Cooperative Distributed Source Seeking by Multiple Robots: Algorithms and Experiments

Shuai Li, Ruofan Kong, and Yi Guo, *Senior Member, IEEE*

***Abstract*—We consider the problem of source seeking using a group of mobile robots equipped with sensors for source concentration measurement. In the formulation, the robot team cooperatively estimates the gradient of the source field, moves to the source by tracing the gradient-ascending direction, and keeps a predefined formation in movement. We present two control algorithms with all-to-all and limited communications, respectively. For the case of all-to-all communication, rigorous analytic analysis proves that the formation center of the robots converges to the source in the presence of estimation errors with a bounded error, the upper bound of which is explicitly given. In the case of limited communication where centralized quantities are not available, distributed consensus filters are used to distributively estimate the centralized quantities, and then embedded in the distributed control laws. Numerical simulations are given to validate the effectiveness of the proposed approaches. Experimental results on the E-puck robot platform demonstrate satisfactory performances in a light source seeking application.**

***Index Terms*—Consensus filter, distributed control, multirobots, source seeking.**

I. Introduction

WE STUDY the source seeking problem in this paper: a source forms a scalar value field in space. We design algorithms to drive a group of robots, which can only sample local values, to the source. Potential applications include source localization of oil spill [1], scalar field mapping [2], cooperative foraging [3], chemical plume tracing [4], multirobot radio source localization [5], [6], and cooperative path planning [7]. For a robot with the ability of measuring concentration gradients, a simple moving strategy by following gradient-ascending direction can complete the task. However, in practice, most robots are only equipped with sensors for concentration measurement instead of the gradient. Similar problems to source seeking can be found in nature. For example, a male moth is able to approach a female one from far away by tracing the pheromone plume [8]. For a swarm of bacteria, without the ability of concentration gradient measurements, they are able to find the source of beneficial chemicals [9]. The phenomena give insight into the problem and inspire many studies in this field.

Different methods have been proposed to solve the problem, which include behavior-based source seeking [10]–[12] and control-based source seeking [13]–[18]. The behavior-based source seeking method often defines a set of elementary behaviors and a set of behavior combination rules. Different combinations of the elementary behaviors are activated in schedule to steer a single robot or a group of robots. The authors in [10] and [11] design behavior-based source seeking algorithms for a single robot. In [12], the method is extended to scenarios with a group of robots. However, the inherent dynamics of the robot are often ignored for behavior-based source seeking, and there is no guarantee to reach the source eventually. In contrast, the control-based source seeking method directly takes the robot dynamics into account to develop a control algorithm. Zhang *et al.* [13] use extremum-seeking control theories to design a source seeking control. Ogren *et al.* [14] solve the problem by decoupling it into formation maintenance and leader following. In [15], two strategies for source seeking are considered. The first strategy uses one robot to perform the task with historical data for estimating the gradient along the trajectory, while the other one uses a group of robots with projected gradient estimation. In [16], a stochastic gradient-ascent algorithm is proposed to drive a single robot to the maximum of a scalar field. In [17], the authors give a control law by combining a potential field control law and a gradient-based control law. In [18], the control law is obtained by separately considering the motion along the baseline and the motion perpendicular to the baseline. Among existing literature works using control-based method for multiple robot source seeking, most of them ignore the gradient estimation error to ease the treatment, and the control algorithm often requires shared information with all robots in the group. These limitations inspire our study to design control algorithms that are robust to the gradient estimation error and also relax the communication requirements.

In this paper, we use distributed robots for source seeking. The problem is modeled as a cooperative estimation and control problem. We first propose a source seeking algorithm with all-to-all communications. Theoretical analysis are given to prove that this algorithm enables the group of robots to approach the source. Then, a modified algorithm, which is fully distributed with limited communications among robots, is presented to solve the same source seeking problem. Both the all-to-all and the limited communication algorithms are validated by simulations and real robot experiments. The algorithm part of this paper was presented in our early conference paper [19] without robot experiments.

Manuscript received March 8, 2013; revised September 7, 2013 and November 24, 2013; accepted December 4, 2013. Date of publication January 2, 2014; date of current version June 13, 2014. Recommended by Technical Editor D. Sun. This work was supported in part by the National Science Foundation under Grant EFRI-1024660, Grant IIS-1218155, and Grant CNS-1318748.

The authors are with the Department of Electrical and Computer Engineering, Stevens Institute of Technology, Hoboken, NJ 07030 USA (e-mail: lshuai@stevens.edu; rkong1@stevens.edu; yguo1@stevens.edu).

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

Digital Object Identifier 10.1109/TMECH.2013.2295036

The contribution of this paper is twofold. First, we use gradient estimation to guide the robot movement, and provide a theoretical upper bound on the tracking error. Second, among existing methods using a group of robots, such as [14], [15], and [17], each robot needs global information from all the other robots. In contrast, the algorithm with limited communications presented in this paper is fully distributed and scalable, i.e., each robot only needs to communicate with its one-hop neighbors and no across-hop message passing is required.

The remainder of the paper is organized as follows. Section II gives the assumptions and the problem formulation. Section III presents the first control algorithm with all-to-all communications. The second algorithm requiring only limited communications is proposed in Section IV. In Section V, MATLAB simulations are performed to show the effectiveness of the proposed algorithms and their comparison with existing ones. In Section VI, experiments are conducted on a multirobot test bed. Section VII concludes this paper.

Notations and symbols used in this paper can be found in Appendix A, and useful preliminaries on graph theory for the theoretical analysis are in Appendix B.

II. Problem Formulation

Same as in [13]–[15], we assume that the robot's motion is described by a double integrator

$$\begin{aligned}\dot{\mathbf{x}}_i &= \mathbf{v}_i \\ \dot{\mathbf{v}}_i &= \mathbf{u}_i \quad \text{for } i = 1, 2, \ldots, n \end{aligned} \tag{1}$$

where $\mathbf{x}_i \in \mathbb{R}^k$, $\mathbf{v}_i \in \mathbb{R}^k$ and $\mathbf{u}_i \in \mathbb{R}^k$ are the position, the velocity, and the control input (acceleration) of the ith robot in a k-dimensional workspace.

Remark 1: The double integrator model is a simplification of the real model of a physical robot. A physical robot can be modeled as a Lagrangian system. Using feedback linearization to compensate the nonlinearity, a Lagrangian system can be converted to a double integrator model [22]. For this reason, we start our analysis with the double integrator dynamics of robots.

We make the following assumptions on the source and environments:

Assumption 1: The scalar-valued distribution $p(\mathbf{x}) : \mathbb{R}^k \to \mathbb{R}$ is a deterministic, time-invariant, concave function with respect to $\mathbf{x}$ and reaches its maximum at $\mathbf{x} = \mathbf{x}_s$.

Assumption 2: The Hessian matrix of $p(\mathbf{x})$ satisfies: $-\xi_2 \leq \lambda_{\max}(\mathbf{H})$ and $\lambda_{\max}(\mathbf{H}) \leq -\xi_1$ for all $\mathbf{x}$ in the domain with $\xi_2 > \xi_1 > 0$.

Remark 2: $p(\mathbf{x})$ models the deterministic spatial distribution of a scalar value field. For fields demonstrating certain stochastic property, the value of which fluctuates at a fixed position, it is sometimes possible to be converted into a deterministic field by averaging or filtering over time. In this paper, we focus on considering deterministic scalar field. For the field $p(\mathbf{x})$ formed by a single source, such as a temperature field or an electric field, it has the maximum value at the source position $\mathbf{x}_s$ and reduces with the increase of distance from it. Assumption 1 is a simplification to this observation. It is also a commonly made assumption in optimization (equivalently, $-p(\mathbf{x})$ is convex). In geometry, the greatest and the least eigenvalues of $\mathbf{H}(\mathbf{x})$ measure the greatest and the least curvatures of $p(\mathbf{x})$, respectively [23]. In practice, the absolute values of both the greatest and least curvature are bounded, by which we conclude that $\mathbf{H}(\mathbf{x})$ is both upper and lower bounded in the eigenvalue sense. Assumption 2 states this fact.

We distinguish the formation graph and communication graph in the paper:

Formation Graph: The graph with robots as nodes and the links between robots on the virtual structure of the desired formation as edges. We use $\mathbb{F}(i)$, which is a set of robots on the formation graph with edges connected with the ith robot, to represent the neighbor set of the ith robot on the formation graph.

Communication Graph: The graph with robots as nodes and the communication links as edges. We use $\mathbb{N}(i)$, which is a set of robots on the communication graph with edges connected with the ith robot, to represent the neighbor set of the ith robot on the communication graph.

We define two different communication topologies: All-to-all and limited communications, which are described in the following two assumptions, respectively.

Assumption 3 (All-to-all communication): The communication topology is an *all-to-all connected graph* with the robots as nodes and the one-hop communication link between every pair of robots as edges. Under this communication topology, the information available for the control design of the ith robot includes: the robot position $\mathbf{x}_j$, the robot velocity $\mathbf{v}_j$, and the sampled value of source concentration $p(\mathbf{x}_j(t))$, with $j = 1, 2, \ldots, n$.

Assumption 4 (Limited communication): The communication topology is a *connected undirected graph* with the robots as nodes and the communication link between one-hop neighboring robots as edges. Under this limited communication topology, the information available for the control design of the ith robot includes: the robot's own position $\mathbf{x}_i$ and velocity $\mathbf{v}_i$, its neighboring robots' position $\mathbf{x}_j$ and velocity $\mathbf{v}_j$, and the sampled concentration value of its own $p(\mathbf{x}_i(t))$ and neighbors $p(\mathbf{x}_j(t))$, with $j \in \mathbb{N}(i)$ denoting the neighbor set of the ith robot in the communication graph.

For the all-to-all communication case, the communication graph has an all-to-all connection, which allows exchange of information between any pair of neighboring robots on the formation graph. For the limited communication case, the selection of the formation graph is not unique for a given desired formation. We can always choose the formation graph the same as the communication graph under the initially connected configuration. We have the following assumption on this point.

Assumption 5: In the limited communication case, we assume the communication graph is initially connected. Given a desired formation, we assume that the formation graph can be chosen the same as the communication graph under the initial configuration. That is, $\mathbb{F}(i) = \mathbb{N}(i)$ for all possible i.

Remark 3: As we assume the initial communication graph is connected, our control goal includes maintaining connectivity under the initial configuration. We do not consider obstacles in the environment which may make it impossible to maintain

initial connectivity configuration. For connectivity maintenance in the presence of obstacles, readers can refer to techniques introduced in [24].

In this paper, we consider the source seeking problem with all-to-all and limited communication topologies. We define our control problem as follows.

Problem 1 (Source seeking with all-to-all communications): Under Assumptions 1, 2, and 3, in a k-dimensional workspace with a scalar valued distribution $p(\mathbf{x})$, design an algorithm to drive the center of a group of robots to the source $\mathbf{x}_s = \text{argmax}(p(\mathbf{x}))$ and simultaneously drive all robots to the desired formation.

Problem 2 (Source seeking with limited communications): Under Assumptions 1, 2, and 4, in a k-dimensional workspace with a scalar-valued distribution $p(\mathbf{x})$, design an algorithm to drive the center of a group of robots to the source $\mathbf{x}_s = \text{argmax}(p(\mathbf{x}))$ and simultaneously drive all robots to a desired formation.

Note that the differences of the two problems defined previously is the assumption on communication topology made in Assumptions 3 and 4.

III. Source Seeking With All-to-All Communications

In this section, we present the algorithm for source seeking with all-to-all communications based on the cooperative gradient estimation.

A. Cooperative Estimation of Gradients

Since the robots are equipped with sensors for concentration measurement (instead of gradient), each robot needs to make estimation of gradient and then follows the gradient-ascending direction to the source. In this section, we use a least square (LS) estimator for gradient estimation.

Generally, the measurement of $p(\mathbf{x})$ are different for robots located at different positions. Our goal is to estimate the gradient at $\mathbf{x}_c(t)$, which is the center of the formation, i.e., $\mathbf{x}_c(t) = \frac{1}{n}\sum_{i=1}^{n}\mathbf{x}_i(t)$, based on the sampling of $p(\mathbf{x})$ acquired by each robot. We use the following linear parameterization model to adaptively approximate the distribution of $p(\mathbf{x})$:

$$\hat{\mathbf{y}}(t) = [\,\mathbf{X}(t) \quad \mathbf{1}\,]\,\boldsymbol{\theta}(t) \tag{2}$$

with

$$\mathbf{X}(t) = \begin{bmatrix} \mathbf{x}_1^T(t) \\ \mathbf{x}_2^T(t) \\ \cdots \\ \mathbf{x}_n^T(t) \end{bmatrix} \tag{3}$$

and the measurement

$$\mathbf{y}(t) = \begin{bmatrix} p(\mathbf{x}_1(t)) \\ p(\mathbf{x}_2(t)) \\ \cdots \\ p(\mathbf{x}_n(t)) \end{bmatrix} \tag{4}$$

where $\mathbf{x}_1(t), \mathbf{x}_2(t), \ldots, \mathbf{x}_n(t)$ are all $k \times 1$ vectors with k denoting the space dimension, $\boldsymbol{\theta}(t)$ is the estimation parameter, which is a $(k+1) \times 1$ vector, $\hat{p}_i(\mathbf{x}(t))$ is the estimation of $p(\mathbf{x}(t))$ by the ith robot at time t, and $\mathbf{1}$ is an $n \times 1$ vector with all entries equal to 1. The estimation error is defined to be the difference between the estimation $\hat{\mathbf{y}}(t)$ and the measurement $\mathbf{y}(t) = [\,p(\mathbf{x}_1(t)), \quad p(\mathbf{x}_2(t)), \ldots, \quad p(\mathbf{x}_n(t))\,]^T = [\,p_1(t), \quad p_2(t), \ldots, \quad p_n(t)\,]^T$. The LS estimator minimizes the norm of the estimation error, i.e., to make $\hat{\mathbf{y}}(t) \approx \mathbf{y}(t)$. Solving $\hat{\mathbf{y}}(t) \approx \mathbf{y}(t)$ yields

$$\begin{aligned} \boldsymbol{\theta}(t) &= [\,\mathbf{X}(t) \quad \mathbf{1}\,]^{+}\,\mathbf{y}(t) \\ &= \begin{bmatrix} \mathbf{X}^T(t)\mathbf{X}(t) & \mathbf{X}^T(t)\mathbf{1} \\ \mathbf{1}^T\mathbf{X}(t) & n \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{X}^T(t) \\ \mathbf{1}^T \end{bmatrix} \mathbf{y}(t) \end{aligned} \tag{5}$$

where $[\,\mathbf{X}(t) \quad \mathbf{1}\,]^{+}$ is the pseudoinverse of $[\,\mathbf{X}(t) \quad \mathbf{1}\,]$. Therefore, we obtain the following parameter estimation:

$$\begin{aligned} \boldsymbol{\theta}(t) &= \begin{bmatrix} \mathbf{X}^T(t)\mathbf{X}(t) & \mathbf{X}^T(t)\mathbf{1} \\ \mathbf{1}^T\mathbf{X}(t) & n \end{bmatrix}^{-1} \begin{bmatrix} c\mathbf{X}^T(t) \\ \mathbf{1}^T \end{bmatrix} \mathbf{y}(t) \\ \hat{\mathbf{g}}_c(t) &= [\,\mathbf{I} \quad \mathbf{0}\,]\,\boldsymbol{\theta}(t) \end{aligned} \tag{6}$$

where $\hat{\mathbf{g}}_c(t)$ is the gradient estimation at the formation center $\mathbf{x}_c(t)$ at time t, $\mathbf{I}$ is a $k \times k$ identity matrix, $\mathbf{1}$ is a k-row vector with all entries equal to 1, and $\mathbf{0}$ is a k-row vector with all entries equal to 0. This equation gives us an optimal estimation of $\hat{\mathbf{g}}_c(t)$ in the sense of LSs.

B. Control Law Design

To solve the cooperative source seeking problem, we need to design two behaviors for the robot: one is the gradient climbing behavior, which drives the robot to the source and the other one is the formation achieving behavior, which guides robots to the desired formation. However, the goal is not realizable with the two behaviors only. In addition, we introduce a velocity damping behavior to avoid oscillation or overshooting around the source and an estimation error compensation behavior to reduce the effect of the gradient estimation error. We present the following control input to the ith robot:

$$\begin{aligned} \mathbf{u}_i = &- \sum_{j \in \mathbb{F}(i)} \omega_{1ij}(\mathbf{x}_i - \mathbf{x}_j - \mathbf{x}_{di} + \mathbf{x}_{dj}) + c_0\hat{\mathbf{g}}_c \\ &- \sum_{j \in \mathbb{F}(i)} \omega_{2ij}(\mathbf{v}_i - \mathbf{v}_j) - \frac{c_1}{n}\sum_{i=1}^{n}\mathbf{v}_i - c_2\,\text{sgn}\left(\sum_{i=1}^{n}\mathbf{v}_i\right) \end{aligned} \tag{7}$$

where $\mathbb{F}(i)$ denotes the neighbor set of the ith robot on the formation graph; n denotes the number of robots in the group; $\omega_{1ij} = \omega_{1ji}$ and $\omega_{2ij} = \omega_{2ji}$, which are positive constants; c_0, c_1, and c_2 are also positive constants; $\text{sgn}(\cdot)$ is the sign function, which equals to 1, -1, and 0 for a positive input, negative input, and the input of 0, respectively; $\mathbf{x}_i$ and $\mathbf{v}_i$ are the ith robot's position and velocity, respectively; $\mathbf{x}_{di}$ is the desired position of the ith robot in the virtual structure of the desired formation; $\hat{\mathbf{g}}_c$ is the gradient estimation given by (6).

In (7), the first and the third terms drive the robot to the desired formation, the second term generates the gradient climbing movement, the last two terms are velocity damping terms, which dissipates the kinematic energy of the robot.

Algorithm 1 Source seeking control for the ith robot with all-to-all communication

Require:

Concentration measurement p, position $\mathbf{x}$, velocity $\mathbf{v}$, desired formation $\mathbf{x}_d$, of all robots are available to the ith robot, a pre-set error constant ϵ.

Ensure:

To achieve the desired formation and drive the formation center to the source.

1: **repeat**
2: $p_i, \mathbf{x}_i, \mathbf{v}_i \Leftarrow$ Sensor readings.
3: $p_1, p_2, ..., p_{i-1}, p_{i+1}, ..., p_n; \mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_{i-1}, \mathbf{x}_{i+1}, ..., \mathbf{x}_n$ $\mathbf{v}_1, \mathbf{v}_2, ..., \mathbf{v}_{i-1}, \mathbf{v}_{i+1}, ..., \mathbf{v}_n$ $\Leftarrow$ Communication with all the other robots.
4: Calculate position matrix $\mathbf{X}$ in (3).
5: Calculate gradient estimation $\hat{\mathbf{g}}_c(t)$ in (6).
6: Calculate control input $\mathbf{u}_i$ (7).
7: **until** $(\|\hat{\mathbf{g}}_c(t)\| < \epsilon)$

Remark 4: In the control law (7), the term $\sum_{j\in\mathbb{F}(i)} \omega_{1ij}(\mathbf{x}_i - \mathbf{x}_j - \mathbf{x}_{di} + \mathbf{x}_{dj})$ and the term $\sum_{j\in\mathbb{F}(i)} \omega_{2ij}(\mathbf{v}_i - \mathbf{v}_j)$ are defined in the neighborhood $\mathbb{F}(i)$ of the formation graph while the rest terms are defined on the all-to-all topology and require information from all robots.

Combining the control input of all robots in the group, the control algorithm can be written into a compact form:

$$\mathbf{u} = -(\mathbf{L}_1 \otimes \mathbf{I})(\mathbf{x} - \mathbf{x}_d) - (\mathbf{L}_2 \otimes \mathbf{I})\mathbf{v} + c_0\mathbf{1} \otimes \hat{\mathbf{g}}_c - \frac{c_1}{n}\mathbf{1} \otimes ((\mathbf{1}^T \otimes \mathbf{I})\mathbf{v}) - \frac{c_2}{n}\mathbf{1} \otimes \mathrm{sgn}((\mathbf{1}^T \otimes \mathbf{I})\mathbf{v}) \quad (8)$$

where $\mathbf{u} = [\mathbf{u}_1^T, \mathbf{u}_2^T, \ldots, \mathbf{u}_n^T]^T$ is the control input, $\mathbf{x} = [\mathbf{x}_1^T, \mathbf{x}_2^T, \ldots, \mathbf{x}_n^T]^T$ is the position vector of all robots, $\mathbf{v} = [\mathbf{v}_1^T, \mathbf{v}_2^T, \ldots, \mathbf{v}_n^T]^T$ is the velocity vector of all robots; both $\mathbf{L}_1$ and $\mathbf{L}_2$ are symmetric Laplacian matrices on the formation graph, the i,jth entry of $\mathbf{L}_1$ is $-\omega_{1ij}$ for $i \neq j$ and $i \in \mathbb{F}(j)$, 0 for $i \neq j$ and $i \notin \mathbb{F}(j)$, and $\sum_{l\in\mathbb{F}(i)} \omega_{1il}$ for $i = j$, the i,jth entry of $\mathbf{L}_2$ is $-\omega_{2ij}$ for $i \neq j$ and $i \notin \mathbb{F}(j)$, 0 for $i \neq j$ and $i \in \mathbb{F}(j)$, and $\sum_{l\in\mathbb{F}(i)} \omega_{2il}$ for $i = j$; $\mathbf{I}$ is a $k \times k$ identity matrix with k denoting the dimension of the space, n is the number of robots, $\mathbf{x}_d$ is a constant vector with $n \times k$ rows, which represents the desired formation, $\mathbf{1}$ is an $n \times 1$ vector with all entries equal to 1, and $\otimes$ is the Kronecker product.

The procedures of the proposed control algorithm with all-to-all communications is stated in Algorithm 1.

In Algorithm 1, the ith robot first collects its measurement of concentration p_i, position $\mathbf{x}_i$, and velocity $\mathbf{v}_i$ (Line 2), and collects concentration, position, and velocity of all the other robots by communication (Line 3). After this, the position matrix $\mathbf{X}$ is constructed according to (3) (Line 4). Then, gradient estimation of $\hat{\mathbf{g}}_c(t)$ is made according to (6) (Line 5). Subsequently, (7) is used to calculate the control input $\mathbf{u}_i$. Lines 2, 3, 4, 5, and 6 are repeated in sequence until the norm of the estimated gradient $\hat{\mathbf{g}}_c(t)$ is less than a predefined positive constant ϵ, i.e., $\|\hat{\mathbf{g}}_c(t)\| < \epsilon$.

To validate the effectiveness of Algorithm 1 in theory, we make the following assumption.

Assumption 6: The gradient estimation $\hat{\mathbf{g}}_c(t)$ obtained by (6) has a bounded error, i.e., $\|\hat{\mathbf{g}}_c(t) - \mathbf{g}_c(t)\| \leq e_0$ (e_0 is a positive constant), in which $\mathbf{g}_c(t)$ denotes the true value of the gradient at the formation center.

We have the following theorem to state the convergence of the designed control algorithm.

Theorem 1: Under Assumptions 1, 2, 3, and 6, Algorithm 1 with the control law (7), where the parameter $c_2 > c_0\sqrt{k}e_0$ (k denotes the dimension of space), solves Problem 1. The formation center $\mathbf{x}_c(t) = \frac{1}{n}\sum_{i=1}^n \mathbf{x}_i(t)$ converges to $\mathbf{x}^*$, which satisfies $\hat{\mathbf{g}}_c(\mathbf{x}^*) = \mathbf{0}$. Moreover, $\mathbf{x}^*$ has a bounded distance away from the source $\mathbf{x}_s$

$$\|\mathbf{x}^* - \mathbf{x}_s\| \leq \frac{2e_0}{\xi_1} \quad (9)$$

where ξ_1 and e_0 are defined in Assumptions 2 and 6, respectively.

Proof: See Appendix C. ■

IV. Distributed Source Seeking With Limited Communications

In the previous section, we developed a control algorithm for cooperative source seeking, which requires all-to-all communications. Aiming at reducing communication burdens, in this section, we develop a fully distributed control algorithm, which only requires limited neighbor-to-neighbor communications.

From (7), it is clear that $\hat{\mathbf{g}}_c$ and $\frac{1}{n}(\mathbf{1}^T \otimes \mathbf{I})\mathbf{v}$ require information from every robots in the team. We use consensus filters to estimate them in a distributed manner [20], [21], [25]. With consensus filters, a robot is able to estimate the average of filter inputs by running the following protocol on every robot:

$$\dot{z}_i = \sum_{j\in\mathbb{N}(i)} a_{ij}(z_j - z_i) + \gamma(\tau_i - z_i) \quad (10)$$

where $\mathbb{N}(i)$ denotes the neighbor set of the ith robot on the communication graph defined in Assumption 4; z_i is a scalar state maintained by the ith robot; a_{ij} is a positive constant for $j \in \mathbb{N}(i)$ and satisfies $a_{ij} = a_{ji}$; γ is a positive constant, and τ_i is the scalar input to the ith robot. By running (10) on every robot, z_i is able to track the average of inputs, i.e., $\frac{1}{n}\sum_j^n \tau_j$.

To estimate $\hat{\mathbf{g}}_c$ in a distributed manner, we first rewrite the expression of $\boldsymbol{\theta}(t)$ in (6) into average forms (without confusions, the time t aside the time-varying variables are omitted)

$$\boldsymbol{\theta} = \begin{bmatrix} \frac{1}{n}\mathbf{X}^T\mathbf{X} & \frac{1}{n}\mathbf{X}^T\mathbf{1} \\ \frac{1}{n}\mathbf{1}^T\mathbf{X} & 1 \end{bmatrix}^{-1} \begin{bmatrix} \frac{1}{n}\mathbf{X}^T\mathbf{y} \\ \frac{1}{n}\mathbf{1}^T\mathbf{y} \end{bmatrix} = \begin{bmatrix} \frac{1}{n}\sum_{i=1}^n \mathbf{x}_i\mathbf{x}_i^T & \frac{1}{n}\sum_{i=1}^n \mathbf{x}_i \\ \frac{1}{n}\sum_{i=1}^n \mathbf{x}_i^T & 1 \end{bmatrix}^{-1} \begin{bmatrix} \frac{1}{n}\sum_{i=1}^n \mathbf{x}_i p_i \\ \frac{1}{n}\sum_{i=1}^n p_i \end{bmatrix}. \quad (11)$$

We can online estimate $\frac{1}{n}\sum_{i=1}^{n}\mathbf{x}_i\mathbf{x}_i^T$, $\frac{1}{n}\sum_{i=1}^{n}\mathbf{x}_i$, $\frac{1}{n}\sum_{i=1}^{n}\mathbf{x}_i p_i$, and $\frac{1}{n}\sum_{i=1}^{n}p_i$ distributively by running four separate consensus filters on every robot. We have the following filter expressions for $\mathbf{Z}_{1i}$, $\mathbf{z}_{2i}$, $\mathbf{z}_{3i}$, z_{4i}, $\boldsymbol{\theta}_{ei}$, and $\hat{\mathbf{g}}_{cei}$, which are estimations of $\frac{1}{n}\sum_{i=1}^{n}\mathbf{x}_i\mathbf{x}_i^T$, $\frac{1}{n}\sum_{i=1}^{n}\mathbf{x}_i$, $\frac{1}{n}\sum_{i=1}^{n}\mathbf{x}_i p_i$, $\frac{1}{n}\sum_{i=1}^{n}p_i$, $\boldsymbol{\theta}$, and $\hat{\mathbf{g}}_c$ by the ith robot, respectively (Note that $\mathbf{Z}_{1i}$ is a $k\times k$ matrix, $\mathbf{z}_{2i}$ and $\mathbf{z}_{3i}$ are both $k\times 1$ vectors, and z_{4i} is a scalar.)

$$\begin{aligned}
\dot{\mathbf{Z}}_{1i} &= \sum\nolimits_{j\in\mathbb{N}(i)} a_{ij}(\mathbf{Z}_{1j}-\mathbf{Z}_{1i}) + \gamma(\mathbf{x}_i\mathbf{x}_i^T - \mathbf{Z}_{1i})\\
\dot{\mathbf{z}}_{2i} &= \sum\nolimits_{j\in\mathbb{N}(i)} a_{ij}(\mathbf{z}_{2j}-\mathbf{z}_{2i}) + \gamma(\mathbf{x}_i - \mathbf{z}_{2i})\\
\dot{\mathbf{z}}_{3i} &= \sum\nolimits_{j\in\mathbb{N}(i)} a_{ij}(\mathbf{z}_{3j}-\mathbf{z}_{3i}) + \gamma(\mathbf{x}_i p_i - \mathbf{z}_{3i})\\
\dot{z}_{4i} &= \sum\nolimits_{j\in\mathbb{N}(i)} a_{ij}(z_{4j}-z_{4i}) + \gamma(p_i - z_{4i}) \qquad (12)\\
\boldsymbol{\theta}_{ei} &= \begin{bmatrix}\mathbf{Z}_{1i} & \mathbf{z}_{2i}\\ \mathbf{z}_{2i}^T & 1\end{bmatrix}^{-1}\begin{bmatrix}\mathbf{z}_{3i}\\ z_{4i}\end{bmatrix}\\
\hat{\mathbf{g}}_{cei} &= [\,\mathbf{I}\quad \mathbf{0}\,]\,\boldsymbol{\theta}_{ei}. \qquad (13)
\end{aligned}$$

To estimate $\frac{1}{n}(\mathbf{1}^T\otimes\mathbf{I})\mathbf{v}$, we first express it into the form that $\frac{1}{n}(\mathbf{1}^T\otimes\mathbf{I})\mathbf{v} = \frac{1}{n}\sum_{i=1}^{n}\mathbf{v}_i$. Denoting $\mathbf{z}_{5i}$ the estimation of $\frac{1}{n}(\mathbf{1}^T\otimes\mathbf{I})\mathbf{v}$ by the ith robot, we have

$$\dot{\mathbf{z}}_{5i} = \sum_{j\in\mathbb{N}(i)} a_{ij}(\mathbf{z}_{5j}-\mathbf{z}_{5i}) + \gamma(\mathbf{v}_i - \mathbf{z}_{5i}). \qquad (14)$$

We can now replace $\hat{\mathbf{g}}_c$ and $\frac{1}{n}(\mathbf{1}^T\otimes\mathbf{I})\mathbf{v}$ in (7) with $\hat{\mathbf{g}}_{cei}$ in (13) and z_{5i} in (14) using the distributed filter scheme shown previously. Note that we need the neighboring robots' positions and velocities in the first and the third terms of (7). As $x_{di}, i = 1,\ldots,n$, are known to all robots, we can choose the formation graph the same as the communication graph, i.e., $\mathbb{F}(i) = \mathbb{N}(i)$ for all possible i. Therefore, the distributed control law for the limited communication case is

$$\begin{aligned}
\mathbf{u}_i = &-\sum_{j\in\mathbb{N}(i)} \omega_{1ij}(\mathbf{x}_i - \mathbf{x}_j - \mathbf{x}_{di} + \mathbf{x}_{dj}) + c_0\hat{\mathbf{g}}_{cei}\\
&-\sum_{j\in\mathbb{N}(i)} \omega_{2ij}(\mathbf{v}_i - \mathbf{v}_j) - c_1\mathbf{z}_{5i} - c_2\,\mathrm{sgn}(\mathbf{z}_{5i}). \qquad (15)
\end{aligned}$$

The procedures of the proposed control algorithm with limited communications is stated in Algorithm 2.

The difference of Algorithms 1 and 2 lies in that consensus filters are used to estimate $\hat{\mathbf{g}}_{cei}$ in a distritbuted way in Algorithm 2 (Lines 5, 6, and 7).

Corollary 1: Under Assumptions 1, 2, 4, 5, and 6, Algorithm 2 with the control law (15) and the estimators (12), (13), and (14) running on a fast enough time scale solves Problem 2.

Remark 5: In this section, we present a distributed control algorithm for cooperative source seeking. The core idea is to use distributed consensus filters running on a fast enough time scale to estimate the centralized quantities in Algorithm 1 and use the estimated values to replace the actual centralized quantity in the control. Theoretical analysis of the convergence of consensus filter plus distributed controllers can be found in [27].

Algorithm 2 Distributed source seeking control for the ith robot with limited communications

Require:

Concentration measurement p, position $\mathbf{x}$, velocity $\mathbf{v}$ and the desired formation $\mathbf{x}_d$ of itself and its neighbors are available to the ith robot, a pre-set error constant ϵ.

Ensure:

To achieve the desired formation and drive the formation center to the source.

1: Initializing state variables $\mathbf{Z}_{1i}$, $\mathbf{z}_{2i}$, $\mathbf{z}_{3i}$, $\mathbf{z}_{4i}$, $\mathbf{z}_{5i}$.
2: **repeat**
3: $\mathbf{x}_i, \mathbf{v}_i, \mathbf{p}_i \Leftarrow$ Sensor readings.
4: $\mathbf{x}_j$, $\mathbf{v}_j \Leftarrow$ Communication with neighbor $j\in\mathbb{N}(i)$.
5: Collect jth robot's consensus filter output $\mathbf{Z}_{1j}$, $\mathbf{z}_{2j}$, $\mathbf{z}_{3j}$, $\mathbf{z}_{4j}$, $\mathbf{z}_{5j}$ using neighbor-to-neighbor communications.
6: Calculate state update of consensus filter $\mathbf{Z}_{1i}$, $\mathbf{z}_{2i}$, $\mathbf{z}_{3i}$, $\mathbf{z}_{4i}$, $\mathbf{z}_{5i}$ using equation (12) and equation (14).
7: Calculate distributed gradient estimation $\theta_{ei}, \hat{\mathbf{g}}_{cei}$ using (13).
8: Calculate distributed control input $\mathbf{u}_i$ according to (15).
9: **until** $(\|\hat{\mathbf{g}}_{cei}(t)\| < \epsilon)$

Remark 6: For an n robot network with an all-to-all communication, each robot communicates with all other $(n-1)$ ones, which results in a communication complexity of $O(n^2)$. With the increase of the number of robot n, the communication burden increases significantly. In contrast, each robot only communicates with a few neighboring robots in the limited communication case. For instance, for a communication graph with a two-dimensional (2-D) lattice topology, each robot only communicates with four neighbors even when the total number of robots n increases, thus the communication complexity is $O(n)$ in this system. Therefore, the proposed distributed control in limited communication is scalable.

V. Simulation Comparisons

In this section, we compare our algorithms with the methods proposed in [15] and the method proposed in [13]. There are two methods proposed in [15]: one uses a single robot to perform the task and the other one uses a group of robots. We call the two strategies PGS and PGM for short, respectively, and we call the strategy proposed in [13] ES method for short.

MATLAB simulations are performed under a representative set of parameters. For the ES method, parameters are chosen to be the same as in [13]. For the PGS and PGM method, parameter setup cannot be found in the associated paper [15]. We choose $k_d = 5$, $k_s = 1$, $d_0 = 1$, and $\kappa = 1$ (see that paper for definitions of each parameter) and the potential function is chosen to be the one suggested in the paper. The desired formation is shown in Fig. 1(a). For simplicity, we choose the formation and communication graph to be the same for the all-to-all and the limited communication cases, as shown in Fig. 1(b). We choose $c_0 = 20, c_1 = 7, c_2 = 7, \gamma = 1, L_1 = 6L_0$, $L_2 = L_0$, and $L_3 = 30L_0$ [where L_3 is the Laplacian matrix

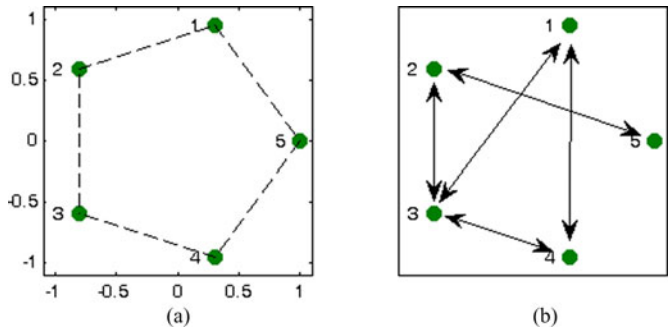


Fig. 1. Desired formation and the communication topologies used in the simulations. (a) Desired formation. (b) Communication graph.

associated with the weight matrix $[a_{ij}]$ in (12)] with

$$L_0 = \begin{bmatrix} 2 & 0 & -1 & -1 & 0 \\ 0 & 2 & -1 & 0 & -1 \\ -1 & -1 & 3 & -1 & 0 \\ -1 & 0 & -1 & 2 & 0 \\ 0 & -1 & 0 & 0 & 1 \end{bmatrix}.$$

For the proposed methods and the PGM method, five robots are employed. Robots start from different positions. For PGS and ES methods, which use a single robot, the initial position is set to be the center of robots in the multiple robot case. For simplicity, we choose $p(\mathbf{x}) = 100 - \|\mathbf{x}\|$ [13], [15].

We first compare the trajectories of the formation center (for methods using a single robot, we use its trajectory for evaluation). From Fig. 2(a), we can see that Algorithm 1, Algorithm 2, the PGM method, and EM method, are able to steer the formation center to the vicinity of the source. Among them, ES method uses a single robot, which uses fewer robots in number than Algorithm 1, Algorithm 2, and the PGM method. However, the robot takes a spiral like trajectory to the source and the travelled distance is much longer than other methods. Algorithm 1 outperforms Algorithm 2, PGM method, and EM method according to the traveled distance. Compared to Algorithm 1, the trajectory by using Algorithm 2 is a little longer, resulting from the dynamic interaction of the consensus filter and the robot dynamics in Algorithm 2. However, different from all the other methods simulated here, as stated previously, Algorithm 2 is a fully distributed algorithm, which only requires information exchanges between one-hop neighbors. Nevertheless, our Algorithm 2 still outperforms PGS, PGM, and ES methods in the sense that Algorithm 2 has a shorter trajectory than them.

We then compare the proposed Algorithm 1, Algorithm 2, and the PGM method, which use a group of robots, to see whether the desired formation are reached. From Fig. 2(b), we can see that PGM method does not reach a uniform distribution on a circle, while both Algorithms 1 and 2 reach the desired formation. The robot velocity profiles of our proposed methods are shown in Figs. 3 and 4. As observed, the velocity of robots converges to a common value, which eventually reaches zero. This indicates that robots reach a common velocity in order to reach a fixed formation, while this common velocity converges to zero when the source is reached.

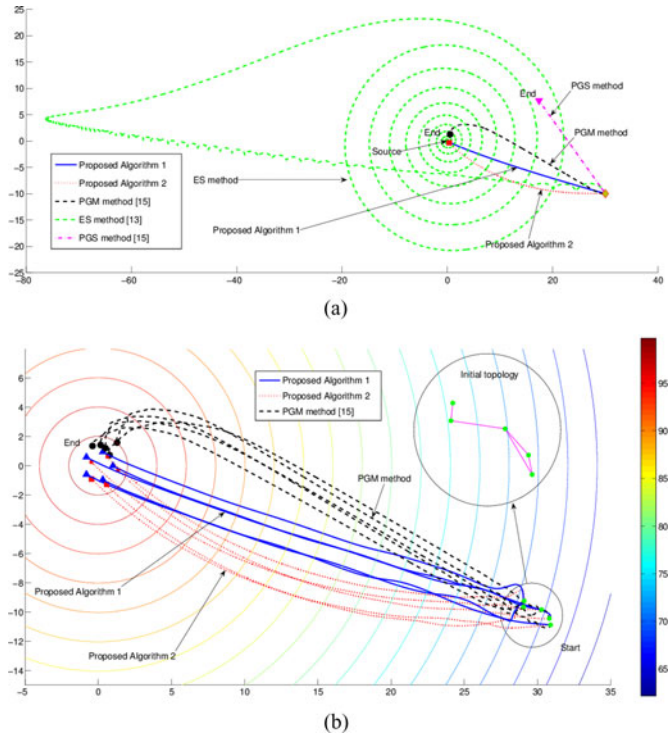


Fig. 2. Simulation comparisons between PGS method [15], ES method [13], PGM method [15], our Algorithm 1, and our Algorithm 2. The color bar denotes the concentration value of the source. (a) Trajectory of formation centers (for the single robot case, it is the robot's own trajectory). The yellow diamond represents the start position. The red square is the end position of the robot by ES method, the proposed Algorithm 1 and the proposed Algorithm 2. It is also the source position. The black dot and the pink triangle are the end positions of the robot by PGM method and that of the robot by PGS method. (b) Trajectories of each robot for methods using multiple robots. The black dots, the blue triangles, and the red squares are the end positions of robots by PGM method, Algorithm 1, and Algorithm 2, respectively. The green dots represents the start positions. The contour of the field $p(\mathbf{x})$ is plotted in the figure.

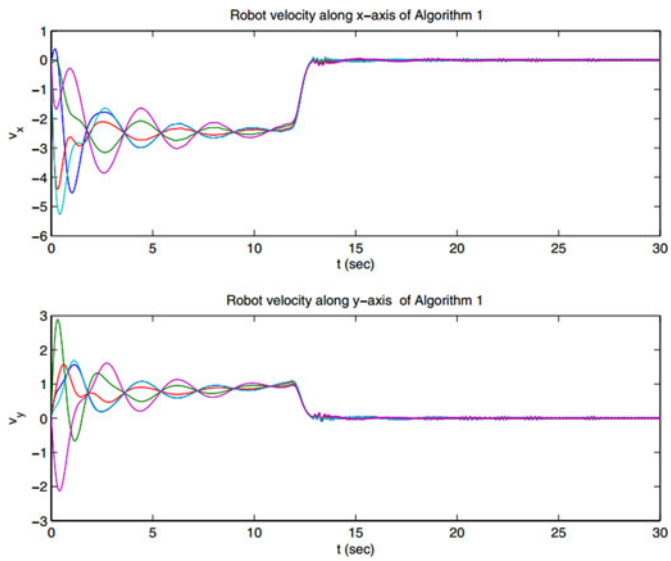


Fig. 3. Robot velocity profiles of the proposed Algorithm 1.

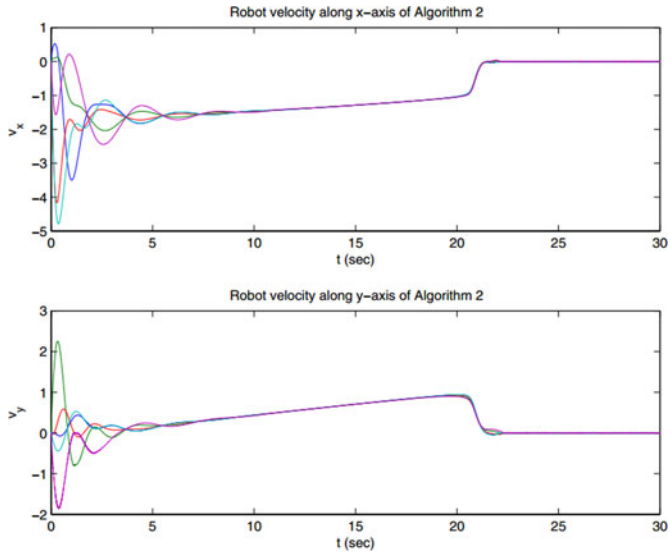


Fig. 4. Robot velocity profiles of the proposed Algorithm 2.

VI. Experimental Validation

In this section, the proposed algorithms are tested by robot experiments using E-puck robots in a light source seeking application. The experimental implementation of the algorithms and the experimental results are described next.

A. Robot

In the experiment, the E-puck mobile robot is used to serve as the robot platform. It is a differential-driven-wheeled mobile robot equipped with two step motors for movement, Bluetooth module for communication, and an on-board microprocessor for programming and control. Navigation sensors including infrared (IR) sensors, a VGA camera, and a 3-D accelerometer are equipped on the robot. Encoders are equipped to record the relative pose (position and bearing). The kinematic model of the ith E-puck robot is

$$\begin{bmatrix} \dot{x}_i \\ \dot{y}_i \\ \dot{\alpha}_i \end{bmatrix} = \begin{bmatrix} \frac{\cos\alpha_i}{2} & \frac{\cos\alpha_i}{2} \\ \frac{\sin\alpha_i}{2} & \frac{\sin\alpha_i}{2} \\ -\frac{1}{l} & \frac{1}{l} \end{bmatrix} \begin{bmatrix} \Gamma_{i1} \\ \Gamma_{i2} \end{bmatrix} \tag{16}$$

where (x_i, y_i) represents the Cartesian coordinates of the middle point of the driving wheel axle; α_i is the bearing of the robot body with respect to the x-axis, l is the length between the two driving wheels; Γ_{i1} and Γ_{i2} are the speeds of the left and the right wheels, respectively (see Fig. 5).

Using feedback linearization, we are able to transform the robot model (16) into new coordinates [26],

$$\begin{bmatrix} \dot{x}_i' \\ \dot{y}_i' \end{bmatrix} = \begin{bmatrix} \tau_{i1} \\ \tau_{i2} \end{bmatrix} \tag{17}$$

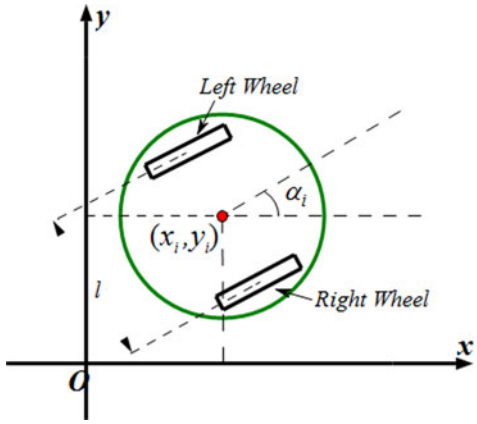


Fig. 5. E-puck model.

where τ_{i1} and τ_{i2} are the new inputs, and

$$\begin{bmatrix} x_i' \\ y_i' \end{bmatrix} = \begin{bmatrix} x_i + c\cos\alpha_i \\ y_i + c\sin\alpha_i \end{bmatrix} \tag{18}$$

with c a positive constant. The relationship between the wheel velocity input Γ_i and new input τ_i is expressed as

$$\begin{bmatrix} \Gamma_{i1} \\ \Gamma_{i2} \end{bmatrix} = M(\alpha_i) \begin{bmatrix} \tau_{i1} \\ \tau_{i2} \end{bmatrix} \tag{19}$$

with

$$M(\alpha_i) = \begin{bmatrix} \frac{l}{2c}\sin\alpha_i + \cos\alpha_i & -\frac{l}{2c}\cos\alpha_i + \sin\alpha_i \\ -\frac{l}{2c}\sin\alpha_i + \cos\alpha_i & +\frac{l}{2c}\cos\alpha_i + \sin\alpha_i \end{bmatrix}. \tag{20}$$

For the E-puck robot, the length between wheels is $l = 0.053$ m. Also, we set $c = 0.0265$ m for our experimental validation.

B. Experiment Setup

Light Source: A standard 75-W incandescent light bulb is placed on a big table locating at $(0.59, -1.36)$ in the global coordinates (x-axis is pointing up, and y-axis is pointing left, with the origin at the center of the workspace).

Sensors: The E-puck robot has eight IR sensors, which are placed around the top of it, and are used to measure the light intensity. Closer to the light source, the sensor gets higher light intensity measurement values. To balance the light intensity measured by sensors from all directions, each E-puck robot utilizes the averaged value of its eight IR sensors to measure the local light intensity.

Localization System: The positions and bearings of the E-puck robots are obtained by odometry based on the information recorded by encoders with their initial positions and bearings calibrated in the global coordinate.

Robot Communication: The communication type among robots is set at "radio mode" (channel 1) with a communication range covering all peers. In the all-to-all communication case, the information from all other robots are used for state update, while the communication packages from the nonneighboring robots are intentionally dropped in the limited communication

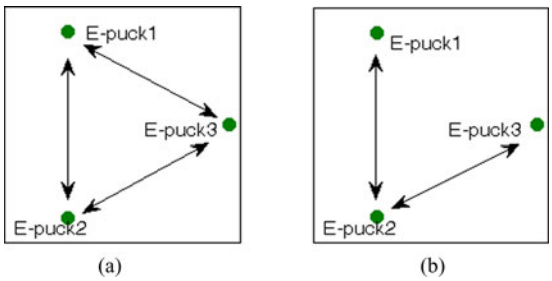


Fig. 6. E-puck communication topologies considered in the experiment. (a) All-to-all communication. (b) Limited communication.

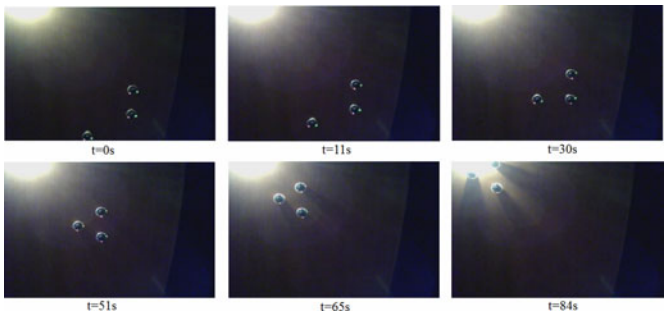


Fig. 7. Light source seeking experiment with the all-to-all communication on three E-puck robots.

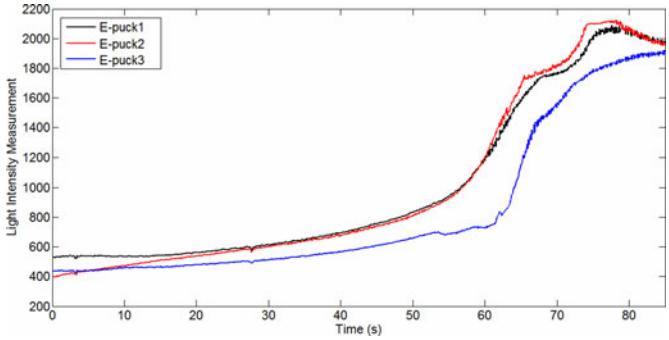


Fig. 8. Robots' light intensity measurements in the all-to-all communication case.

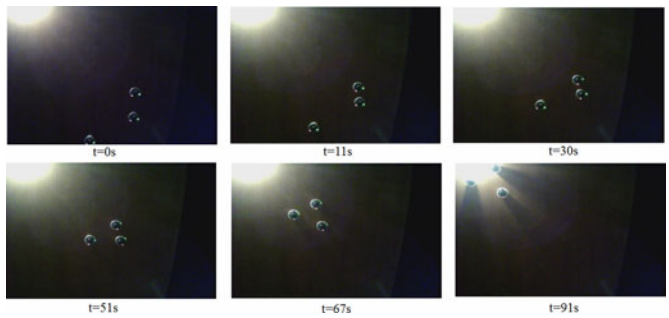


Fig. 9. Light source seeking experiment with the limited communication on three E-puck robots.

case to form the limited communication topology. The supervisory mode is implemented on E-pucks in the experiment.

Three robots, E-puck 1, 2, and 3, are deployed in the light field. The initial relative positions of E-puck robots are randomly chosen as $(0, -2)$, $(-0.4, -1.7)$, and $(-0.2, -2)$ with bearings 0, 0, and 0, respectively, in the global coordinates. The virtual structure of the desired formation is set as $(x_{d1}, y_{d1}) = (0, 0.2)$, $(x_{d2}, y_{d2}) = (0.1, 0.1732)$, and $(x_{d3}, y_{d3}) = (0, 0)$, which corresponds to an equilateral triangle with edge length 0.2 m. The control coefficient c_0 is chosen as $c_0 = 10$. According to the communication topologies shown in Fig. 6, the Laplacian matrices in the all-to-all communication case and the limited communication case are chosen, respectively, as follows:

$$\begin{bmatrix} 12 & -6 & -6 \\ -6 & 12 & -6 \\ -6 & -6 & 12 \end{bmatrix} \quad \begin{bmatrix} 6 & -6 & 0 \\ -6 & 12 & -6 \\ 0 & -6 & 6 \end{bmatrix}.$$

C. Experimental Results

The experiment results are reported for both the all-to-all communication case and the limited communication case. The performance in these two scenarios are then compared.

1) All-to-All Communication: Algorithm 1 is implemented for the all-to-all communication case. The communication topology of three robots is shown in Fig. 6(a). Fig. 7 displays the experimental results. The initial positions of E-puck1, E-puck2, and E-puck3 are randomly chosen, and the robots move toward the light source. At time $t = 11$ s, the robots are still in the process of forming the equilateral triangle during their movement to the light source. The snapshot at time $t = 30$ s reveals that this process continues. At time $t = 51$ s, the equilateral triangle formation is almost reached. Then, they keep the formation, and gradually approach to the light source, shown in Fig. 7 at $t = 65$ s, until the formation center gets to the light source at $t = 84$ s. Fig. 8 shows that the light intensity measurement of each robot increases as the robots approach to the light source, where noises and measurement uncertainties exist.

2) Limited Communication: The limited communication algorithm is tested under the same setup as in the case of all-to-all communication except the communication topology is as shown in Fig. 6(b). This implementation follows the procedure of Algorithm 2. Fig. 9 shows the experimental results with limited communication. The three robots start from a dark place with a low light intensity value at $t = 0$ s. The robots move toward the desired formation and simultaneously move toward the light source at $t = 11$ s and $t = 30$ s. At time $t = 51$ s, the robots are still away from the desired formation. At time $t = 67$ s, the equilateral formation is almost formed. They maintain the formation until they finally reach the light source at $t = 91$ s.

3) Performance Comparison: Fig. 10 shows odometry trajectories in both the all-to-all and the limited communication cases. The solid-line trajectory shows the robot center trajectory in the all-to-all communication, and the dashed-line trajectory shows the robot center trajectory in the limited communication. Comparing the all-to-all and limited communication cases, it can be seen that the robot source seeking in the limited communication case takes longer time to converge to the source due to the use of consensus filter. Also, the path length in the limited communication case is slightly longer.

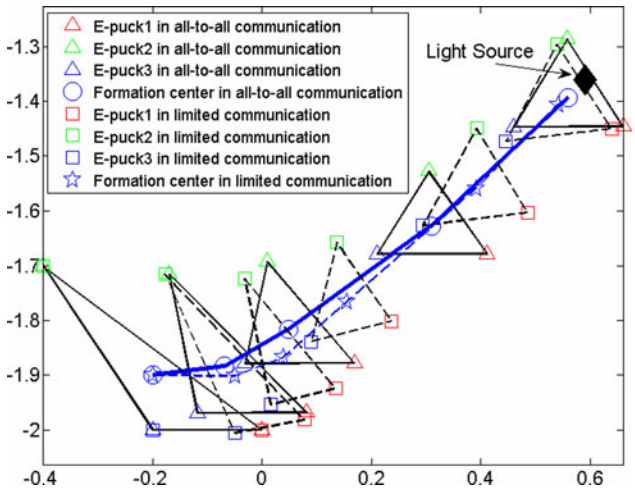


Fig. 10. Trajectories of the multirobot light source seeking on the E-puck platform with both the all-to-all and the limited communications. The robot positions and formation center are drawn in every 21 s (except that the last two dashed triangles are drawn in 7 s to show the end positions of the robots in limited communication).

VII. Conclusion

Cooperative distributed source seeking was studied in this paper. Two control algorithms were proposed to solve the problem with all-to-all and limited communications, respectively. In the case of all-to-all communication, theoretical analysis proved that the proposed algorithm guarantees convergence to the source with all robots reaching the desired formation. With the constraint of limited communication, consensus filters were used to distributively estimate centralized quantities, and then embedded in the distributed control laws. MATLAB simulations in an ideal environmental setup demonstrated the effectiveness of the proposed algorithms. Experiments on the E-puck robot platform were performed to show the applicability of the proposed algorithms in a light source seeking scenario.

Appendix A
Notations and Symbols

$\lambda_{\max}(\mathbf{A})$ and $\lambda_{\min}(\mathbf{A})$ represent the greatest and the smallest eigenvalues of a symmetric matrix A. $\|\mathbf{x}\|$ and $\|\mathbf{x}\|_1$ are the Euclidean norm and the 1-norm of a vector $\mathbf{x}$, respectively, defined as follows:

$$\|\mathbf{z}\| = \sqrt{z_1^2 + z_2^2 + \cdots + z_n^2} \tag{21}$$

$$\|\mathbf{z}\|_1 = |z_1| + |z_2| + \cdots + |z_n| \tag{22}$$

for $\mathbf{z} = [z_1, z_2, \ldots, z_n]^T$ $\otimes$ denotes the Kronecker product. We use $\mathbf{1}$ to denote a vector with all 1 entries, $p(\mathbf{x})$ to denote the concentration at $\mathbf{x}$ and $\text{argmax}(p(\mathbf{x}))$ to denote the optimal point where $p(\mathbf{x})$ reaches the maximum. The Hessian matrix of $p(\mathbf{x})$, denoted as $\mathbf{H}(\mathbf{x})$, is an $n \times n$ square matrix with the i, jth entry is defined as $\frac{\partial^2 p}{\partial x_i \partial x_j}$ for $\mathbf{x} = [x_1, x_2, \ldots, x_n]^T$.

Appendix B
Preliminaries on Graph Theory

An undirected graph $G(V, E, \mathbf{A})$ is denoted by $(V, E, \mathbf{A})$, where V is the set of nodes, E is the set of edges with $E \subseteq V \times V$, and $\mathbf{A} = [a_{ij}]$ with $a_{ij} = a_{ji} > 0$ for $j \in \mathbb{N}_i$ ($\mathbb{N}(i)$ denotes the neighbor set of node i) and $a_{ij} = a_{ji} = 0$ for $j \notin \mathbb{N}_i$, is the weighted adjacency matrix. The degree of a node in the graph is defined as $\deg(v_i) = \sum_{j=1}^{n} a_{ji}$. An undirected graph G is called connected if there always exists a sequence of consecutive edges starting from a given node i to another given node j, where nodes i and j could be any node in the graph only if $i \neq j$. The degree matrix $\mathbf{\Delta} = [\Delta_{ij}]$ is a diagonal matrix with $\Delta_{ij} = 0$ for all $i \neq j$ and $\Delta_{ii} = \deg_{\text{out}}(v_i)$ for all i. The Laplacian matrix $\mathbf{L}$ of the graph G is defined as $\mathbf{L} = \mathbf{\Delta} - \mathbf{A}$. The Laplacian matrix always has a zero eigenvalue associated with the eigenvector $\mathbf{1}$. A Laplacian matrix defined on a connected undirected graph is semidefinite positive and has rank $n - 1$, where n denotes the dimension of the Laplacian matrix. All nonzero eigenvalues of such a Laplacian matrix are positive.

Appendix C
Proof of Theorem 1

There are three steps in the proof: Step 1-derivation of the formation center's dynamics; Step 2-stability analysis using Lyapunov Theory; Step 3-derivation of the source seeking error.

Step 1: Derivation of the Formation Center's Dynamics

Substituting the control input (8) into the robot dynamics (1) yields

$$\dot{\mathbf{v}} = -(\mathbf{L}_1 \otimes \mathbf{I})(\mathbf{x} - \mathbf{x}_d) - (\mathbf{L}_2 \otimes \mathbf{I})\mathbf{v} + c_0 \mathbf{1} \otimes \hat{\mathbf{g}}_c - \frac{c_1}{n}\mathbf{1} \otimes ((\mathbf{1}^T \otimes \mathbf{I})\mathbf{v}) - c_2 \mathbf{1} \otimes \text{sgn}((\mathbf{1}^T \otimes \mathbf{I})\mathbf{v}). \tag{23}$$

As to the formation center $\mathbf{x}_c$, we have

$$\mathbf{x}_c = \frac{1}{n}\sum_{i=1}^{n} \mathbf{x}_i = \frac{1}{n}(\mathbf{1}^T \otimes \mathbf{I})\mathbf{x}. \tag{24}$$

Recalling the property of Laplacian matrices that $\mathbf{L}_1 \mathbf{1} = \mathbf{0}$, the fact that $\mathbf{L}_1$ and $\mathbf{L}_2$ are symmetric, and the mixed-product property of Kronecker products, we have $(\mathbf{1}^T \otimes \mathbf{I})(\mathbf{L}_1 \otimes \mathbf{I}) = \mathbf{0}$, $(\mathbf{1}^T \otimes \mathbf{I})(\mathbf{L}_2 \otimes \mathbf{I}) = \mathbf{0}$, $(\mathbf{1}^T \otimes \mathbf{I})(\mathbf{1} \otimes \hat{\mathbf{g}}_c) = n\hat{\mathbf{g}}_c$, $(\mathbf{1}^T \otimes \mathbf{I})(\mathbf{1} \otimes ((\mathbf{1}^T \otimes \mathbf{I})\dot{\mathbf{x}})) = n^2 \dot{\mathbf{x}}_c$, and $(\mathbf{1}^T \otimes \mathbf{I})(\mathbf{1} \otimes \text{sgn}((\mathbf{1}^T \otimes \mathbf{I})\dot{\mathbf{x}})) = n\text{sgn}(\dot{\mathbf{x}}_c)$. Based on these results, left multiplicating the matrix $\frac{1}{n}(\mathbf{1}^T \otimes \mathbf{I})$ on both sides of (23) yields the following dynamics of $\mathbf{x}_c$:

$$\ddot{\mathbf{x}}_c = c_0 \hat{\mathbf{g}}_c - c_1 \dot{\mathbf{x}}_c - c_2 \text{sgn}(\dot{\mathbf{x}}_c). \tag{25}$$

Defining $\mathbf{v}_c = \dot{\mathbf{x}}_c$, we have

$$\dot{\mathbf{v}}_c = c_0 \hat{\mathbf{g}}_c - c_1 \mathbf{v}_c - c_2 \text{sgn}(\mathbf{v}_c). \tag{26}$$

Step 2: Stability Analysis Using the Lyapunov Stability Theory

We choose the following Lyapunov function:

$$V = (\mathbf{x} - \mathbf{x}_d - \mathbf{1} \otimes \mathbf{x}_c)^T (\mathbf{L}_1 \otimes \mathbf{I})(\mathbf{x} - \mathbf{x}_d - \mathbf{1} \otimes \mathbf{x}_c) + p(\mathbf{x}_s) - p(\mathbf{x}_c) + \frac{1}{2c_0}\mathbf{v}_c^T \mathbf{v}_c + (\mathbf{v} - \mathbf{1} \otimes \mathbf{v}_c)^T (\mathbf{v} - \mathbf{1} \otimes \mathbf{v}_c).$$

Note that in the expression of V, $\mathbf{L}_1 \otimes \mathbf{I}$ is symmetric since $(\mathbf{L}_1 \otimes \mathbf{I})^T = \mathbf{L}_1^T \otimes \mathbf{I}^T = \mathbf{L}_1 \otimes \mathbf{I}$, and $(\mathbf{x} - \mathbf{x}_d - \mathbf{1} \otimes \mathbf{x}_d)^T (\mathbf{L}_1 \otimes \mathbf{I})(\mathbf{x} - \mathbf{x}_d - \mathbf{1} \otimes \mathbf{x}_c)$ is semipositive definite since the Laplacian matrix $\mathbf{L}_1$ is semipositive definite [21]. In addition, $p(\mathbf{x}_s) - p(\mathbf{x}_c) \geq 0$ due to Assumption 1. Therefore, V is indeed semipositive definite. Calculating the time derivative of V along the trajectory of (23) and (26) yields

$$\dot{V} = 2(\mathbf{x} - \mathbf{x}_d)^T (\mathbf{L}_1 \otimes \mathbf{I})\mathbf{v} - \mathbf{g}_c^T \mathbf{v}_c + \frac{1}{c_0} \dot{\mathbf{v}}_c^T \mathbf{v}_c + 2(\dot{\mathbf{v}} - \mathbf{1} \otimes \dot{\mathbf{v}}_c)^T (\mathbf{v} - \mathbf{1} \otimes \mathbf{v}_c). \quad (27)$$

From (23) and (26), we obtain

$$\dot{\mathbf{v}} - \mathbf{1} \otimes \dot{\mathbf{v}}_c = -(\mathbf{L}_1 \otimes \mathbf{I})(\mathbf{x} - \mathbf{x}_d) - (\mathbf{L}_2 \otimes \mathbf{I})\mathbf{v} - \frac{c_1}{n} \mathbf{1} \otimes ((\mathbf{1}^T \otimes \mathbf{I})\mathbf{v}) - \frac{c_2}{n} \mathbf{1} \otimes \operatorname{sgn}((\mathbf{1}^T \otimes \mathbf{I})\mathbf{v}) + c_1 \mathbf{1} \otimes \mathbf{v}_c + c_2 \mathbf{1} \otimes \operatorname{sgn}(\mathbf{v}_c). \quad (28)$$

Along with (24), we have

$$\dot{\mathbf{v}} - \mathbf{1} \otimes \dot{\mathbf{v}}_c = -(\mathbf{L}_1 \otimes \mathbf{I})(\mathbf{x} - \mathbf{x}_d) - (\mathbf{L}_2 \otimes \mathbf{I})\mathbf{v}. \quad (29)$$

Substituting (25) and (29) into (27) yields $\dot{V} = -2\mathbf{v}^T (\mathbf{L}_2 \otimes \mathbf{I})\mathbf{v} + (\hat{\mathbf{g}}_c - \mathbf{g}_c)^T \mathbf{v}_c - \frac{c_1}{c_0} \mathbf{v}_c^T \mathbf{v}_c - \frac{c_2}{c_0} \|\mathbf{v}_c\|_1$, where $\|\cdot\|_1$ denotes the 1-norm of a vector. Using properties of vector norms, we further obtain

$$\dot{V} \leq -2\mathbf{v}^T (\mathbf{L}_2 \otimes \mathbf{I})\mathbf{v} + \|\hat{\mathbf{g}}_c - \mathbf{g}_c\|_1 \|\mathbf{v}_c\|_1 - \frac{c_1}{c_0} \mathbf{v}_c^T \mathbf{v}_c - \frac{c_2}{c_0} \|\mathbf{v}_c\|_1 \leq -2\mathbf{v}^T (\mathbf{L}_2 \otimes \mathbf{I})\mathbf{v} - \frac{c_1}{c_0} \mathbf{v}_c^T \mathbf{v}_c - \left(\frac{c_2}{c_0} - \sqrt{k} e_0 \right) \|\mathbf{v}_c\|_1 \leq 0. \quad (30)$$

Note that $\|\cdot\|$ denotes the Euclidean norm in (30). In the derivation of (30), the norm inequalities $(\hat{\mathbf{g}}_c - \mathbf{g}_c)^T \mathbf{v}_c \leq \|\hat{\mathbf{g}}_c - \mathbf{g}_c\|_1 \|\mathbf{v}_c\|_1$ and $\|\mathbf{v}_c\|_1 \leq \sqrt{k} \|\mathbf{v}_c\|$ are employed. The right side of (30) is seminegative definite by noting that $\mathbf{L}_2 \otimes \mathbf{I}$ is semipositive definite and $c_2 > c_0 \sqrt{k} e_0$. Together with the fact that the right side of (30) is a function of state variables and the invariance-like theorem (see [28, Th. 8.4]), we draw the conclusion that the right side of (30) goes to zeros as time elapses. Therefore,

$$\lim_{t \to \infty} \mathbf{v} = \mathbf{1} \otimes \boldsymbol{\alpha}_1 \quad \text{and} \quad \lim_{t \to \infty} \mathbf{v}_c = \mathbf{0} \quad (31)$$

where $\boldsymbol{\alpha}_1$ is a vector with k rows. With (25) and (29), we further conclude the following holds as time goes to infinity

$$\hat{\mathbf{g}}_c \to \mathbf{0} \quad (32)$$

$$-(\mathbf{L}_1 \otimes \mathbf{I})(\mathbf{x} - \mathbf{x}_d) \to \mathbf{1} \otimes \dot{\boldsymbol{\alpha}}_1. \quad (33)$$

For any $k \times 1$ vector $\boldsymbol{\alpha}_2$, we have $\dot{\boldsymbol{\alpha}}_1^T \boldsymbol{\alpha}_2 = 0$ by left multiplication of $\mathbf{1}^T \otimes \boldsymbol{\alpha}_2^T$ on both sides of (33). Due to the arbitrarity of the choice of $\boldsymbol{\alpha}_2$, we obtain $\dot{\boldsymbol{\alpha}}_1 = \mathbf{0}$. In other words, we conclude that $\boldsymbol{\alpha}_1$ is a constant vector. With (33), we obtain $\mathbf{x} - \mathbf{x}_d \to \mathbf{1} \otimes \boldsymbol{\alpha}_3$ as $t \to \infty$, where $\boldsymbol{\alpha}_3$ is a vector with k rows. This equation means that, when time elapses, the position $\mathbf{x}$ is a translation from $\mathbf{x}_d$ given by the desired formation. Since transformation does not change relative positions, we obtain the conclusion that $\mathbf{x}$ converges to the formation given by $\mathbf{x}_d$. Moreover, we get $\mathbf{x}_c \to \mathbf{x}^*$ by employing $\hat{\mathbf{g}}_c(\mathbf{x}^*) = \mathbf{0}$ and (32).

Step 3: Derivation of the Source Seeking Error

The Taylor expansion of $p(\mathbf{x})$ at $\mathbf{x}^*$ yields $p(\mathbf{x}) = p(\mathbf{x}^*) + \nabla^T p(\mathbf{x}^*)(\mathbf{x} - \mathbf{x}^*) + \frac{1}{2}(\mathbf{x} - \mathbf{x}^*)^T H(\mathbf{x}_1)(\mathbf{x} - \mathbf{x}^*)$, where $\mathbf{x}_1$ is between $\mathbf{x}$ and $\mathbf{x}^*$. For $\mathbf{x} = \mathbf{x}_s$, this equation yields $p(\mathbf{x}_s) = p(\mathbf{x}^*) + \nabla^T p(\mathbf{x}^*)(\mathbf{x}_s - \mathbf{x}^*) + \frac{1}{2}(\mathbf{x}_s - \mathbf{x}^*)^T H(\mathbf{x}_1)(\mathbf{x}_s - \mathbf{x}^*)$. Thus, we get

$$0 \leq p(\mathbf{x}_s) - p(\mathbf{x}^*) \leq \|\nabla p(\mathbf{x}^*)\| \|\mathbf{x}_s - \mathbf{x}^*\| - \frac{\xi_1}{2} \|\mathbf{x}_s - \mathbf{x}^*\|^2 = \|\nabla p(\mathbf{x}^*) - \hat{\mathbf{g}}_c(\mathbf{x}^*)\| \|\mathbf{x}_s - \mathbf{x}^*\| - \frac{\xi_1}{2} \|\mathbf{x}_s - \mathbf{x}^*\|^2 \leq e_0 \|\mathbf{x}_s - \mathbf{x}^*\| - \frac{\xi_1}{2} \|\mathbf{x}_s - \mathbf{x}^*\|^2. \quad (34)$$

Note that Assumption 6 is employed in the previous derivation of (34). Also note that the left side of (34) is not larger than 0 according to Assumption 1, so we have $0 \leq e_0 \|\mathbf{x}_s - \mathbf{x}^*\| - \frac{\xi_1}{2} \|\mathbf{x}_s - \mathbf{x}^*\|^2$ Therefore, $\|\mathbf{x}^* - \mathbf{x}_s\| \leq \frac{2e_0}{\xi_1}$. This concludes the proof.

REFERENCES

[1] H. Senga, N. Kato, A. Ito, H. Niou, M. Yoshie, I. Fujita, K. Igarashi, and E. Okuyama, "Development of spilled oil tracking autonomous buoy system," in *Proc. Oceans Conf.*, 2007, pp. 1–10.

[2] H. M. La and W. Sheng, "Distributed sensor fusion for scalar field mapping using mobile sensor networks," *IEEE Trans. Cybern.*, vol. 43, no. 2, pp. 766–778, Apr. 2013.

[3] K. Sugawara, T. Kazama, and T. Watanabe, "Foraging behavior of interacting robots with virtual pheromone," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2004, vol. 3, pp. 3074–3079.

[4] W. Li, J. Farrell, S. Pang, and R. Arrieta, "Moth-inspired chemical plume tracing on an autonomous underwater vehicle," *IEEE Trans. Robot.*, vol. 22, no. 2, pp. 292–307, Apr. 2006.

[5] D. Song, C. Y. Kim, and J. Yi, "Simultaneous localization of multiple unknown and transient radio sources using a mobile robot," *IEEE Trans. Robot.*, vol. 28, no. 3, pp. 668–680, Jun. 2012.

[6] H. Ishida, H. Tanaka, H. Taniguchi, and T. Moriizumi, "Localization of multiple unknown transient radio sources using multiple paired mobile robots with limited sensing ranges," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2011, vol. 1, pp. 5167–5172.

[7] S. Liu, D. Sun, and C. Zhu, "Coordinated motion planning for multiple mobile robots along designed paths with formation requirement," *IEEE/ASME Trans. Mechatronics*, vol. 16, no. 6, pp. 1021–1031, Dec. 2011.

[8] J. S. Kennedy, A. R. Ludlow, and C. J. Sanders, "Guidance of flying male moths by wind-borne sex pheromone," *Physiol. Entomol.*, vol. 6, no. 4, pp. 395–412, 2008.

[9] K. Passino, "Biomimicry of bacterial foraging for distributed optimization and control," *IEEE Control Syst. Mag.*, vol. 22, no. 3, pp. 52–67, Jun. 2002.

[10] J. Farrell, S. Pang, and W. Li, "Chemical plume tracing via an autonomous underwater vehicle," *IEEE J. Ocean. Eng.*, vol. 30, no. 2, pp. 428–442, Apr. 2005.

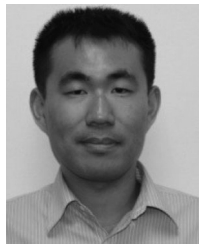
[11] H. Ishida, H. Tanaka, H. Taniguchi, and T. Moriizumi, "Mobile robot navigation using vision and olfaction to search for a gas/odor source," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2004, vol. 1, pp. 313–318.

[12] A. Hayes, A. Martinoli, and R. Goodman, "Distributed odor source localization," *IEEE Sens. J.*, vol. 2, no. 3, pp. 260–271, Jun. 2002.

[13] C. Zhang, A. Siranosian, and M. Krstic, "Extremum seeking for moderately unstable systems and for autonomous vehicle target tracking without position measurements," *Automatica*, vol. 43, no. 10, pp. 1832–1839, 2007.

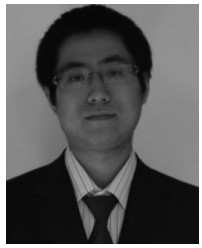
[14] P. Ogren, E. Fiorelli, N. E. Leonard, and S. Member, "Cooperative control of mobile sensor networks: Adaptive gradient climbing in a distributed environment," *IEEE Trans. Autom. Control*, vol. 49, no. 8, pp. 1292–1302, Aug. 2004.

[15] R. Bachmayer and N. Leonard, "Vehicle networks for gradient descent in a sampled environment," in *Proc. IEEE Conf. Decision Control*, 2002, vol. 1, pp. 112–117.
[16] N. Atanasov, J. Le Ny, N. Michael, and G. J. Pappas, "Stochastic source seeking in complex environments," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2012, pp. 3013–3018.
[17] V. Gazi and K. M. Passino, "Stability analysis of social foraging swarms," *IEEE Trans. Syst., Man, Cybern.*, vol. 34, no. 1, pp. 539–557, Feb. 2004.
[18] W. Wu, I. D. Couzin, and F. Zhang, "Bio-inspired source seeking with no explicit gradient estimation," in *Proc. IFAC Workshop Distrib. Estimation Control Netw. Syst.*, 2012, pp. 240–245.
[19] S. Li and Y. Guo, "Distributed source seeking by cooperative robots: All-to-all and limited communications," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2012, pp. 1107–1112.
[20] R. Olfati-saber and J. S. Shamma, "Consensus filters for sensor networks and distributed sensor fusion," in *Proc. IEEE Conf. Decision Control*, 2005, pp. 6699–6703.
[21] R. Freeman, P. Yang, and K. Lynch, "Stability and convergence properties of dynamic average consensus estimators," in *Proc. IEEE Conf. Decision Control*, 2006, pp. 338–343.
[22] M. W. Spong, S. Hutchinson, and M. Vidyasagar, *Robot Modeling and Control*, 1st ed. Hoboken, NJ, USA: Wiley, 2006.
[23] R. Lopez, *Differential Geometry of Curves and Surfaces in Lorentz-Minkowski Space*, Notes of Mini-Course of University of Sao Paulo, Brazil, eprint arXiv:0810.3351, 2008.
[24] X. Li, D. Sun, and J. Yang, "Bounded controller for multirobot navigation while maintaining network connectivity in the presence of obstacles," *Automatica*, vol. 49, pp. 285–292, 2013.
[25] P. Yang, R. Freeman, and K. Lynch, "Distributed cooperative active sensing using consensus filters," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2007, pp. 405–410.
[26] M. Hwang, M. Elwin, P. Yang, R. Freeman, and K. Lynch, "Experimental validation of multi-agent coordination by decentralized estimation control," in *Networking Humans, Robots, and Environments*, N. Y. Chong, Ed. Sharjah, U.A.E.: Bentham, 2011.
[27] P. Yang, R. Freeman, and K. Lynch, "Multi-agent coordination by decentralized estimation and control," *IEEE Trans. Autom. Control*, vol. 53, no. 11, pp. 2480–2496, Dec. 2008.
[28] H. K. Khalil, *Nonlinear Systems*, 3rd ed. Englewood Cliffs, NJ, USA: Prentice-Hall, 2001.



Shuai Li received the B.E. degree in precision mechanical engineering from Hefei University of Technology, Hefei, China, and the M.E. degree in automatic control engineering from the University of Science and Technology of China, Hefei, China. He is currently working toward the Ph.D. degree in electrical engineering at Stevens Institute of Technology, Hoboken, NJ, USA.

His research interests include distributed estimation and control and its applications in robotics.



Ruofan Kong received the B.E. degree in optoelectronic information engineering from Nanjing University of Science and Technology, Nanjing, China, and the M.E. degree in electrical engineering from Stevens Institute of Technology, Hoboken, NJ, USA.

His research interests include robotic networks and distributed control.

Yi Guo (SM'04) received the B.S. and M.S. degrees from Xi'an University of Technology, Xi'an, China, in 1992 and 1995, respectively, and the Ph.D. degree from the University of Sydney, Darlington, Australia, in 1999, all in electrical engineering.

She is an Associate Professor in the Department of Electrical and Computer Engineering, Stevens Institute of Technology, Hoboken, NJ, USA, which she joined in 2005. She was a Postdoctoral Research Fellow at Oak Ridge National Laboratory from 2000 to 2002, and a Visiting Assistant Professor at the University of Central Florida, Orlando, FL, USA, from 2002 to 2005. Her main research interests include autonomous mobile robotics, distributed sensor networks, and nonlinear control systems.

Dr. Guo is an Associate Editor of the *IEEE Robotics and Automation Magazine*.