(k)$:

(a) $d_{ij}(k) = \|p_i(k) - p_j(k)\|$: Euclidean distance of robots i and j,

(b) $\hat{p}_j(k)$: Estimated position of robot j.

(iv) In (Eren et al., 2004), the geometric conditions that need to be satisfied in order to uniquely determine a position using distance information are shown. In that method, 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

3.1 Proprioceptive Estimation

First, we will 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:

$$\tilde{s}_i(k+1) = g(\tilde{s}_i(k), u_i(k)). \tag{6}$$

Hereafter, the method based on Eq. (6) is called the *proprioceptive method*. Unlike the exteroceptive method detailed in the next section, 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. (6) 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 based on the Distributed Gradient Method

Here, we present a localization method based on communication with neighboring robots. Suppose that each robot i receives an observed distance value 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:

$$f(\hat{p}(k)) = \sum_{i \in \mathcal{N}} \sum_{j \in \mathcal{N}_i} (\|\hat{p}_i(k) - \hat{p}_j(k)\| - d_{ij}(k))^2, \tag{7}$$

$$\hat{p}(k) := [\hat{p}_1(k)^{\mathrm{T}}\ \hat{p}_2(k)^{\mathrm{T}}\ \cdots\ \hat{p}_N(k)^{\mathrm{T}}]^{\mathrm{T}}. \tag{8}$$

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

$$\text{minimize } f(\hat{p}(k)). \tag{9}$$

One of the most common methods for solving the optimization problem above is the gradient method (Eren et al., 2004). This method first estimates an appropriate initial position and updates the estimated position via:

$$\hat{p}(k+1) = \hat{p}(k) - \alpha(k)\nabla f(\hat{p}(k)), \tag{10}$$

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. (10) converges to the local optimal solution (Nocedal, 2006).

In Eq. (10), the gradient ∇f depends on the estimated position of all robots $\hat{p}(k)$, but it consists of the sum of the pair terms of the estimated positions that are located in the neighborhood. Using this property, we can decompose it and rewrite Eq. (10) for each robot i as follows:

$$\hat{p}_i(k+1) = \hat{p}_i(k) - \alpha_i(k)\nabla_i f_i(\hat{p}(k)), \tag{11}$$

where we defined f_i as

$$f_i(\hat{p}(k)) = \sum_{j \in \mathcal{N}_i} (\|\hat{p}_i(k) - \hat{p}_j(k)\| - d_{ij}(k))^2, \tag{12}$$

and $\nabla_i f$ denotes the i-th 1×2 block in the gradient ∇f in Eq. (10). We generalized the step length as $\alpha_i(k)$.

By applying Eq. (11), 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* (Calafiore et al., 2010). In this study, it is adopted as the *exteroceptive method*. In order to satisfy the Walfe condition, the author of (Calafiore et al., 2010) 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 section.

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 will 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 $\hat{z}$,

$$\bar{z} = \tilde{z} + (\tilde{V}^{-1} + \hat{V}^{-1})^{-1}\hat{V}^{-1}(\hat{z} - \tilde{z}), \tag{13}$$

is known to minimize the error variance of estimated value (Gustafsson, 2010). In the following subsection, we evaluate the error variance of position estimated using both the proprioceptive and exteroceptive methods and then combine the estimation values using the Eq. (13).

3.3.1 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 $u'_i(k)$, 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

$$\begin{aligned} s_i(k+1) &= g(s_i(k), u_i(k)) \\ &= g(\tilde{s}_i(k) + \delta_{s_i}(k), u'_i(k) + \delta_{u_i}(k)). \end{aligned} \tag{14}$$

By linearizing the right-hand side of Eq. (14), we obtain

$$s_i(k+1) \approx g(\tilde{s}_i(k), u'_i(k)) + J_{s_i(k)}\delta_{s_i}(k) + J_{u_i(k)}\delta_{u_i}(k), \tag{15}$$

where we have defined

$$J_{s_i(k)} := \begin{bmatrix} 1 & 0 & -\varepsilon\sin(\tilde{\theta}(k))v'_i(k) \\ 0 & 1 & \varepsilon\cos(\tilde{\theta}(k))v'_i(k) \\ 0 & 0 & 1 \end{bmatrix}, \tag{16}$$

$$J_{u_i(k)} := \begin{bmatrix} \varepsilon\cos(\tilde{\theta}(k)) & 0 \\ \varepsilon\sin(\tilde{\theta}(k)) & 0 \\ 0 & \varepsilon \end{bmatrix}. \tag{17}$$

Therefore, the estimation error $\delta_{s_i}(k+1)$ caused by Eq. (6) is written as the following expression:

$$\begin{aligned} \delta_{s_i}(k+1) &:= s_i(k+1) - \tilde{s}_i(k+1) \\ &= s_i(k+1) - g(\tilde{s}_i(k), u'_i(k)) \\ &\approx J_{s_i(k)}\delta_{s_i}(k) + J_{u_i(k)}\delta_{u_i}(k). \end{aligned} \tag{18}$$

We then assume the following in order to obtain an approximate representation of the error variance using the above equation: (i) 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 $\tilde{V}_{s_i}(k)$. (ii) Similarly, let $\delta_{u_i}(k)$ be a random variable that follows the Gaussian distribution $N(0, V_{u_i})$ at all times k. (iii) $\delta_{s_i}(k)$ 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

$$\tilde{V}_{s_i}(k+1) = J_{s_i(k)}\tilde{V}_{s_i}(k)J^{\mathrm{T}}_{s_i(k)} + J_{u_i(k)}V_{u_i}J^{\mathrm{T}}_{u_i(k)}. \tag{19}$$

3.3.2 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. (11), 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. (11), $\hat{p}_i(k+1) = p_i(k+1)$ 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

$$\begin{aligned} &p_i(k+1) \\ &= p_i(k) - \alpha\nabla_i \sum_{j\in\mathcal{N}_i} (\|p_i(k) - p_j(k)\| - d_{ij}(k))^2 \\ &= \hat{p}_i(k) + \delta_{p_i}(k) \\ &\quad - \alpha\nabla_i \sum_{j\in\mathcal{N}_i} \big(\|(\hat{p}_i(k) + \delta_{p_i}(k)) - (\hat{p}_j(k) + \delta_{p_j}(k))\| \\ &\quad - (d'_{ij}(k) + \delta_{d_{ij}}(k))\big)^2. \end{aligned} \tag{20}$$

By linearizing the right-hand side of Eq (20), we obtain

$$\begin{aligned} &p_i(k+1) \\ &= \hat{p}_i(k) - \alpha\nabla_i \sum_{j\in\mathcal{N}_i} (\|\hat{p}_i(k) - \hat{p}_j(k)\| - d'_{ij}(k))^2 \\ &+ J_{p_i(k)}\delta_{p_i}(k) + \sum_{j\in\mathcal{N}_i} J_{p_{ij}(k)}\delta_{p_j}(k) + \sum_{j\in\mathcal{N}_i} J_{d_{ij}(k)}\delta_{d_{ij}}(k), \end{aligned} \tag{21}$$

where we defined

$$J_{p_i(k)} := \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} - \alpha \sum_{j \in \mathcal{N}_i} P_{ij}, \tag{22}$$

$$J_{p_{ij}(k)} := \alpha P_{ij}, \tag{23}$$

$$P_{ij} := \begin{bmatrix} 1 - \frac{d_{ij}(k)\hat{y}_{ij}(k)^2}{\|\hat{p}_{ij}(k)\|^3} & \frac{d_{ij}(k)\hat{x}_{ij}(k)\hat{y}_{ij}(k)}{\|\hat{p}_{ij}(k)\|^3} \\ \frac{d_{ij}(k)\hat{x}_{ij}(k)\hat{y}_{ij}(k)}{\|\hat{p}_{ij}(k)\|^3} & 1 - \frac{d_{ij}(k)\hat{x}_{ij}(k)^2}{\|\hat{p}_{ij}(k)\|^3} \end{bmatrix}, \tag{24}$$

$$J_{d_{ij}(k)} := \alpha \begin{bmatrix} \frac{\hat{x}_{ij}(k)}{\|\hat{p}_{ij}(k)\|} \\ \frac{\hat{y}_{ij}(k)}{\|\hat{p}_{ij}(k)\|} \end{bmatrix}, \tag{25}$$

$$\hat{x}_{ij}(k) := \hat{x}_i(k) - \hat{x}_j(k), \tag{26}$$

$$\hat{y}_{ij}(k) := \hat{y}_i(k) - \hat{y}_j(k), \tag{27}$$

$$\hat{p}_{ij}(k) := \hat{p}_i(k) - \hat{p}_j(k). \tag{28}$$

Therefore, the estimation error $\delta_{p_i}(k+1)$ resulting from Eq. (11) is written as:

$$\begin{aligned} &\delta_{p_i}(k+1) \\ &:= p_i(k+1) - \hat{p}_i(k+1) \\ &\approx J_{p_i(k)}\delta_{p_i}(k) + \sum_{j \in \mathcal{N}_i} J_{p_{ij}(k)}\delta_{p_j}(k) + \sum_{j \in \mathcal{N}_i} J_{d_{ij}(k)}\delta_{d_{ij}}(k). \end{aligned} \tag{29}$$

As in the previous subsection, the following assumptions are made in order to obtain an approximate representation of error variance: (i) 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 $\hat{V}_{p_i}(k)$. (ii) $\delta_{d_{ij}}(k)$ is also a random variable that follows the Gaussian distribution $N(0, V_{d_{ij}})$. (iii) $\delta_{p_i}(k), \delta_{d_{ij}}(k)$ 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:

$$\begin{aligned} \hat{V}_{p_i}(k+1) &= J_{p_i(k)}\hat{V}_{p_i}(k)J_{p_i(k)}^{\mathrm{T}} \\ &+ \sum_{j \in \mathcal{N}_i} J_{p_{ij}}\hat{V}_{p_j}(k)J_{p_{ij}}^{\mathrm{T}} + \sum_{j \in \mathcal{N}_i} J_{d_{ij}(k)}V_{d_{ij}}J_{d_{ij}(k)}^{\mathrm{T}}. \end{aligned} \tag{30}$$

3.3.3 Fusion of Estimated Values

We use Eq. (13) 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

$$\hat{V}_{s_i}(k) := \begin{bmatrix} \hat{V}_{p_i}(k) & 0 \\ 0 & \infty \end{bmatrix}. \tag{31}$$

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

$$\begin{aligned} \bar{s}_i(k) &= \tilde{s}_i(k) \\ &+ (\tilde{V}_{s_i}(k)^{-1} + \hat{V}_{s_i}(k)^{-1})^{-1}\hat{V}_{s_i}(k)^{-1}(\hat{s}_i(k) - \tilde{s}_i(k)). \end{aligned} \tag{32}$$

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 $\tilde{p}_i(k)$, the following two indices are considered for evaluating the estimation quality:

(i) Function J_1 that represents the distance between the estimated value and the true value:

$$J_1 = \frac{1}{N} \sum_{i \in \mathcal{N}} \|p_i(k) - \hat{p}_j(k)\|. \tag{33}$$

(ii) Function J_2 representing the consistency with respect to observed distance:

$$J_2 = \frac{1}{N} \sum_{i \in \mathcal{N}, j \in \mathcal{N}_i} |\|\hat{p}_i(k) - \hat{p}_j(k)\| - d_{ij}(k)|, \tag{34}$$

We also consider the mean over the calculation time,

$$\bar{J}_1 = \frac{1}{K} \sum_{k=0}^{K} J_1, \tag{35}$$

$$\bar{J}_2 = \frac{1}{K} \sum_{k=0}^{K} J_2, \tag{36}$$

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. In the dynamics of the robot Eq. (1), the sampling period ε, which should be sufficiently small compared to a typical velocity of robots (in the following simulation, velocity is typically set to 5.0), is assumed that $\varepsilon = 0.01$. 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

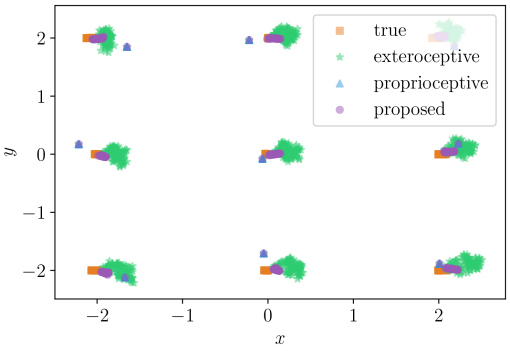

Figure 1: Trajectories of nine robots. All robots are shown stopped at their initial position. The positions estimated by the proposed method (purple circle) are compared to the actual location (orange square). The positions estimated by the exteroceptive method only (green star) and the proprioceptive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

the fusion of the weights in the proposed method:

$$\tilde{V}_{p_i}(0) = \mathrm{diag}(0.09, 0.09), \tag{37}$$
$$\hat{V}_{p_i}(0) = \mathrm{diag}(0.09, 0.09), \tag{38}$$
$$V_{u_i} = \mathrm{diag}(0.04, 0.04), \tag{39}$$
$$V_{d_{ij}} = 0.09, \tag{40}$$

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 Eqs. (37)-(40) 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:

$$u_i(k) = \begin{bmatrix} 0.0 & 0.0 \end{bmatrix}^{\mathrm{T}}. \tag{41}$$

Figure 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 Eqs. (33), (34) are shown.

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

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.

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

$$u_i(k) = \begin{cases} \begin{bmatrix} 0.0 & 0.0 \end{bmatrix}^{\mathrm{T}} & \text{for } y_i(0) = 0, \\ \begin{bmatrix} 5.0 & 5.0 \end{bmatrix}^{\mathrm{T}} & \text{for } y_i(0) = -2, \\ \begin{bmatrix} 5.0 & -5.0 \end{bmatrix}^{\mathrm{T}} & \text{for } y_i(0) = 2. \end{cases} \tag{42}$$

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.

Figure 4 plots the time responses of two evaluation functions J_1, J_2 defined by Eqs. (33), (34), 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

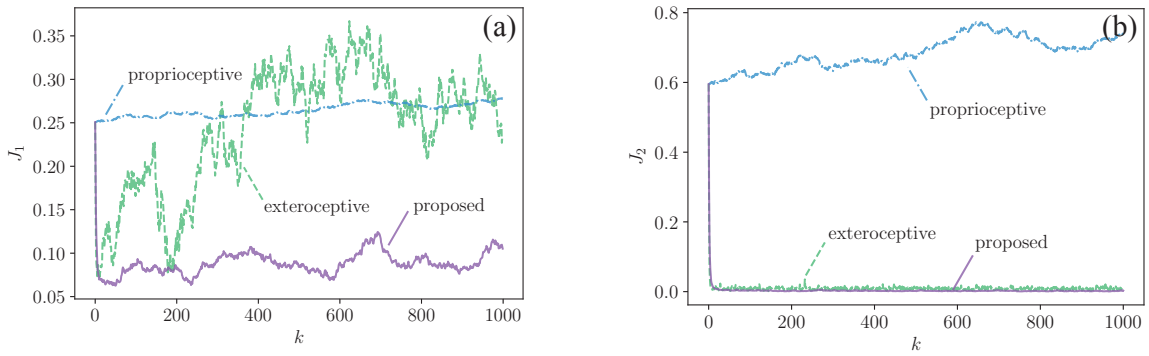

Figure 2: Histories of two evaluation functions: (a) J_1, (b) J_2. The value of the present method (purple solid) is compared to that of the exteroceptive method (green dashed) and of the proprioceptive method (blue dash-dot).

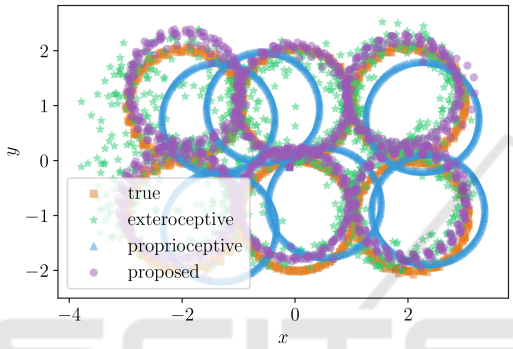

Figure 3: Trajectories of nine robots. Three units of $y_i(0) = -2$ turn to the left, three units of $y_i(0) = 2$ turn to the right, and three units $y_i(0) = 0$ remain on their initial positions. The positions estimated by the proposed method (purple circle) are compared to the actual location (orange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

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

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.

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. (35), (36) with respect to σ_u are shown. The termination time is $K = 200$ unless otherwise is stated.

Figure 5-(a) shows that, when σ_u is small in the proposed method, the value of Eq. (35) is also small. However, as σ_u gets larger, it approaches the value of Eq. (35) 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. 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. (35), (36) 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 Figs. 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.

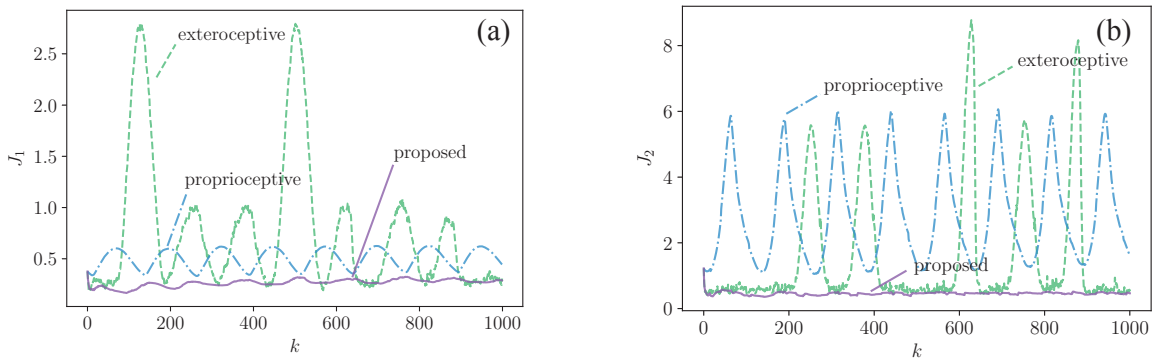

Figure 4: Histories of two evaluation functions: (a) J_1, (b) J_2. The value of the present method (purple solid) is compared to that of the exteroceptive method (green dashed) and of the proprioceptive method (blue dash-dot).

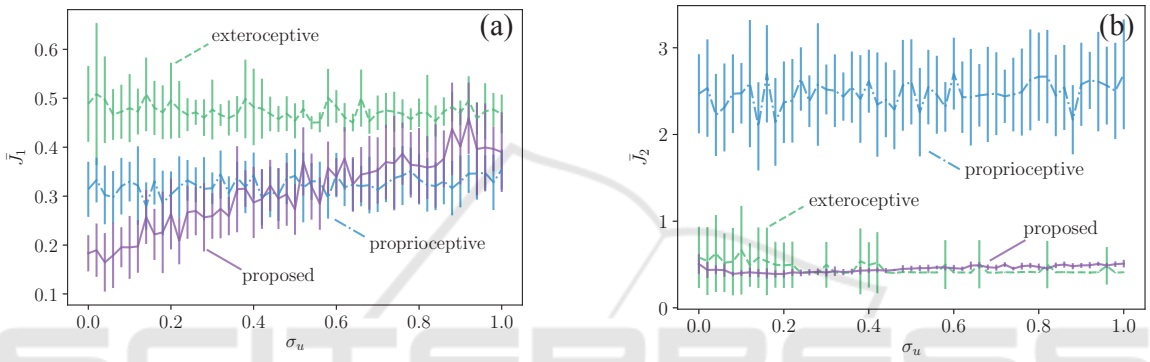

Figure 5: Value of two evaluation functions: (a) $\bar{J}_1$, (b) $\bar{J}_2$, compared with the magnitude of input noise σ_u. Each line represents the average value obtained when the numerical calculation was performed 10 times, and the length of the vertical bar represents the standard deviation of 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).

4.5 Effect of the 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

$$u_i(k) = \begin{cases} \begin{bmatrix} 0.0 & 0.0 \end{bmatrix}^{\mathrm{T}} & \text{for } i = 3n \quad (n \in \mathbb{N}), \\ \begin{bmatrix} 5.0 & 5.0 \end{bmatrix}^{\mathrm{T}} & \text{for } i = 3n+1 \quad (n \in \mathbb{N}), \\ \begin{bmatrix} 5.0 & -5.0 \end{bmatrix}^{\mathrm{T}} & \text{for } i = 3n+2 \quad (n \in \mathbb{N}). \end{cases} \tag{43}$$

The values of two evaluation functions $\bar{J}_1, \bar{J}_2$ defined in Eqs. (35), (36) 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 Figs. 7-(a), (b), 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.

5 CONCLUSION

In this paper, we proposed a self-position estimate method that uses the distance between the agents of swarm robot systems. We began by presuming that none of the robots were equipped with GPS or wheel encoders, but were capable of measuring their distances from neighboring robots. 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

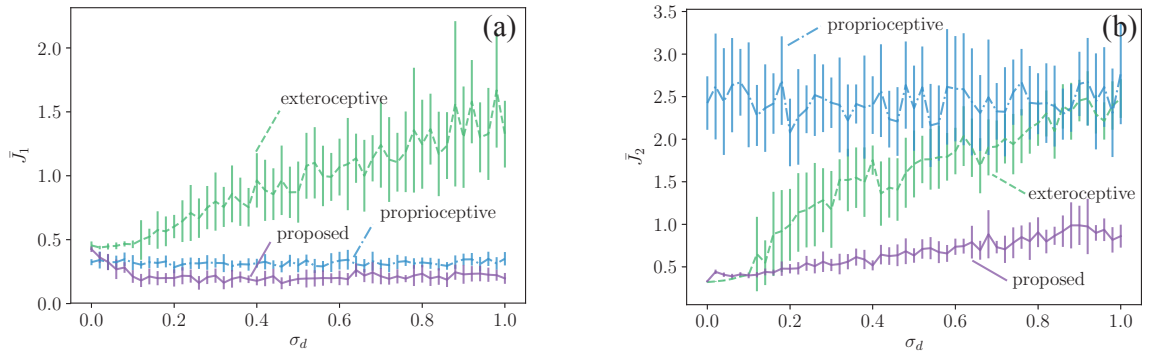

Figure 6: Value of two evaluation functions: (a) $\bar{J}_1$, (b) $\bar{J}_2$, compared with the magnitude of observation noise σ_d. Each line represents the average value when the numerical calculation was performed 10 times, and the length of the vertical bar represents the standard deviation of 10 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).

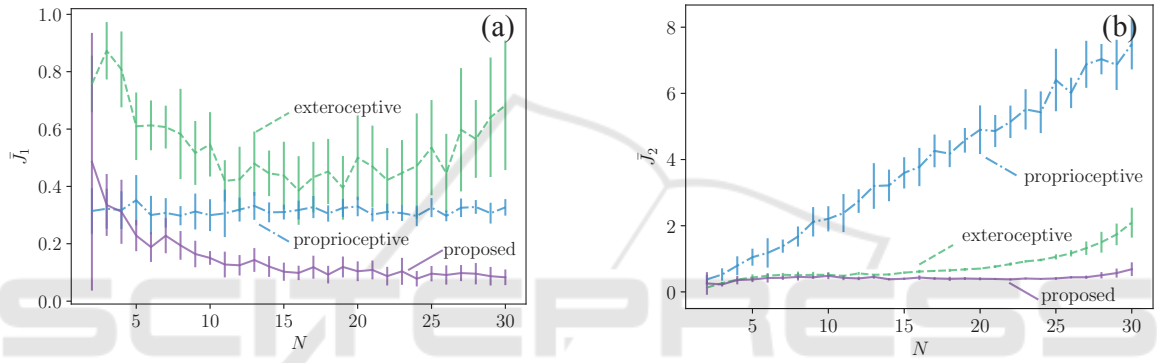

Figure 7: Values of two evaluation functions: (a) $\bar{J}_1$, (b) $\bar{J}_2$, compared with the magnitude of the number of robots N. Each line represents the average value when a numerical calculation was performed 10 times, and the length of the vertical bar represents the standard deviation of the 10 calculation results. The value of the present method (purple solid) is compared to that of the exteroceptive method (green dashed) and that of the proprioceptive method (blue dash-dot).

weight in combination is determined in a way that ensures the error variance is minimized, as shown in Eq. (32). That is, in situations where the exteroceptive method is more reliable, it increases autonomously the weight of $\hat{s}_i(k)$.

We then performed numerical simulations for various practical situations and verified the performance of our proposed algorithm. Specifically, we considered situations with (i) robots at rest, (ii) moving robots, (iii) varying input noise, (iv) varying observation noise, and (v) 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.

One of the extensions of our investigation would be comparison with other recent methods such as one in (Martinelli et al., 2005). 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. We also plan to verify the control performance of the proposed method when combined with existing control algorithms for swarm robot systems.

REFERENCES

Bandyopadhyay, S., Chung, S.-J., and Hadaegh, F. Y. (2017). Probabilistic and distributed control of a large-scale swarm of autonomous agents. *IEEE Transactions on Robotics*, 33(5):1103–1123.

Barca, J. C. and Sekercioglu, Y. A. (2013). Swarm robotics reviewed. *Robotica*, 31(3):345–359.

Biswas, P., Liang, T.-C., Toh, K.-C., Ye, Y., and Wang, T.-C. (2006). Semidefinite programming approaches for sensor network localization with noisy distance measurements. *IEEE transactions on automation science and engineering*, 3(4):360–371.

Blais, F. (2004). Review of 20 years of range sensor development. *Journal of electronic imaging*, 13(1):231–244.

Brambilla, M., Ferrante, E., Birattari, M., and Dorigo, M.

(2013). Swarm robotics: A review from the swarm engineering perspective. *Swarm Intelligence*, 7(1):1–41.

Calafiore, G. C., Carlone, L., and Wei, M. (2010). Distributed optimization techniques for range localization in networked systems. In *49th IEEE Conference on Decision and Control (CDC)*, pages 2221–2226.

Cao, Y. U., Fukunaga, A. S., and Kahng, A. (1997). Cooperative mobile robotics: Antecedents and directions. *Autonomous robots*, 4(1):7–27.

Castillo, O., Martínez-Marroquín, R., Melin, P., Valdez, F., and Soria, J. (2012). Comparative study of bio-inspired algorithms applied to the optimization of type-1 and type-2 fuzzy controllers for an autonomous mobile robot. *Information sciences*, 192:19–38.

Cheah, C. C., Hou, S. P., and Slotine, J. J. E. (2009). Region-based shape control for a swarm of robots. *Automatica*, 45(10):2406–2411.

Choi, B. S., Lee, J. W., Lee, J. J., and Park, K. T. (2011). A Hierarchical Algorithm for Indoor Mobile Robot Localization Using RFID Sensor Fusion. *IEEE Transactions on Industrial Electronics*, 58(6):2226–2235.

Deming, R. (2005). Sensor fusion for swarms of unmanned aerial vehicles using modeling field theory. In *International Conference on Integration of Knowledge Intensive Multi-Agent Systems, 2005.*, pages 122–127.

Dil, B., Dulman, S., and Havinga, P. (2006). Range-based localization in mobile sensor networks. In *European Workshop on Wireless Sensor Networks*, pages 164–179. Springer.

Eren, T., Goldenberg, O. K., Whiteley, W., Yang, Y. R., Morse, A. S., Anderson, B. D. O., and Belhumeur, P. N. (2004). Rigidity, computation, and randomization in network localization. In *IEEE INFOCOM 2004*, volume 4, pages 2673–2684 vol.4.

Gustafsson, F. (2010). *Statistical Sensor Fusion*. Studentlitteratur.

Kourogi, M., Sakata, N., Okuma, T., and Kurata, T. (2006). Indoor/outdoor pedestrian navigation with an embedded GPS/RFID/self-contained sensor system. In *Advances in Artificial Reality and Tele-Existence*, pages 1310–1321. Springer.

La, H. M. and Sheng, W. (2013). Distributed Sensor Fusion for Scalar Field Mapping Using Mobile Sensor Networks. *IEEE Transactions on Cybernetics*, 43(2):766–778.

Luo, W., Zhao, X., and Kim, T.-K. (2014). Multiple object tracking: A review. *arXiv preprint arXiv:1409.7618*, 1.

Mao, G., Drake, S., and Anderson, B. D. O. (2007). Design of an Extended Kalman Filter for UAV Localization. In *2007 Information, Decision and Control*, pages 224–229.

Martinelli, A., Pont, F., and Siegwart, R. (2005). Multi-Robot Localization Using Relative Observations. In *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, pages 2797–2802.

Moore, D., Leonard, J., Rus, D., and Teller, S. (2004). Robust Distributed Network Localization with Noisy Range Measurements. In *Proceedings of the 2Nd International Conference on Embedded Networked Sensor Systems*, SenSys '04, pages 50–61, Baltimore, MD, USA. ACM.

Nocedal, J. (2006). *Numerical Optimization*. Springer.

Rubenstein, M., Ahler, C., and Nagpal, R. (2012). Kilobot: A low cost scalable robot system for collective behaviors. In *2012 IEEE International Conference on Robotics and Automation*, pages 3293–3298.

Rubenstein, M. and Shen, W. M. (2009). Scalable self-assembly and self-repair in a collective of robots. In *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1484–1489.

Sahin, E. (2004). Swarm robotics: From sources of inspiration to domains of application. In *International Workshop on Swarm Robotics*, pages 10–20. Springer.

Shang, Y., Rumi, W., Zhang, Y., and Fromherz, M. (2004). Localization from connectivity in sensor networks. *IEEE Transactions on Parallel and Distributed Systems*, 15(11):961–974.

Shang, Y. and Ruml, W. (2004). Improved MDS-based localization. In *IEEE INFOCOM 2004*, volume 4, pages 2640–2651 vol.4.

Spears, W. M., Spears, D. F., Hamann, J. C., and Heil, R. (2004). Distributed, physics-based control of swarms of vehicles. *Autonomous Robots*, 17(2-3):137–162.

Tan, Y. (2015). *Handbook of Research on Design, Control, and Modeling of Swarm Robotics*. IGI Global, Hershey, PA, USA, 1st edition.

Wang, H. and Guo, Y. (2008). A decentralized control for mobile sensor network effective coverage. In *2008 7th World Congress on Intelligent Control and Automation*, pages 473–478. IEEE.

Xie, G. and Wang, L. (2007). Consensus control for a class of networks of dynamic agents. *International Journal of Robust and Nonlinear Control: IFAC-Affiliated Journal*, 17(10-11):941–959.

Zhou, C., Xu, T., and Dong, H. (2015). Distributed locating algorithm MDS-MAP (LF) based on low-frequency signal. *Computer Science and Information Systems*, 12(4):1289–1305.