2012 IEEE International Conference on Robotics and Automation
RiverCentre, Saint Paul, Minnesota, USA
May 14-18, 2012

Distributed Source Seeking by Cooperative Robots: All-to-All and Limited Communications*

Shuai Li and Yi Guo

Abstract— **We consider the problem of source seeking using a group of mobile robots equipped with sensors for concentration measurement (instead of the gradient). In our formulation, each robot maintains a gradient estimation, moves to the source by tracing the gradient, and all together keep a predefined formation in movement. We present two control algorithms with all-to-all and limited communications, respectively. The estimation error is taken into account to derive robust control algorithms. Comparing to existing methods, the proposed algorithm with limited communications is fully distributed. Both theoretical analysis and numerical simulations are given to validate the effectiveness of our methods.**

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], chemical plume tracing [2], cooperative foraging [3], etc. 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 [2]. For a swarm of bacteria, without the ability of concentration gradient measurements, they still are able to find the source of beneficial chemicals [4]. The phenomenons give insight into the problem and inspire many studies in this field.

Many different methods are proposed to solve problem including behavior-based source seeking [5]–[8] and control-based source seeking [9], [10], [13], [15]. 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. Authors in [5], [6] design behavior-based source seeking algorithms for a single robot. In [7], the method is extended to the 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. In [13], extremum-seeking control theories are used to design a source seeking control. [9] gives a control law by combining a potential field control law and a gradient control law. In [15], two strategies for source seeking are considered. The first strategy uses a single robot with the historical data for estimating the gradient along the trajectory while the other one use a group of robots with projected gradient estimation. Ogren et al. in [10] solve the problem by decoupling it into formation maintenance and leader following. Among existing literatures using control-based method, 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 modelled as a cooperative optimal estimation and control problem and we first derive an algorithm with all-to-all communications. Theoretical analysis are given to prove this algorithm enables the group of robots to approach the source. Then, based on the results, a modified algorithm, which is fully distributed with limited communications, is presented to solve the same problem.

The contribution of this paper is twofold. First, we use the gradient estimation to guide the movement, and we provide a theoretical upper bound on the tracking error. Second, among existing methods using a group of robots, such as [9], [10], [15], each robot needs global information from all the others. In contrast, the algorithm with limited communications presented in this paper is a fully distributed and scalable one, i.e., each robot only needs to communicate with one-hop neighbors and no across-hop message passing is required.

The rest of the paper is organized as follows. Section II gives assumptions and formulations. In Section III, the parameter estimation is given by using a least-square estimator. Section IV presents the first control algorithm with all-to-all communications. The second algorithm requiring only limited communications is proposed in Section V. In Section VI, simulations are performed to show the effectiveness.

Notations and symbols: $\lambda_{max}(\mathbf{A})$ and $\lambda_{min}(\mathbf{A})$ represent the greatest and the smallest eigenvalues of a square matrix A. We use $p(\mathbf{x})$ to denote the concentration at $\mathbf{x}$ and $\mathbf{H}(\mathbf{x})$ to denote the Hessian matrix of $p(\mathbf{x})$. argmax$(p(\mathbf{x}))$ denotes the optimal point where $p(\mathbf{x})$ reaches the maximum.

*The work was partially supported by the National Science Foundation under Grant EFRI-1024660.

S. Li and Y. Guo are with the Department of Electrical and Computer Engineering, Stevens Institute of Technology, Hoboken, NJ 07030, USA. Email addresses: lshuai@stevens.edu (S. Li) and yguo1@stevens.edu (Y. Guo).

II. Problem Formulation

Same as in [10], [13], [15], we assume that the robot's motion is described by a double integrator:

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

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.

We make the following assumptions on the environments:

Assumption 1: The scalar valued distribution $p(\mathbf{x}) : \mathbb{R}^k \to \mathbb{R}$ is a 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: $-\xi_2 \leq \lambda_{min}(\mathbf{H})$ and $\lambda_{max}(\mathbf{H}) \leq -\xi_1$ for all $\mathbf{x}$ in the domain with $\xi_2 > \xi_1 > 0$.

Remark 1: For the field $p(\mathbf{x})$ formed by a single source, such as a temperature field and 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 $H(\mathbf{x})$ measure the greatest and the least curvatures of $p(\mathbf{x})$, respectively [11]. In practice, the absolute values of both the greatest and the least curvature are bounded, by which we conclude that $H(\mathbf{x})$ is both upper and lower bounded in the eigenvalue sense. Assumption 2 states this fact.

We define the cooperative source seeking problem as below:

Problem 1**:** *Under Assumptions* 1 *and* 2*, 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 $\mathbf{x}_s = argmax(p(\mathbf{x}))$ and simultaneously drive all robots to a desired formation. The available information for the control algorithm of the ith robot is the sampled value $p(\mathbf{x}_i(t))$ at time t.*

III. 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 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 $p(\mathbf{x})$:

$$\hat{\mathbf{y}}(t) = \begin{bmatrix} \mathbf{X}(t) & \mathbf{1} \end{bmatrix} \theta(t) \tag{2}$$

with

$$\hat{\mathbf{y}}(t) = \begin{bmatrix} \hat{p}(\mathbf{x}_1(t)) \\ \hat{p}(\mathbf{x}_2(t)) \\ ... \\ \hat{p}(\mathbf{x}_n(t)) \end{bmatrix}, \ \mathbf{X}(t) = \begin{bmatrix} \mathbf{x}_1^T(t) \\ \mathbf{x}_2^T(t) \\ ... \\ \mathbf{x}_n^T(t) \end{bmatrix} \tag{3}$$

where $\mathbf{x}_1(t)$, $\mathbf{x}_2(t)$, ..., $\mathbf{x}_n(t)$ are all $k \times 1$ vectors with k denoting the space dimension, $\theta(t)$ 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 a $n \times 1$ 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 $\mathbf{y}(t) = \begin{bmatrix} p(\mathbf{x}_1(t)), & p(\mathbf{x}_2(t)), & ..., & p(\mathbf{x}_n(t)) \end{bmatrix}^T = \begin{bmatrix} p_1(t), & p_2(t), & ..., & p_n(t) \end{bmatrix}^T$. 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:

$$\begin{aligned} \theta(t) &= \begin{bmatrix} \mathbf{X}(t) & \mathbf{1} \end{bmatrix}^+ \mathbf{y}(t) \\ &= \begin{bmatrix} \mathbf{X}^T(t)\mathbf{X}(t) & \mathbf{X}^T(t)\mathbf{1} \\ \mathbf{1}^T\mathbf{X}(t) & n \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{X}^T(t) \\ \mathbf{1}^T \end{bmatrix} \mathbf{y}(t) \end{aligned} \tag{4}$$

where $\begin{bmatrix} \mathbf{X}(t) & \mathbf{1} \end{bmatrix}^+$ is the pseudoinverse of $\begin{bmatrix} \mathbf{X}(t) & \mathbf{1} \end{bmatrix}$. Therefore, we get the following parameter estimation:

$$\begin{aligned} \theta(t) &= \begin{bmatrix} \mathbf{X}^T(t)\mathbf{X}(t) & \mathbf{X}^T(t)\mathbf{1} \\ \mathbf{1}^T\mathbf{X}(t) & n \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{X}^T(t) \\ \mathbf{1}^T \end{bmatrix} \mathbf{y}(t) \\ \hat{\mathbf{g}}_c(t) &= \begin{bmatrix} \mathbf{I} & \mathbf{0} \end{bmatrix} \theta(t) \end{aligned} \tag{5}$$

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 least squares.

IV. 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 gradient estimation (5).

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 only the two behaviors. 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. Therefore, we present the following control input to the ith robot,

$$\begin{aligned} \mathbf{u}_i = &- \sum_{j \in \mathbb{N}(i)} \omega_{1ij}(\mathbf{x}_i - \mathbf{x}_j - \mathbf{x}_{di} + \mathbf{x}_{dj}) + c_0 \hat{\mathbf{g}}_c \\ &- \sum_{j \in \mathbb{N}(i)} \omega_{2ij}(\mathbf{v}_i - \mathbf{v}_j) - \frac{c_1}{n}\sum_{i=1}^{n} \mathbf{v}_i - c_2 \mathrm{sgn}(\sum_{i=1}^{n} \mathbf{v}_i) \end{aligned} \tag{6}$$

where $\mathbb{N}(i)$ denotes the neighbor set of the ith robot, 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, 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 relative position of the ith robot in the desired formation, $\hat{\mathbf{g}}_c$ is the gradient estimation given by (5). In (6), the first two terms drive the robot to the desired formation, the third term generates the gradient climbing movement, the fourth term is a velocity damping term, which dissipates the kinematic energy of the robot, and the last term is an extra damping term to compensate the inaccuracy of gradient estimation. Combining the control input of all robots in the group, the control algorithm can be written into a compact form:

$$\begin{aligned}\mathbf{u} \;=\;& -(\mathbf{L}_1 \otimes \mathbf{I})(\mathbf{x} - \mathbf{x}_d) - (\mathbf{L}_2 \otimes \mathbf{I})\mathbf{v} + c_0 \mathbf{1} \otimes \hat{\mathbf{g}}_c \\ & - \frac{c_1}{n} \mathbf{1} \otimes \big((\mathbf{1}^T \otimes \mathbf{I})\mathbf{v}\big) - \frac{c_2}{n} \mathbf{1} \otimes \mathrm{sgn}\big((\mathbf{1}^T \otimes \mathbf{I})\mathbf{v}\big) \end{aligned} \tag{7}$$

where $\mathbf{u} = [\mathbf{u}_1^T, \mathbf{u}_2^T, ..., \mathbf{u}_n^T]^T$ is the control input, $\mathbf{x} = [\mathbf{x}_1^T, \mathbf{x}_2^T, ..., \mathbf{x}_n^T]^T$ is the position vector of all robots, $\mathbf{v} = [\mathbf{v}_1^T, \mathbf{v}_2^T, ..., \mathbf{v}_n^T]^T$ is the velocity vector of all robots, both $\mathbf{L}_1$ and $\mathbf{L}_2$ are symmetric Laplacian matrices on the undirected graph constructed by the group of robots, the i-jth entry of $\mathbf{L}_1$ is $-\omega_{1ij}$ for $i \neq j$ and $\sum_{l=1}^{n} \omega_{1il}$ for $i = j$, the i-jth entry of $\mathbf{L}_2$ is $-\omega_{2ij}$ for $i \neq j$ and $\sum_{l=1}^{n} \omega_{2il}$ for $i = j$, $\mathbf{I}$ is an $k \times k$ 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 a $n \times 1$ 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 for clarity.

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

Require:
Concentration measurement p, position $\mathbf{x}$ and velocity $\mathbf{v}$ of all robots are available to the ith robot.

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: Position matrix $\mathbf{X} \Leftarrow$ Equation (3).
5: Gradient estimation $\hat{\mathbf{g}}_c(t) \Leftarrow$ Equation (5).
6: Control input $\mathbf{u}_i \Leftarrow$ Equation (6).
7: **until** $(\|\hat{\mathbf{g}}_c(t)\| < \epsilon)$

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 equation (3) (Line 4). Then, gradient estimation of $\hat{\mathbf{g}}_c(t)$ is made according to equation (5) (Line 5). Subsequently, equation (6) is used to calculate the control input $\mathbf{u}_i$. Line 2, 3, 4, 5, 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 3: The gradient estimation $\hat{\mathbf{g}}_c(t)$ obtained by (5) 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, Algorithm 1 with the control law (6), 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 from the source $\mathbf{x}_s$:

$$\|\mathbf{x}^* - \mathbf{x}_s\| \leq \frac{2e_0}{\xi_1} \tag{8}$$

where ξ_1 and e_0 are defined in Assumption 2 and Assumption 3, respectively.

Proof: See Appendix I. ■

V. Distributed Source Seeking with Limited Communications

In Section IV, we developed a control algorithm for cooperative source seeking. It requires all-to-all communications. Aiming at reducing communication burdens, in this section we develop a fully distributed control algorithm, which only requires neighbor-to-neighbor communications. That is, we impose the following assumption in this section:

Assumption 4: The communication topology is a connected undirected graph. For the ith robot, only information from its one-hop neighbors and that from itself are available to the control design of $\mathbf{u}_i$.

From (6), it is clear that all necessary information for the ith robot can be derived from its one-hop neighbors except $\hat{\mathbf{g}}_c$ and $\frac{1}{n}(\mathbf{1}^T \otimes \mathbf{I})\mathbf{v}$. We use consensus filters to estimate them in a distributed manner [12], [14]. With consensus filters, a robot is able to estimate the average of inputs, i.e. $\frac{1}{n}\sum_{j}^{n} \tau_j$, by running the following protocol on every robot

$$\dot{z}_i = \sum_{j \in \mathbb{N}(i)} a_{ij}(z_j - z_i) + \gamma(\tau_i - z_i) \tag{9}$$

where z_i is a scalar state maintained by the ith robot, $\mathbb{N}(i)$ denotes the neighbor set of the ith robot, a_{ij} is a positive constant, which satisfies $a_{ij} = a_{ji}$. γ is a positive constant and u_i is the scalar input to the ith robot. By running (9) 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 distributed manners, we first re-write the expression of $\theta(t)$ in (5) into average forms (without confusions, the time t aside the time varying variables are omitted.)

$$
\begin{aligned}
\theta &= \begin{bmatrix} \frac{1}{n}\mathbf{X}^T\mathbf{X} & \frac{1}{n}\mathbf{X}^T\mathbf{1} \\ \frac{1}{n}\mathbf{1}^T\mathbf{X} & 1 \end{bmatrix}^{-1} \begin{bmatrix} \frac{1}{n}\mathbf{X}^T\mathbf{y} \\ \frac{1}{n}\mathbf{1}^T\mathbf{y} \end{bmatrix} \\
&= \begin{bmatrix} \frac{1}{n}\sum_{i=1}^{n}\mathbf{x}_i\mathbf{x}_i^T & \frac{1}{n}\sum_{i=1}^{n}\mathbf{x}_i \\ \frac{1}{n}\sum_{i=1}^{n}\mathbf{x}_i^T & 1 \end{bmatrix}^{-1} \begin{bmatrix} \frac{1}{n}\sum_{i=1}^{n}\mathbf{x}_i p_i \\ \frac{1}{n}\sum_{i=1}^{n} p_i \end{bmatrix}
\end{aligned} \tag{10}
$$

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}, θ_{ei}, and $\hat{\mathbf{g}}_{cei}$, which are estimations of $\frac{1}{n}\sum_{i=1}^{n}\mathbf{x}_i\mathbf{x}_i^T$, $\frac{1}{n}\sum_{i=1}^{n}\mathbf{x}_i$, $\frac{1}{n}\sum_{i=1}^{n}\mathbf{x}_i p_i$, $\frac{1}{n}\sum_{i=1}^{n} p_i$, θ, and $\hat{\mathbf{g}}_c$ by the ith robot (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.):

$$
\begin{aligned}
\dot{\mathbf{Z}}_{1i} &= \sum\nolimits_{j\in\mathbb{N}(i)} a_{ij}(\mathbf{Z}_{1j} - \mathbf{Z}_{1i}) + \gamma(\mathbf{x}_i\mathbf{x}_i^T - \mathbf{Z}_{1i}) \\
\dot{\mathbf{z}}_{2i} &= \sum\nolimits_{j\in\mathbb{N}(i)} a_{ij}(\mathbf{z}_{2j} - \mathbf{z}_{2i}) + \gamma(\mathbf{x}_i - \mathbf{z}_{2i}) \\
\dot{\mathbf{z}}_{3i} &= \sum\nolimits_{j\in\mathbb{N}(i)} a_{ij}(\mathbf{z}_{3j} - \mathbf{z}_{3i}) + \gamma(\mathbf{x}_i p_i - \mathbf{z}_{3i}) \\
\dot{z}_{4i} &= \sum\nolimits_{j\in\mathbb{N}(i)} a_{ij}(z_{4j} - z_{4i}) + \gamma(p_i - z_{4i}) \\
\theta_{ei} &= \begin{bmatrix} \mathbf{Z}_{1i} & \mathbf{z}_{2i} \\ \mathbf{z}_{2i}^T & 1 \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{z}_{3i} \\ z_{4i} \end{bmatrix} \\
\hat{\mathbf{g}}_{cei} &= \begin{bmatrix} \mathbf{I} & \mathbf{0} \end{bmatrix} \theta_{ei}
\end{aligned} \tag{11}
$$

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

$$
\dot{\mathbf{z}}_{5i} = \sum_{j\in\mathbb{N}(i)} a_{ij}(\mathbf{z}_{5j} - \mathbf{z}_{5i}) + \gamma(\mathbf{v}_i - \mathbf{z}_{5i}) \tag{12}
$$

Replacing $\hat{\mathbf{g}}_c$ and $\frac{1}{n}(\mathbf{1}^T \otimes \mathbf{I})\mathbf{v}$ in (6) with $\hat{\mathbf{g}}_{cei}$ in (11) and z_{5i} in (12), the distributed control algorithm for the ith robot writes:

$$
\begin{aligned}
\mathbf{u}_i = &- \sum_{j\in\mathbb{N}(i)} \omega_{1ij}(\mathbf{x}_i - \mathbf{x}_j - \mathbf{x}_{vi} + \mathbf{x}_{vj}) + c_0\hat{\mathbf{g}}_{cei} \\
&- \sum_{j\in\mathbb{N}(i)} \omega_{2ij}(\mathbf{v}_i - \mathbf{v}_j) - c_1\mathbf{z}_{5i} - c_2\mathrm{sgn}(\mathbf{z}_{5i})
\end{aligned} \tag{13}
$$

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

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

VI. Simulations

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.

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 $k_d = 5$, $k_s = 1$, $d_0 = 1$, $\kappa = 1$ (see that paper for definitions of each parameter) and the potential function is chosen to be the one suggested in the paper. For our methods, Algorithm 1 and Algorithm 2, we choose $c_0 = 20$, $c_1 = 7$, $c_2 = 7$, $\gamma = 1$, $L_1 = 6L_0$, $L_2 = L_0$, $L_3 = 30L_0$ 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, 5 robots are employed. Robots start from different positions. For PGS and ES method, 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].

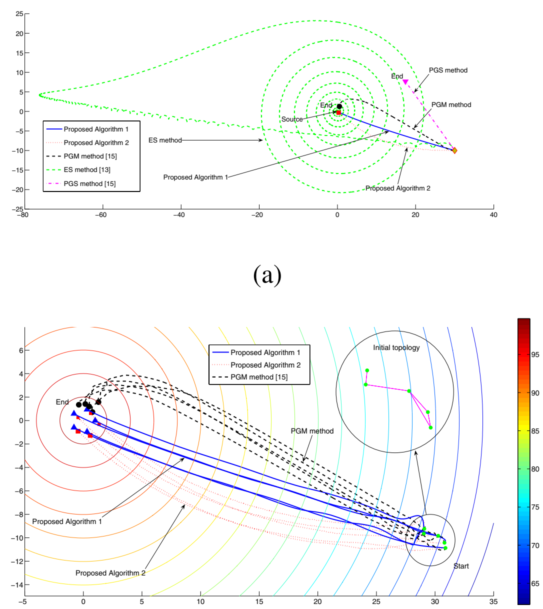
We first compare the trajectories of the formation center (for methods using a single robot, we use its trajectory for evaluation). From Fig. 1 (a), we can see that Algorithm 1, Algorithm 2, PGM method and EM method, are able to steer the formation center to the vicinity of the source. Among them, ES 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 before, Algorithm 2 is a fully distributed algorithm,

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

Require:
Concentration measurement p, position $\mathbf{x}$ and velocity $\mathbf{v}$ of itself and its neighbors are available to the ith robot.

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 $\mathbf{Z}_{1j}$, $\mathbf{z}_{2j}$, $\mathbf{z}_{3j}$, $\mathbf{z}_{4j}$, $\mathbf{z}_{5j}$.
6: State update of consensus filter $\mathbf{Z}_{1i}, \mathbf{z}_{2i}, \mathbf{z}_{3i}, \mathbf{z}_{4i}, \mathbf{z}_{5i}$ $\Leftarrow$ Equation (11) and Equation (12).
7: Distributed gradient estimation $\theta_{ei}, \hat{\mathbf{g}}_{cei}$ $\Leftarrow$ Equation (11).
8: Distributed control input $\mathbf{u}_i \Leftarrow$ Equation (13).
9: **until** $(\|\hat{\mathbf{g}}_{cei}(t)\| < \epsilon)$



(a)

(b)

Fig. 1. Simulation comparisons between PGS method [15], ES method [13], PGM method [15], Algorithm 1, and Algorithm 2. (a): Trajectory of formation centers (for 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 position 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.

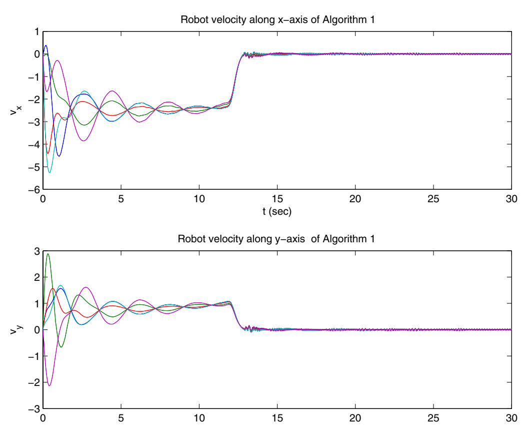


Fig. 2. Robot velocity profiles of Algorithm 1.

which only requires information exchanges between one-hop neighbors. Nevertheless, Algorithm 2 still outperforms PGS, PGM and ES methods in the sense that Algorithm 2 has a shorter trajectory than them.

We then compare Algorithm 1, Algorithm 2 and the PGM method, which use a group of robots, to see whether the desired formation are reached. From Fig. 1(b), we can see that PGM method does not reach a uniform distribution on

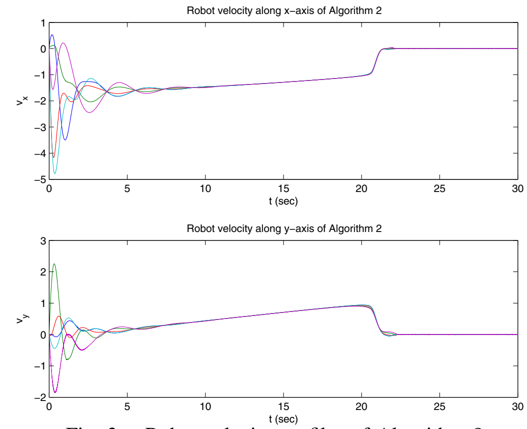


Fig. 3. Robot velocity profiles of Algorithm 2.

a circle, while both Algorithm 1 and Algorithm 2 reach the desired formation. The robot velocity profiles of our proposed methods are shown in Fig. 2 and Fig. 3. 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.

VII. Conclusions

The problem of source seeking is studied. Two control algorithms are given to solve the problem. Theoretical analysis proves that the proposed algorithm guarantees convergence to the source with all robots reaching the desired formation. Simulation results validate the theoretical results.

Appendix I
Proof of Theorem 1

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

Step 1: derivation of the formation center's dynamics.

Substituting the control input (7) into the robot dynamics (1) yields

$$\begin{aligned}\dot{\mathbf{v}} &= -(\mathbf{L}_1 \otimes \mathbf{I})(\mathbf{x}-\mathbf{x}_d) - (\mathbf{L}_2 \otimes \mathbf{I})\mathbf{v} + c_0 \mathbf{1} \otimes \hat{\mathbf{g}}_c \\ &\quad - \tfrac{c_1}{n}\mathbf{1} \otimes \big((\mathbf{1}^T \otimes \mathbf{I})\mathbf{v}\big) - c_2 \mathbf{1} \otimes \mathrm{sgn}\big((\mathbf{1}^T \otimes \mathbf{I})\mathbf{v}\big) \quad (14)\end{aligned}$$

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

$$\mathbf{x}_c = \frac{1}{n}\sum_{i=1}^{n} \mathbf{x}_i = \frac{1}{n}(\mathbf{1}^T \otimes \mathbf{I})\mathbf{x} \quad (15)$$

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 $(\mathbf{1}^T \otimes \mathbf{I})(\mathbf{L}_1 \otimes \mathbf{I}) = \mathbf{0}$, $(\mathbf{1}^T \otimes \mathbf{I})(\mathbf{L}_2 \otimes \mathbf{I}) = \mathbf{0}$, $(\mathbf{1}^T \otimes \mathbf{I})(\mathbf{1} \otimes \hat{\mathbf{g}}_c) = n\hat{\mathbf{g}}_c$, $(\mathbf{1}^T \otimes \mathbf{I})\Big(\mathbf{1} \otimes \big((\mathbf{1}^T \otimes \mathbf{I})\dot{\mathbf{x}}\big)\Big) = n^2\dot{\mathbf{x}}_c$, $(\mathbf{1}^T \otimes \mathbf{I})\Big(\mathbf{1} \otimes \mathrm{sgn}\big((\mathbf{1}^T \otimes \mathbf{I})\dot{\mathbf{x}}\big)\Big) = n\mathrm{sgn}(\dot{\mathbf{x}}_c)$. Based on these results, left multiplicating the matrix $\frac{1}{n}(\mathbf{1}^T \otimes \mathbf{I})$ on both sides of (14) yields the following dynamics of $\mathbf{x}_c$:

$$\ddot{\mathbf{x}}_c = c_0 \hat{\mathbf{g}}_c - c_1 \dot{\mathbf{x}}_c - c_2 \mathrm{sgn}(\dot{\mathbf{x}}_c) \quad (16)$$

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

$$\dot{\mathbf{v}}_c = c_0 \hat{\mathbf{g}}_c - c_1 \mathbf{v}_c - c_2 \text{sgn}(\mathbf{v}_c) \quad (17)$$

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 semi-positive definite since the Laplacian matrix $\mathbf{L}_1$ is semi-positive definite [12] and the eigenvalues of $\mathbf{A} \otimes \mathbf{B}$ is $\lambda_i \mu_j$ for all i and j with λ_i denoting the ith eigenvalue of the square matrix $\mathbf{A}$ while μ_j denoting the jth eigenvalue of the square matrix $\mathbf{B}$. In addition, $p(\mathbf{x}_s) - p(\mathbf{x}_c) \geq 0$ due to Assumption 1. Therefore, V is indeed semi-positive definite. Calculating the time derivative of V along the trajectory of (14) and (17) yields

$$\begin{aligned} \dot{V} = \quad & 2(\mathbf{x} - \mathbf{x}_d)^T (\mathbf{L}_1 \otimes \mathbf{I}) \mathbf{v} - \mathbf{g}_c^T \mathbf{v}_c + \frac{1}{c_0} \dot{\mathbf{v}}_c^T \mathbf{v}_c \\ & + 2(\dot{\mathbf{v}} - \mathbf{1} \otimes \dot{\mathbf{v}}_c)^T (\mathbf{v} - \mathbf{1} \otimes \mathbf{v}_c) \quad (18) \end{aligned}$$

From (14) and (17), we get

$$\begin{aligned} & \dot{\mathbf{v}} - \mathbf{1} \otimes \dot{\mathbf{v}}_c = -(\mathbf{L}_1 \otimes \mathbf{I})(\mathbf{x} - \mathbf{x}_d) - (\mathbf{L}_2 \otimes \mathbf{I}) \mathbf{v} \\ & -\tfrac{c_1}{n} \mathbf{1} \otimes \big((\mathbf{1}^T \otimes \mathbf{I}) \mathbf{v} \big) - \tfrac{c_2}{n} \mathbf{1} \otimes \text{sgn}\big((\mathbf{1}^T \otimes \mathbf{I}) \mathbf{v} \big) \\ & + c_1 \mathbf{1} \otimes \mathbf{v}_c + c_2 \mathbf{1} \otimes \text{sgn}(\mathbf{v}_c) \quad (19) \end{aligned}$$

Along with (15), we have

$$\dot{\mathbf{v}} - \mathbf{1} \otimes \dot{\mathbf{v}}_c = -(\mathbf{L}_1 \otimes \mathbf{I})(\mathbf{x} - \mathbf{x}_d) - (\mathbf{L}_2 \otimes \mathbf{I}) \mathbf{v} \quad (20)$$

Substituting (16) and (20) into (18) 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 get

$$\begin{aligned} \dot{V} & \leq -2\mathbf{v}^T (\mathbf{L}_2 \otimes \mathbf{I}) \mathbf{v} + \|\hat{\mathbf{g}}_c - \mathbf{g}_c\|_1 \|\mathbf{v}_c\|_1 - \tfrac{c_1}{c_0} \mathbf{v}_c^T \mathbf{v}_c - \tfrac{c_2}{c_0} \|\mathbf{v}_c\|_1 \\ & \leq -2\mathbf{v}^T (\mathbf{L}_2 \otimes \mathbf{I}) \mathbf{v} - \frac{c_1}{c_0} \mathbf{v}_c^T \mathbf{v}_c - (\frac{c_2}{c_0} - \sqrt{k} e_0) \|\mathbf{v}_c\|_1 \quad (21) \end{aligned}$$

Note that $\|\cdot\|$ denotes the Euclidean norm in (21). In the derivation of (21), 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 (21) is semi-negative definite by noting that $\mathbf{L}_2 \otimes \mathbf{I}$ is semi-positive definite and $c_2 > c_0 \sqrt{k} e_0$. Together with the fact that the right side of (21) is a function of state variables and the invariance-like theorem (Theorem 8.4 in [16]), we draw the conclusion that the right side of (21) goes to zeros as time elapses. Therefore,

$$\lim_{t \to \infty} \mathbf{v} = \mathbf{1} \otimes \alpha_1 \text{ and } \lim_{t \to \infty} \mathbf{v}_c = \mathbf{0} \quad (22)$$

where α_1 is a vector with k rows. With (16)and (20), we further conclude the following holds as time goes to infinity,

$$\hat{\mathbf{g}}_c \to \mathbf{0} \quad (23)$$

$$-(\mathbf{L}_1 \otimes \mathbf{I})(\mathbf{x} - \mathbf{x}_d) \to \mathbf{1} \otimes \dot{\alpha}_1 \quad (24)$$

For any $k \times 1$ vector α_2, we have $\dot{\alpha}_1^T \alpha_2 = 0$ by left multiplication of $\mathbf{1}^T \otimes \alpha_2^T$ on both sides of (24). Due to the arbitrary of the choice of α_2, we get $\dot{\alpha}_1 = \mathbf{0}$. In other words, we conclude that α_1 is a constant vector. With (24), we get $\mathbf{x} - \mathbf{x}_d \to \mathbf{1} \otimes \alpha_3$ as $t \to \infty$, where α_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 (23).

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

$$\begin{aligned} p(\mathbf{x}_s) - p(\mathbf{x}^*) \quad & \leq \quad \|\nabla p(\mathbf{x}^*)\| \|\mathbf{x}_s - \mathbf{x}^*\| - \frac{\xi_1}{2} \|\mathbf{x}_s - \mathbf{x}^*\|^2 \\ & = \quad \|\nabla p(\mathbf{x}^*) - \hat{\mathbf{g}}_c(\mathbf{x}^*)\| \|\mathbf{x}_s - \mathbf{x}^*\| \\ & \quad\quad - \frac{\xi_1}{2} \|\mathbf{x}_s - \mathbf{x}^*\|^2 \\ & \leq \quad e_0 \|\mathbf{x}_s - \mathbf{x}^*\| - \frac{\xi_1}{2} \|\mathbf{x}_s - \mathbf{x}^*\|^2 \quad (25) \end{aligned}$$

Note that Assumption 3 is employed in the above derivation of (25). Also note that the left side of (25) 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 *OCEANS 2007*, pp. 1 –10, 2007.

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

[3] K. Sugawara, T. Kazama, and T. Watanabe, "Foraging behavior of interacting robots with virtual pheromone," in *Intelligent Robots and Systems, 2004 IEEE/RSJ International Conference on*, vol. 3, pp. 3074 – 3079, 2004.

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

[5] J. Farrell, S. Pang, and W. Li, "Chemical plume tracing via an autonomous underwater vehicle," *Oceanic Engineering, IEEE Journal of*, vol. 30, no. 2, pp. 428 – 442, 2005.

[6] H. Ishida, H. Tanaka, H. Taniguchi, and T. Moriizumi, "Mobile robot navigation using vision and olfaction to search for a gas/odor source," in *Intelligent Robots and Systems, 2004 IEEE/RSJ International Conference on*, vol. 1, pp. 313 – 318, 2004.

[7] A. Hayes and A. Martinoli, "Distributed odor source localization," *Sensors Journal, IEEE*, vol. 2, no. 3, pp. 260 – 271, 2002.

[8] W. Spears, F. Diana, J. Hamann, and R. Heil, "Distributed, Physics-Based Control of Swarms of Vehicles," in *Autonomous Robots*, vol. 17, pp. 137 – 162, 2004.

[9] V. Gazi, K. M. Passino, and S. Member, "Stability analysis of social foraging swarms," *IEEE Trans. on Systems, Man and Cybernetics*, vol. 34, pp. 539–557, 2004.

[10] P. Ogren, E. Fiorelli, N. E. Leonard, and S. Member, "Cooperative control of mobile sensor networks: Adaptive gradient climbing in a distributed environment," *IEEE Transactions on Automatic Control*, vol. 49, pp. 1292–1302, 2004.

[11] R. Lopez, *Differential Geometry of Curves and Surfaces in Lorentz-Minkowski Space*, Notes of Mini-Course of University of Sao Paulo, Brasil, eprint arXiv:0810.3351, 2008.

[12] R. Freeman, P. Yang, and K. Lynch, "Stability and convergence properties of dynamic average consensus estimators," in *Decision and Control, 2006 45th IEEE Conference on*, pp. 338 –343, 2006.

[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. Yang, R. Freeman, and K. Lynch, "Distributed cooperative active sensing using consensus filters," in *Robotics and Automation, 2007 IEEE International Conference on*, pp. 405 –410, 2007.

[15] R. Bachmayer and N. Leonard, "Vehicle networks for gradient descent in a sampled environment," in *Decision and Control, 2002, Proceedings of the 41st IEEE Conference on*, vol. 1, pp. 112 – 117, 2002.

[16] H. K. Khalil, *Nonlinear Systems*, 3rd edition. Prentice Hall, 2001.