

# Consensus Learning for Distributed Coverage Control

Mac Schwager\*, Jean-Jacques Slotine†, and Daniela Rus\*

\*Computer Science and Artificial Intelligence Lab  
MIT, Cambridge, MA 02139

Email: schwager@mit.edu, rus@csail.mit.edu

†Nonlinear Systems Lab  
MIT, Cambridge, MA 02139

Email: jjs@mit.edu

**Abstract**—A decentralized controller is presented that causes a network of robots to converge to an optimal sensing configuration, while simultaneously learning the distribution of sensory information in the environment. A consensus (or flocking) term is introduced in the adaptation law to allow sharing of parameters among neighbors, greatly increasing parameter convergence rates. Convergence and consensus is proven using a Lyapunov-type proof. The controller with parameter consensus is shown to perform better than the basic controller in numerical simulations.

## I. INTRODUCTION

We present a decentralized controller to cause a group of robots to spread out over an environment in an optimal configuration for sensing. The robots position themselves in such a way that their density is greater in regions of the environment with more sensory interest and less in regions of less sensory interest. The controller simultaneously learns the distribution of sensory information in the environment while driving the robots to their optimal positions.

The controller improves upon the one described in [1] by allowing parameter information to be shared among neighboring robots. Specifically, a consensus term is introduced in the parameter adaptation laws to couple the adaptation among neighboring robots. The main effect of this coupling is that sensor measurements from any one robot propagate around the network to be used by all robots. Figure 1 shows an overview of the control scheme. We prove that the robots converge to an optimal configuration and their parameters reach a common value. The control laws we discuss are both adaptive and decentralized, thereby combining two of the defining qualities of biological systems. Our controller would be useful in controlling teams of robots to carry out a number of tasks including search and rescue missions, environmental monitoring (e.g. for forest fires), automatic surveillance of rooms, buildings, or towns, or simulating collaborative predatory behavior. Virtually any application in which a group of automated mobile agents is required to monitor an area could benefit from the proposed control law.

We present results from numerical simulations that demonstrate the effectiveness of the parameter consensus controller

in comparison to the basic controller in [1]. In particular, parameter convergence rates are greatly increased, and parameters for all robots in the network are guaranteed to converge to a common parameter vector.

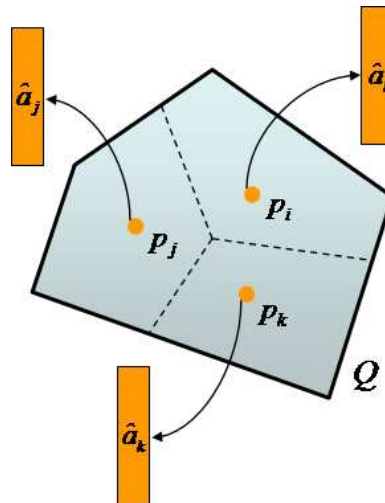


Fig. 1. A schematic of the overall control scheme is shown. The position of the robots ( $p_i, p_j$ , and  $p_k$ ) evolve to cover the space  $Q$ . Simultaneously, each robot adapts a parameter vector ( $\hat{a}_i, \hat{a}_j$ , and  $\hat{a}_k$ ) to build an approximation of the sensory environment. For the consensus controller, the parameter vectors are coupled among neighboring robots in such a way that their final value is the same for all robots.

### A. Relation to Previous Work

The coverage control literature most relevant to this work was initiated by [2], which introduced a formalism from locational optimization [3], and proposed a stable, decentralized control law to achieve an optimal coverage configuration. Other works have investigated variations upon this control law [4]–[6], however, in all of these works the robots are required to know *a priori* the distribution of sensory information in the environment. We previously relaxed this requirement by using a simple memoryless approximation from sensor measurements [7], though a stability proof was not found. In [1] we introduced an adaptive controller [8]–[10] with provable convergence properties in order to remove this requirement definitively.

This work was supported in part by the MURI SWARMS project grant number W911NF-05-1-0219, and NSF grant numbers IIS-0513755, IIS-0426838, and CNS-0520305.

Unfortunately, the controller from [1] suffered from slow parameter convergence in numerical simulations. We address this problem in the present work by including a consensus algorithm (sometimes called flocking, herding, swarming, agreement algorithms, gossip algorithms, rendezvous algorithms, oscillator synchronization, and other names) in the parameter adaptation law. Consensus phenomena have been studied in many fields, and appear ubiquitously in biological systems of all scales. However, they have only recently yielded to rigorous mathematical treatment; first in the distributed and parallel computing community [11]–[14] in discrete time, and more recently in the controls community in continuous time [15]–[21]. In the present work, consensus is used to learn the distribution of sensory information in the environment in a decentralized way by propagating sensory information gathered by each robot around the network. Consensus improves parameter convergence rates, which in turn causes the robots to converge more quickly to their optimal positions.

We set up the problem, provide some background on the results of locational optimization, and state the main assumptions in Section II. We present the basic controller and prove its convergence in Section III. Parameter consensus is introduced and convergence is proved in Section IV. In Section V we discuss and compare parameters convergence rates for the consensus and basic controllers. The results of numerical simulations are described in Section VI. Conclusions are given in Section VII.

## II. PROBLEM SET-UP

Let there be  $n$  robots in a convex polytope  $Q \subset \mathbb{R}^N$ . An arbitrary point in  $Q$  is denoted  $q$ , the position of the  $i^{\text{th}}$  robot is denoted  $p_i$ , and the set of all robot positions  $\{p_1, \dots, p_n\}$  is called the configuration of the network. Let  $\{V_1, \dots, V_n\}$  be the Voronoi partition of  $Q$ , for which the robot positions are the generator points. Specifically,

$$V_i = \{q \in Q \mid \|q - p_i\| \leq \|q - p_j\|, \forall j \neq i\}.$$

Define the sensory function as a map  $\phi(q) : Q \mapsto \mathbb{R}_+$  that determines a weighting of importance of points  $q \in Q$ . The function  $\phi(q)$  is not known by the robots in the network, but the robots are equipped with sensors from which a measurement of  $\phi(p_i)$  can be derived at the robot's position  $p_i$ .

Let the *unreliability* of the sensor measurement be defined by a quadratic function  $\frac{1}{2}\|q - p_i\|^2$ . Specifically,  $\frac{1}{2}\|q - p_i\|^2$  describes how unreliable is the measurement of the information at  $q$  by a sensor at  $p_i$  (henceforth,  $\|\cdot\|$  is used to denote the  $\ell^2$ -norm).

### A. Locational Optimization

In this section, we state the basic definitions and results from locational optimization that will be useful in this work. More thorough discussions can be found in [2], [3].

We can formulate the cost incurred by the network sensing over the region  $Q$  as

$$\mathcal{H}(P) = \sum_{i=1}^n \int_{V_i} \frac{1}{2} \|q - p_i\|^2 \phi(q) dq. \quad (1)$$

Notice that unreliable sensing is expensive and high values of  $\phi(q)$  are also expensive. An optimal network configuration corresponds to a set of robot positions that minimize (1).

Next we define three properties analogous to mass-moments of rigid bodies. The mass of  $V_i$  is defined as

$$M_{V_i} = \int_{V_i} \phi(q) dq, \quad (2)$$

the first mass-moment (not normalized) is defined as

$$L_{V_i} = \int_{V_i} q \phi(q) dq \quad (3)$$

and the centroid of  $V_i$  is defined as

$$C_{V_i} = \frac{L_{V_i}}{M_{V_i}}, \quad (4)$$

Note that  $\phi(q)$  strictly positive imply both  $M_{V_i} > 0 \forall V_i \neq \{\emptyset\}$  and  $C_{V_i} \in V_i \setminus \partial V_i$  ( $C_{V_i}$  is in the interior of  $V_i$ ). Thus  $M_{V_i}$  and  $C_{V_i}$  have properties intrinsic to physical masses and centroids. A standard result in locational optimization is that

$$\frac{\partial \mathcal{H}}{\partial p_i} = - \int_{V_i} (q - p_i) \phi(q) dq = -M_{V_i}(C_{V_i} - p_i). \quad (5)$$

Equation (5) implies that local minima of  $\mathcal{H}$  correspond to the configurations such that  $p_i = C_{V_i} \forall i$ , that is, each agent is located at the centroid of its Voronoi region. Thus, the optimal coverage task is to drive the group of robots to a centroidal Voronoi configuration—one in which each robot is positioned at the centroid of its Voronoi region.

### B. Assumptions

Let the robots have dynamics

$$\dot{p}_i = u_i, \quad (6)$$

where  $u_i$  is the control input. We can equivalently assume there is a low-level controller in place to cancel existing dynamics and enforce (6).

The robots also are able to compute their own Voronoi cell,  $V_i = \{q \mid \|q - p_i\| \leq \|q - p_j\|\}$ . This assumption is common in the literature [2], [4], [6], though it presents a practical conundrum. One does not know beforehand how far away the farthest Voronoi neighbor will be, thus this assumption cannot be translated into a communication range constraint (aside from the overly conservative requirement for each robot to have a communication range as large as the largest chord of  $Q$ ). In practice, only Voronoi neighbors within a certain distance will be in communication, in which case results can be derived, though with considerable complication [5]. Numerical simulations show that performance degrades gracefully with decreasing communication range among robots. We will take this assumption as implicit and leave the burden of relaxing this constraint for future work.

More central to this work, we assume that the sensory function  $\phi(q)$  can be parameterized as an unknown linear combination of a set of known basis functions. This requirement is formalized in the following two assumptions.

*Assumption 1 (Matching Conditions):*  $\exists a \in \mathbb{R}_+^m$  and  $\mathcal{K} : Q \mapsto \mathbb{R}_+^m$ , such that

$$\phi(q) = \mathcal{K}(q)^T a, \quad (7)$$

where the vector of basis functions  $\mathcal{K}$  is known by each agent, but the parameter vector  $a$  is unknown.

*Assumption 2 (Lower Bound):*

$$a(j) \geq a_{\min} \quad \forall j = 1, \dots, m, \quad (8)$$

where  $a(j)$  denotes the  $j^{\text{th}}$  element of the vector  $a$ , and  $a_{\min} > 0$  is a lower bound known by each agent.

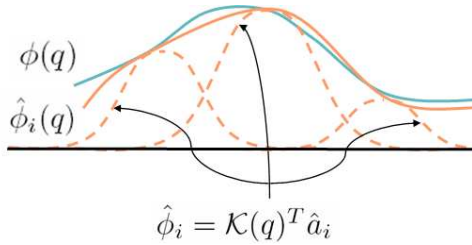


Fig. 2. The sensory function approximation is illustrated in this simplified 2-D schematic. The true sensory function is represented by  $\phi$  (blue line) and robot  $i$ 's approximation of the sensory function is  $\hat{\phi}_i$  (orange line). The vector  $\mathcal{K}(q)$  is shown as 3 Gaussians (dotted lines), and the parameter vector  $\hat{a}_i$  denotes the weighting of each Gaussian. According to Assumption 1 there is some value of  $\hat{a}_i$  that makes the approximation equal to the true function.

Let  $\hat{a}_i(t)$  be robot  $i$ 's approximation of the parameter vector. Naturally,  $\hat{\phi}_i = \mathcal{K}(q)^T \hat{a}_i$  is robot  $i$ 's approximation of  $\phi(q)$ . Figure 2 shows a graphical representation of this function approximation scheme. The figure shows the basis functions as Gaussians, since they are a common choice, though they could also be wavelets, sigmoids, splines, or any number of other function families. The choice is up to the designer's preference and the requirements of the application.

Define the mass moment approximations

$$\begin{aligned} \hat{M}_{V_i} &= \int_{V_i} \hat{\phi}_i dq, & \hat{L}_{V_i} &= \int_{V_i} q \hat{\phi}_i dq, \\ & & \text{and } \hat{C}_{V_i} &= \hat{L}_{V_i} / \hat{M}_{V_i}. \end{aligned} \quad (9)$$

Next, define the parameter error

$$\tilde{a}_i = \hat{a}_i - a, \quad (10)$$

and the sensory function error

$$\tilde{\phi}_i = \hat{\phi}_i - \phi = \mathcal{K}(q)^T \tilde{a}_i. \quad (11)$$

Finally, in order to compress the notation, we introduce the shorthand  $\mathcal{K}_i = \mathcal{K}(p_i(t))$  for the value of the basis function vector at the position of robot  $i$ , and  $\phi_i = \phi(p_i(t))$  for the value of  $\phi$  at the position of robot  $i$ . As previously stated, robot  $i$  can measure  $\phi_i$  with its sensors.

### III. DECENTRALIZED ADAPTIVE CONTROL LAW

We will design a control law with an intuitive interpretation and prove that it causes the network to converge to a near-centroidal Voronoi configuration. The control law will integrate sensory measurements available to each robot to form an on-line approximation of the centroid of its Voronoi region.

We propose to use the control law

$$u_i = k(\hat{C}_{V_i} - p_i), \quad (12)$$

where  $k \in \mathbb{R}_+$  is a proportional control gain. The parameters  $\hat{a}_i$  used to calculate  $\hat{C}_{V_i}$  are adjusted according to a set of adaptation laws which are introduced below.

Define two quantities,

$$\begin{aligned} \Lambda_i &= \int_0^t w(\tau) \mathcal{K}_i(\tau) \mathcal{K}_i(\tau)^T d\tau, \\ \text{and } \lambda_i &= \int_0^t w(\tau) \mathcal{K}_i(\tau) \phi_i(\tau) d\tau. \end{aligned} \quad (13)$$

The function  $w(t) \in \mathcal{L}^1$ , where  $w(t) \geq 0$ , determines a data collection weighting. Note that these quantities can be calculated differentially by robot  $i$  using  $\dot{\Lambda}_i = w(t) \mathcal{K}_i \mathcal{K}_i^T$ , and  $\dot{\lambda}_i = w(t) \mathcal{K}_i \phi_i$ , with zero initial conditions.

Define another quantity

$$F_i = \frac{\int_{V_i} \mathcal{K}(q)(q - p_i)^T dq \int_{V_i} (q - p_i) \mathcal{K}(q)^T dq}{\int_{V_i} \hat{\phi}_i(q) dq}. \quad (14)$$

Notice that  $F_i$  is a positive semi-definite matrix. It can also be computed by robot  $i$  as it does not require any knowledge of  $a$ .

The adaptation law for  $\hat{a}_i$  is defined as

$$\dot{\hat{a}}_{\text{pre}_i} = -F_i \hat{a}_i - \gamma(\Lambda_i \hat{a}_i - \lambda_i), \quad (15)$$

$$\dot{\hat{a}}_i = \Gamma(\hat{\hat{a}}_{\text{pre}_i} - I_{\text{proj}_i} \hat{\hat{a}}_{\text{pre}_i}), \quad (16)$$

where  $\Gamma \in \mathbb{R}^{m \times m}$  is a diagonal, positive definite adaptation gain matrix, and  $\gamma \in \mathbb{R}_+$  is an adaptation gain scalar. The diagonal matrix  $I_{\text{proj}_i}$  is defined element-wise as

$$I_{\text{proj}_i}(j) = \begin{cases} 0 & \text{for } \hat{a}_i(j) > a_{\min} \\ 0 & \text{for } \hat{a}_i(j) = a_{\min} \text{ and } \hat{\hat{a}}_{\text{pre}_i}(j) \geq 0 \\ 1 & \text{otherwise,} \end{cases} \quad (17)$$

where  $(j)$  denotes the  $j^{\text{th}}$  element for a vector and the  $j^{\text{th}}$  diagonal element for a matrix. Equations (16) and (17) implement a projection operation [10], [22] that prevents any element of  $\hat{a}_i$  from dropping below the lower bound  $a_{\min}$ . This is done by forcing  $\hat{\hat{a}}_{\text{pre}_i}(j) = 0$  whenever  $\hat{a}_i(j) = a_{\min}$  and  $\hat{\hat{a}}_{\text{pre}_i}(j) < 0$ . The projection is desirable for two reasons: 1) because the control law has a singularity at  $\hat{a}_i = 0$ , and 2) because we know from Assumption 2 that the true parameters are lower bounded by  $a_{\min}$ .

The controller described above will be referred to as the basic controller, and its behavior is formalized in the following theorem.

*Theorem 1 (Convergence Theorem):* Under Assumptions 1 and 2, for the system of agents with dynamics (6) and the control law (12),

$$i) \quad \lim_{t \rightarrow \infty} \|\hat{C}_{V_i}(t) - p_i(t)\| = 0 \quad \forall i \in \{1, \dots, n\} \quad (18)$$

$$ii) \quad \lim_{t \rightarrow \infty} \mathcal{K}_i^T(\tau) \tilde{a}_i(t) = 0 \quad \forall \tau \mid w(\tau) > 0 \quad (19)$$

and  $\forall i \in \{1, \dots, n\}$ .

**Proof**

We will define a lower-bounded function and show that it is non-increasing along the trajectories of the system, and that its time derivative is uniformly continuous. Theorem 1 is then an implication of Barbalat's lemma.

Let

$$\mathcal{V} = \mathcal{H} + \sum_{i=1}^n \frac{1}{2} \tilde{a}_i^T k \Gamma^{-1} \tilde{a}_i. \quad (20)$$

Taking the time derivative of  $\mathcal{V}$  along the trajectories of the system gives

$$\dot{\mathcal{V}} = - \sum_{i=1}^n \left[ \hat{M}_{V_i} k \|\hat{C}_{V_i} - p_i\|^2 + \tilde{a}_i^T k I_{\text{proj}_i} \dot{\hat{a}}_{\text{pre}_i} + \right. \quad (21)$$

$$\left. k \gamma \int_0^t w(\tau) (\mathcal{K}_i^T(\tau) \tilde{a}_i(t))^2 d\tau \right],$$

Inside the sum, the first and third terms are clearly non-negative. We focus momentarily on the second. Expanding it as a sum of scalar terms, we see that the  $j^{\text{th}}$  scalar term is of the form

$$k \tilde{a}_i(j) I_{\text{proj}_i}(j) \dot{\hat{a}}_{\text{pre}_i}(j). \quad (22)$$

From (17), if  $\hat{a}_i(j) > a_{\min}$ , or  $\hat{a}_i(j) = a_{\min}$  and  $\dot{\hat{a}}_{\text{pre}_i}(j) \geq 0$ , then  $I_{\text{proj}_i}(j) = 0$  and the term vanishes. Now, in the case  $\hat{a}_i(j) = a_{\min}$  and  $\dot{\hat{a}}_{\text{pre}_i}(j) < 0$ , we have  $\tilde{a}_i(j) = \hat{a}_i(j) - a(j) \leq 0$  (from Assumption 2). Furthermore,  $I_{\text{proj}_i}(j) = 1$  and  $\dot{\hat{a}}_{\text{pre}_i}(j) < 0$  implies that the term is non-negative. In all cases, then, each term of the form (22) is non-negative, and all three terms inside the sum in (21) are non-negative. Thus  $\dot{\mathcal{V}} \leq 0$ .

Also, the facts that  $u_i$  is continuous  $\forall i$ ,  $\mathcal{V}$  has continuous first partial derivatives,  $\mathcal{V}$  is radially unbounded, and  $\dot{\mathcal{V}} \leq 0$  imply that  $\dot{\mathcal{V}}$  is uniformly continuous, therefore, by Barbalat's lemma  $\lim_{t \rightarrow \infty} \dot{\mathcal{V}} = 0$ , which directly implies (18) from Theorem 1, and

$$\lim_{t \rightarrow \infty} \int_0^t w(\tau) (\mathcal{K}_i^T(\tau) \tilde{a}_i(t))^2 d\tau = 0 \quad (23)$$

$\forall i = 1, \dots, n$ .

Now notice that the integrand in (23) is non-negative, therefore it must converge to zero for all  $\tau$ , which implies (19) from Theorem 1.  $\square$

*Remark 1:* The first assertion (18) of Theorem 1 implies convergence to what we call a near-optimal sensing configuration. The *estimated* position errors go to zero, but not necessarily the *true* position errors. For the robots to converge to the true centroids of their Voronoi regions, an extra persistent excitation condition must be satisfied.

*Remark 2:* The second assertion (19) of Theorem 1 states that the sensory function estimate  $\hat{\phi}_i$  will converge asymptotically to the true sensory function  $\phi$  for all points on the robot's trajectory with positive weighting  $w(\tau)$ . This does not, however, imply that  $\hat{\phi}_i(q) \rightarrow \phi(q) \forall q \in Q$ . Again, this would require an extra persistent excitation condition.

*A. Weighting Functions*

The form of the function  $w(\cdot)$  can be designed to encourage parameter convergence. One obvious choice is to make  $w(\tau)$  a square wave, such that data is not incorporated into  $\int_0^t w(\tau) \mathcal{K}_i \mathcal{K}_i^T d\tau$  after some fixed time. This can be generalized to an exponential decay,  $w(\tau) = \exp(-\tau)$ , or a decaying sigmoid  $w(\tau) = 1/2(\text{erf}(c-t) + 1)$ . Many other options exist.

One intuitive option for  $w(\cdot)$  is  $w(\tau) = \|\dot{p}_i\|^2$ , since the rate at which new data is collected is directly dependent upon the rate of travel of the robot. This weighting, in a sense, normalizes the effects of the rate of travel so that all new data is incorporated with equal weighting. Likewise, when the robot comes to a stop, the value of  $\phi(p_i)$  at the stopped position does not overwhelm the learning law. This seems to make good sense, but there is an analytical technicality: to ensure that  $\Lambda_i$  and  $\lambda_i$  remain bounded we have to prove that  $\dot{p}_i \in L_2$ . In practice, we can set  $w(\tau) = \|\dot{p}_i\|^2$  up to some fixed time, after which it is zero.

We can also set  $w(t, \tau) = \exp\{-(t-\tau)\}$ , which turns the integrators  $\Lambda_i$  and  $\lambda_i$  into first order systems. This essentially introduces a forgetting factor into the learning law which has the advantage of being able to track slowly varying sensory distributions.

IV. PARAMETER CONSENSUS

In this section we first state some elementary properties of graph Laplacians, then use these properties to prove convergence and consensus of a modified adaptive control law. The controller from (III) is modified so that the adaptation laws among Voronoi neighbors are coupled. A similar idea was introduced in [16], distinguishing knowledge leaders from power leaders in flocks.

*A. Graph Laplacians*

A graph  $G = (V, E)$  is defined by a set of indexed vertices  $V = \{v_1, \dots, v_n\}$  and a set of edges  $E = \{e_1, \dots, e_l\}$ ,  $e_i = \{v_j, v_k\}$ . In the context of our application, a graph is induced in which each agent is identified with a vertex, and an edge exists between any two agents that are Voronoi neighbors. This graph is that of the Delaunay triangulation

Let  $\mathcal{N}_i = \{j \mid \{v_i, v_j\} \in E\}$  be the neighbor set of vertex  $v_i$ . Then  $|\mathcal{N}_i|$  is the number of Voronoi neighbors of agent  $i$ . Let  $A$  be the adjacency matrix of  $G$ , defined element wise by

$$A(i, j) = A(j, i) = \begin{cases} 1 & \text{for } \{v_i, v_j\} \in E \\ 0 & \text{otherwise.} \end{cases}$$

The graph Laplacian is defined as  $L = \text{diag}_{i=1}^n (|\mathcal{N}_i|) - A$ .

Loosely, a graph is connected if there exists a set of edges that defines a path between any two vertices. The

graph of any triangulation is connected, specifically, the graph is connected in our application. It is well known that for a connected graph  $L \geq 0$  and  $L$  has exactly one zero eigenvalue, with the associated eigenvector  $\mathbf{1} = [1, \dots, 1]^T$ . In particular,  $L\mathbf{1} = \mathbf{1}^T L = 0$ , and  $x^T L x > 0, \forall x \neq c\mathbf{1}, c \in \mathbb{R}$ . These properties will be important in what follows.

### B. Consensus Learning Law

We add a term to the parameter adaptation law in (15) to couple the adaptation of parameters among neighboring agents. Let the new adaptation law be given by

$$\dot{\hat{a}}_{\text{pre}_i} = -F_i \hat{a}_i - \gamma (\Lambda_i \hat{a}_i - \lambda_i) - \zeta \sum_{j \in \mathcal{N}_i} (\hat{a}_i - \hat{a}_j), \quad (24)$$

where  $\mathcal{N}_i$  is the neighbor set defined above, and  $\zeta \in \mathbb{R}_+$  is a positive gain. The projection remains the same as in (16), namely

$$\dot{\hat{a}}_i = \Gamma (\dot{\hat{a}}_{\text{pre}_i} - I_{\text{proj}_i} \hat{a}_{\text{pre}_i}). \quad (25)$$

#### Theorem 2 (Convergence with Parameter Consensus):

Under the conditions of Theorem 1, using the parameter adaptation law (24), the two claims from Theorem 1 are true. Additionally,

$$\lim_{t \rightarrow \infty} (\hat{a}_i - \hat{a}_j) = 0 \quad \forall i, j \in \{1, \dots, n\}. \quad (26)$$

#### Proof

We will use the same method as in the proof of Theorem 1, adding the extra term for parameter coupling. It will be shown that this term is non-positive. The claims of the proof follow as before from Barbalat's lemma.

Define  $\mathcal{V}$  to be (20), which leads to

$$\begin{aligned} \dot{\mathcal{V}} = & - \sum_{i=1}^n \left[ \hat{M}_{V_i} k \|\hat{C}_{V_i} - p_i\|^2 + \tilde{a}_i^T k I_{\text{proj}_i} \dot{\hat{a}}_{\text{pre}_i} + \right. \\ & \left. k\gamma \int_0^t w(\tau) (\mathcal{K}_i^T(\tau) \tilde{a}_i(t))^2 d\tau \right] - \\ & \sum_{i=1}^n \tilde{a}_i^T k \zeta \sum_{j \in \mathcal{N}_i} (\hat{a}_i - \hat{a}_j). \end{aligned} \quad (27)$$

We have already shown that the three terms inside the first sum are nonnegative. Now consider the parameter coupling term. We can rewrite this term using the graph Laplacian defined in Section IV-A as

$$\sum_{i=1}^n \tilde{a}_i^T k \zeta \sum_{j \in \mathcal{N}_i} (\hat{a}_i - \hat{a}_j) = k \zeta \sum_{j=1}^m \tilde{\alpha}_j^T L(t) \hat{\alpha}_j,$$

where  $\alpha_j = a(j)\mathbf{1}$ ,  $\hat{\alpha}_j = [\hat{a}_1(j) \ \dots \ \hat{a}_n(j)]^T$ , and  $\tilde{\alpha}_j = \hat{\alpha}_j - \alpha_j$ . Recall the ideal parameter vector  $a = [a(1) \ \dots \ a(j) \ \dots \ a(m)]^T$ , and the parameter estimate for each agent  $\hat{a}_i = [\hat{a}_i(1) \ \dots \ \hat{a}_i(j) \ \dots \ \hat{a}_i(m)]^T$ . We have simply regrouped the parameters by introducing the  $\alpha_j$  notation. The Laplacian is a function of time since as the agents move around they may acquire new neighbors or lose old ones. Fortunately, we are guaranteed that  $L(t)$  will have the properties discussed in Section IV-A for all  $t \geq 0$ .

From Section IV-A we saw that  $\alpha_j^T L(t) = a(j)\mathbf{1}^T L = 0$ . This gives

$$k \zeta \sum_{j=1}^m \tilde{\alpha}_j^T L \hat{\alpha}_j = k \zeta \sum_{j=1}^m \hat{\alpha}_j^T L \hat{\alpha}_j \geq 0,$$

since  $L(t) \geq 0 \ \forall t \geq 0$ .

Thus  $\mathcal{V}$  is again negative semi-definite. The previous argument still applies for the uniform continuity of  $\dot{\mathcal{V}}$ , therefore, by Barbalat's lemma  $\lim_{t \rightarrow \infty} \dot{\mathcal{V}} = 0$ . As before this implies the two claims of Theorem 1. Since the graph Laplacian is positive semi-definite, and  $\hat{a}_i(j) \geq a_{\min}$ ,  $\lim_{t \rightarrow \infty} \hat{\alpha}_j^T L(t) \hat{\alpha}_j = 0 \Rightarrow \lim_{t \rightarrow \infty} \hat{\alpha}_j = a_{\text{final}}(j)\mathbf{1} \ \forall j \in \{1, \dots, m\}$ , where  $a_{\text{final}} \in \mathbb{R}^m$  is some undetermined vector, which is the common final value of the parameters for all of the agents. The consensus assertion (26) follows.  $\square$

*Remark 3:* To guarantee that  $a_{\text{final}} = a$ , an extra persistent excitation condition must be met.

*Remark 4:* Introducing parameter coupling greatly increases parameter convergence rates and makes the controller equations better conditioned for numerical integration, as will be discussed in Section VI. There is, however, a small price in communication overhead. With the basic controller, the robots only have to communicate their position (2 floating point numbers) among Voronoi neighbors. With the parameter consensus controller they must communicate both their position and their parameter vector (2 +  $m$  floating point numbers). Even with a very low bandwidth communication system, this should represent a negligible cost.

## V. PARAMETER CONVERGENCE ANALYSIS

In this section we show that parameter convergence is not exponential, though it can be represented as a stable linear system driven by a signal that converges to zero. In other words, parameter convergence has a number of exponential modes and a number of (presumably slower) asymptotic modes. The exponential modes are shown to be faster for the controller with parameter consensus. In this section we neglect the projection operation (16), as the discrete switching considerably complicates the convergence analysis.

From (15), we have

$$\dot{\hat{a}}_i = -\Gamma (F_i \hat{a}_i + \gamma (\Lambda_i \hat{a}_i - \lambda_i)).$$

We can rewrite this as

$$\dot{\tilde{a}}_i = -\Gamma \gamma \int_0^t w(\tau) \mathcal{K}_i \mathcal{K}_i^T d\tau \tilde{a}_i - \Gamma F_i \hat{a}_i,$$

which is clearly a linear system in  $\tilde{a}_i$  driven by the term  $-\Gamma F_i \hat{a}_i$ . If the robot trajectory is such that  $\int_0^t w(\tau) \mathcal{K}_i \mathcal{K}_i^T d\tau$  is positive definite, the linear system has only real, strictly negative eigenvalues. It therefore behaves like an exponentially stable system driven by the signal  $-\Gamma F_i \hat{a}_i$ . In this case we call the robot's trajectory persistently exciting. We proved in Theorem 1 that  $(\hat{C}_{V_i} - p_i) \rightarrow 0$  and all other quantities in  $F_i \hat{a}_i$  are bounded, therefore  $F_i \hat{a}_i \rightarrow 0$ , but we cannot prove that it does so exponentially. However, the gains  $\Gamma$  and  $\gamma$  can be set such that  $\Gamma F_i \hat{a}_i$  is arbitrarily small compared to

$\Gamma\gamma \left[ \int_0^t w(\tau) \mathcal{K}_i \mathcal{K}_i^T d\tau \right]$  without affecting stability. Thus exponentially fast convergence to an arbitrarily small parameter error can be achieved.

For the parameter consensus controller, from (24) we have

$$\dot{\hat{a}}_i = -\Gamma \left( F_i \hat{a}_i + \gamma (\Lambda_i \hat{a}_i - \lambda_i) + \zeta \sum_{j \in \mathcal{N}_i} (\hat{a}_i - \hat{a}_j) \right),$$

For the basic controller, parameter convergence and persistence of excitation are not coupled among robots, but are determined on a robot by robot basis. This is not the case for the parameter consensus control law. To analyze parameter convergence for this case, we must consider a concatenated vector consisting of all the robots' parameter errors

$$\tilde{A} = [\tilde{a}_1^T \quad \cdots \quad \tilde{a}_n^T]^T.$$

Also, define the block diagonal matrices  $F = \text{diag}_{i=1}^n(\Gamma F_i)$ ,  $K = \text{diag}_{i=1}^n(\Gamma \int_0^t w(\tau) \mathcal{K}_i \mathcal{K}_i^T d\tau)$ , and the generalized graph Laplacian matrix

$$\mathcal{L} = \begin{bmatrix} \Gamma(1)L(1,1)I_m & \cdots & L(1,n)I_m \\ \vdots & \ddots & \vdots \\ L(n,1)I_m & \cdots & \Gamma(n)L(n,n)I_m \end{bmatrix}.$$

The matrix  $\mathcal{L}$  can be thought of as  $\Gamma L$  with each entry multiplied by the  $m \times m$  identity matrix. Then the coupled dynamics of the parameters over the whole network can be written

$$\dot{\tilde{A}} = -(\gamma K + \zeta \mathcal{L})\tilde{A} - F\tilde{A},$$

with  $\hat{A}$  defined in the obvious way. Again this is a linear system in  $\tilde{A}$  driven by a term that converges to zero. The eigenvalues of  $\mathcal{L}$  are the same as those of  $\Gamma L$ , but each eigenvalue has multiplicity  $m$ . As for a typical graph Laplacian,  $\mathcal{L}$  is positive semi-definite, and has  $m$  zero eigenvalues. Therefore, the trajectory of the network is persistently exciting if  $\gamma K + \zeta \mathcal{L}$  is positive definite. This is a less restrictive condition than for the basic controller. Furthermore, if parameter convergence takes place for the basic controller, then it will occur *more quickly* for the parameter consensus controller, since  $\mathcal{L}$  always contributes a stabilizing affect. As before, convergence is presumably limited by the non-exponential driving term  $F\tilde{A}$ , though this term can be made arbitrarily small by choosing  $\Gamma$  small, and  $\gamma$  and  $\zeta$  correspondingly large.

## VI. NUMERICAL SIMULATIONS

### A. Practical Algorithm

A practical method for implementing the proposed control law on a network of robots is detailed in Algorithm 1. Notice that the control law in (12) and adaptation law in (16) both require the computation of integrals over  $V_i$ , thus robot  $i$  must be able to calculate continuously its Voronoi region. Several algorithms exist for computing  $V_i$  in a distributed fashion, for example those given in [2], [23].

Algorithm 1 is fully distributed and can be used on teams of large robots, on teams of small robots such as [24], or on mobile sensor network nodes with limited computation and storage capabilities such as the mobile Mica Motes described by [25].

---

### Algorithm 1 Adaptive Coverage Control Algorithm

---

**Require:** Each robot can compute its Voronoi region

**Require:**  $\phi(q)$  can be parameterized as in (7)

**Require:**  $a(j)$  are lower bounded as in (8)

Initialize  $\Lambda_i$ ,  $\lambda_i$  to zero, and  $\hat{a}_i(j)$  to  $a_{\min}$

**loop**

    Compute the robot's Voronoi region

    Compute  $\hat{C}_{V_i}$  according to (9)

    Update  $\hat{a}_i$  according to (16)

    Update  $\Lambda_i$  and  $\lambda_i$  according to (13)

    Apply control input  $u_i = -k(\hat{C}_{V_i} - p_i)$

**end loop**

---

### B. Implementation

Simulations were carried out in a Matlab environment. The dynamics in (6) with the control law in (12), and the adaptation laws in (16) and (13) for a group of  $n = 20$  robots were modeled as a system of coupled differential equations. The fixed-time-step numerical solver was used to integrate the equations of motion of the group of robots. The region  $Q$  was taken to be the unit square. The sensory function,  $\phi(q)$ , was parameterized as a Gaussian network with 9 Gaussians. In particular, for  $\mathcal{K} = [\mathcal{K}(1) \quad \cdots \quad \mathcal{K}(9)]^T$ , each component  $\mathcal{K}(j)$  was implemented as

$$\mathcal{K}(j) = \frac{1}{\sigma_j \sqrt{2\pi}} \exp \left\{ -\frac{(q - \mu_j)^2}{2\sigma_j^2} \right\}, \quad (28)$$

where  $\sigma_j = .18$ . The unit square was divided into an even  $3 \times 3$  grid and each  $\mu_j$  was chosen so that one of the 9 Gaussians was centered at the middle of each grid square. The parameters were chosen as  $a = [100 \quad a_{\min} \quad \cdots \quad a_{\min} \quad 100]^T$ , with  $a_{\min} = .1$  so that only the lower left and upper right Gaussians contributed significantly to the value of  $\phi(q)$ , producing a bimodal distribution.

The robots in the network were started from random initial positions. Each robot used a copy of the Gaussian network described above for  $\mathcal{K}(q)$ . The estimated parameters  $\hat{a}_i$  for each robot were started at a value of  $a_{\min}$ , and  $\Lambda_i$  and  $\lambda_i$  were each started at zero. The gains used by the robots were  $k = 3$ ,  $\Gamma = I_{10}$ ,  $\gamma = 1000$  for the basic controller, and  $\gamma = 100$  and  $\zeta = 5$  for the consensus controller. In practice, the first integral term in the adaptive law (15) seems to have very little effect on the performance of the controller. Choosing  $\Gamma$  small and  $\gamma$  comparatively large puts more weight on the second term, which is responsible for integrating measurements of  $\phi(p_i)$  into the parameters. The spatial integrals in (9) and (15) required for the control law were computed by discretizing each Voronoi region  $V_i$  into a  $7 \times 7$  grid and summing contributions of the integrand over the grid. Voronoi regions were computed using a decentralized algorithm similar to the one in [2].

### C. Simulation Results

Figure 3 shows the positions of the robots in the network over the course of a simulation run for the parameter

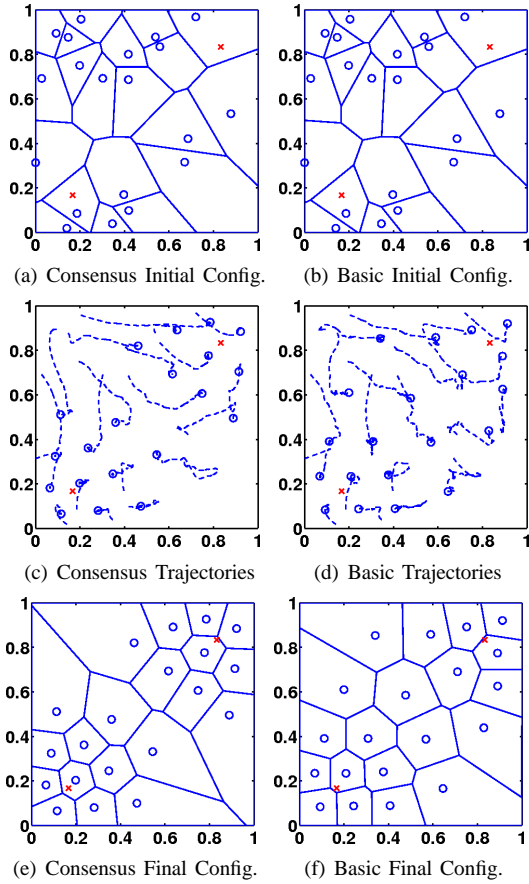
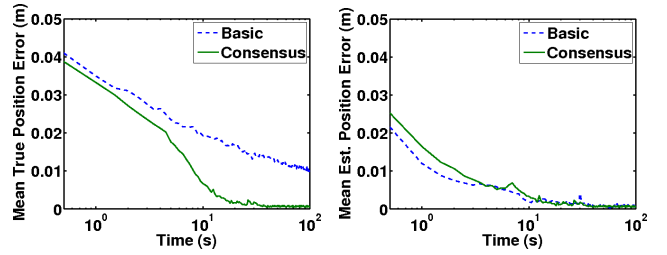


Fig. 3. Simulation results for the parameter consensus controller are shown in the left column (3(a), 3(c), and 3(e)), and for the basic controller in the right column (3(b), 3(d), and 3(f)). The Gaussian centers of  $\phi(q)$  are marked by the red x's.

consensus controller (left column) and the basic controller (right column). The centers of the two contributing Gaussian functions are marked with  $\times$ s. It is apparent from the final configurations that the consensus controller caused the robots to group more tightly around the Gaussian peaks than the basic controller. The somewhat jagged trajectories are caused by the discrete nature of the spatial integration procedure used to compute the control law.

We now investigate quantitative metrics to compare the performance of the consensus and basic controllers. Note that for all metrics shown, the convergence time scales are so different for the two controllers that a logarithmic scale had to be used on the time axis to display both curves on the same plot.

The right of Fig. 4 shows that both controllers achieve a near-optimal configuration—one in which the estimated error converges to zero, in accordance with (18) of Theorem 1. However, the true position error also converged to zero for the consensus controller, indicating that it achieved a true centroidal Voronoi configuration, as shown in the left of Fig. 4. The basic controller did not reach a true centroidal Voronoi configuration. Again, the somewhat jagged time history is a result of the discretized spatial integral computation over the Voronoi region.



(a) Mean True Position Error (b) Mean Estimated Position Error

Fig. 4. The true position error,  $\|C_{V_i} - p_i\|$ , and the estimated position error,  $\|\hat{C}_{V_i} - p_i\|$ , averaged over all the robots in the network is shown for the network of 20 robots for both the basic and parameter sharing controllers. The true position error converges to zero only for the parameter consensus controller, 4(a). However, in accordance with Theorem 1, the estimated error converges to zero in both cases, 4(b). Note the logarithmic time scale.

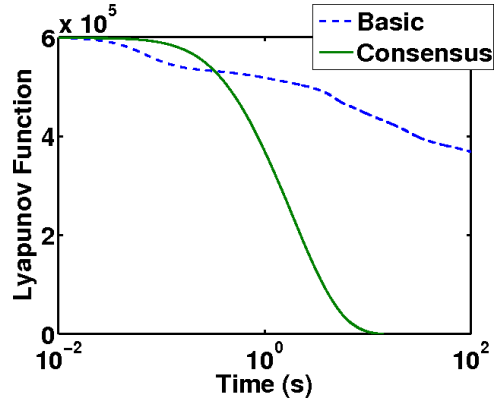


Fig. 5. The Lyapunov function is shown for both the basic and parameter consensus controller. Notice that the parameter consensus controller results in a faster decrease and a lower final value of the function.

Fig. 5 shows that the consensus controller obtained a lower value of the Lyapunov function at a faster rate than the basic controller, indicating both a lower-cost configuration and a better function approximation. The final value for the consensus controller is not zero, as it appears to be in plot, but is several orders of magnitude less than the final value for the basic controller.

Figure 6 shows the normed parameter error  $\|\hat{a}_i\|$  averaged over all of the robots. The parameter errors for the consensus controller all converge to zero, indicating that, in fact, persistent excitation was achieved. This was also evidenced in Fig. 4(a). For the basic controller, on the other hand, the parameters did not converge to the true parameters.

Finally, the disagreement among the parameter values of robots is shown in Fig. 7. The larger the value in the plot, the more different the parameters are from one another. The parameters were initialized to  $a_{\min}$  for all robots, so this value starts from zero in both cases. However, the consensus controller clearly causes the parameters to reach consensus, while for the basic controller the parameters do not converge to a common value.

## VII. CONCLUSION

In this work we introduced parameter coupling into an existing decentralized adaptive control law to drive a network of

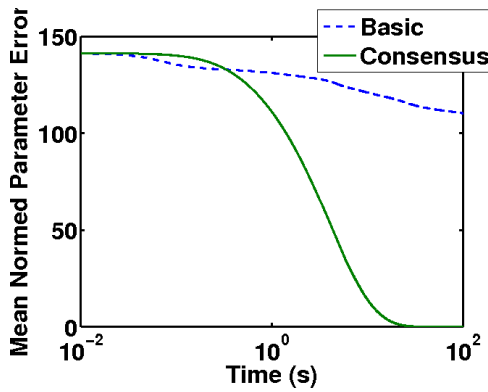


Fig. 6. The normed parameter error  $\|\tilde{a}_i\|$  averaged over all robots is shown for both the basic and parameter consensus controllers. Notice that the parameter error converges to zero with the consensus controller indicating that the robot trajectories were persistently exciting.

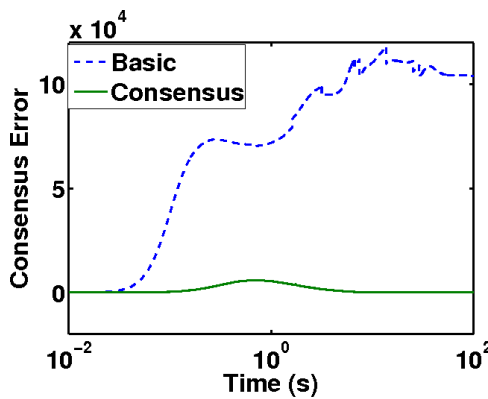


Fig. 7. The quantity  $\sum_{i=1}^n \tilde{a}_i^T \sum_{j \in \mathcal{N}_i} (\hat{a}_i - \hat{a}_j)$  is shown, representing a measure of the disagreement of parameters among robots. The disagreement converges to zero for the consensus controller, as asserted in Theorem 2, but does not converge for the basic controller.

robots to a near-optimal sensing configuration. The controller was proven to cause the robots to move to the estimated centroids of their Voronoi regions, while also causing their estimate of the sensory distribution to improve over time until the estimate converged to the true sensory distribution over the robot's trajectory. Parameter coupling was introduced in the adaptation laws to increase parameter convergence rates and cause consensus among the robots in the network for final parameter values. The control law was demonstrated in numerical simulations of a group of 20 robots sensing over an area with a bimodal Gaussian distribution of sensory information.

We expect that the technique used in this paper will find broader application beyond the problem chosen here. It appears that consensus algorithms could be a fundamental and practical tool for enabling distributed learning, and has compelling parallels with distributed learning mechanisms in biological systems.

#### REFERENCES

[1] M. Schwager, J.-J. Slotine, and D. Rus, "Decentralized, adaptive control for coverage with networked robots," in *Proceedings of International Conference on Robotics and Automation*, Rome, April 2007.

[2] J. Cortés, S. Martínez, T. Karatas, and F. Bullo, "Coverage control for mobile sensing networks," *IEEE Transactions on Robotics and Automation*, vol. 20, no. 2, pp. 243–255, April 2004.

[3] Z. Drezner, *Facility Location: A Survey of Applications and Methods*, ser. Springer Series in Operations Research. New York: Springer-Verlag, 1995.

[4] S. Salapaka, A. Khalak, and M. A. Dahleh, "Constraints on locational optimization problems," in *Proceedings of Conference on Decision and Control*, Maui, Hawaii, USA, December 2003.

[5] J. Cortés, S. Martínez, and F. Bullo, "Spatially-distributed coverage optimization and control with limited-range interactions," *ESIAM: Control, Optimisation and Calculus of Variations*, vol. 11, pp. 691–719, 2005.

[6] A. Ganguli, J. Cortés, and F. Bullo, "Maximizing visibility in nonconvex polygons: nonsmooth analysis and gradient algorithm design," in *Proceedings of the American Control Conference*, Portland, OR, June 2005, pp. 792–797.

[7] M. Schwager, J. McLurkin, and D. Rus, "Distributed coverage control with sensory feedback for networked robots," in *Proceedings of Robotics: Science and Systems*, Philadelphia, PA, August 2006.

[8] J.-J. E. Slotine and W. Li, *Applied Nonlinear Control*. Upper Saddle River, NJ: Prentice-Hall, 1991.

[9] K. S. Narendra and A. M. Annaswamy, *Stable Adaptive Systems*. Englewood Cliffs, NJ: Prentice-Hall, 1989.

[10] P. A. Ioannou and J. Sun, *Robust Adaptive Control*. Englewood Cliffs, NJ: Prentice-Hall, 1996.

[11] J. N. Tsitsiklis, "Problems in decentralized decision making and computation," Ph.D. dissertation, Department of EECS, MIT, November 1984.

[12] J. N. Tsitsiklis, D. P. Bertsekas, and M. Athans, "Distributed asynchronous deterministic and stochastic gradient optimization algorithms," *IEEE Transactions on Automatic Control*, vol. 31, no. 9, pp. 803–812, 1986.

[13] D. Bertsekas and J. Tsitsiklis, *Parallel and Distributed Computation: Numerical Methods*. Prentice Hall, 1989.

[14] J. N. Tsitsiklis and D. P. Bertsekas, "Comment on 'coordination of groups of mobile autonomous agents using nearest neighbor rules'," *IEEE Transactions on Automatic Control*, in press.

[15] W. Wang and J. J. E. Slotine, "On partial contraction analysis for coupled nonlinear oscillators," *Biological Cybernetics*, vol. 23, no. 1, pp. 38–53, December 2004.

[16] —, "A theoretical study of different leader roles in networks," *IEEE Transactions on Automatic Control*, vol. 51, no. 7, pp. 1156–1161, July 2006.

[17] T. Vicsek, A. Czirok, E. Ben-Jacob, I. Cohen, and O. Shochet, "Novel type of phase transition in a system of self-driven particles," *Physical Review Letters*, vol. 75, no. 6, pp. 1226–1229, August 1995.

[18] V. D. Blondel, J. M. Hendrickx, A. Olshevsky, and J. N. Tsitsiklis, "Convergence in multiagent coordination, consensus, and flocking," in *Proceedings of the Joint IEEE Conference on Decision and Control and European Control Conference*, Seville, Spain, December 2005.

[19] R. Olfati-Saber and R. R. Murray, "Consensus problems in networks of agents with switching topology and time-delays," *IEEE Transactions on Automatic Control*, vol. 49, no. 9, pp. 1520–1533, September 2004.

[20] A. Jadbabaie, J. Lin, and A. S. Morse, "Coordination of groups of mobile autonomous agents using nearest neighbor rules," *IEEE Transactions on Automatic Control*, vol. 48, no. 6, pp. 988–1001, June 2003.

[21] F. Cucker and S. Smale, "Emergent behavior in flocks," *IEEE Transactions on Automatic Control*, vol. 52, no. 5, pp. 852–862, May 2007.

[22] J. Slotine and J. Coetsee, "Adaptive sliding controller synthesis for nonlinear systems," *International Journal of Control*, vol. 43, no. 4, 1986.

[23] Q. Li and D. Rus, "Navigation protocols in sensor networks," *ACM Transactions on Sensor Networks*, vol. 1, no. 1, pp. 3–35, Aug. 2005.

[24] J. McLurkin, "Stupid robot tricks: A behavior-based distributed algorithm library for programming swarms of robots," Master's thesis, MIT, 2004.

[25] G. T. Sibley, M. H. Rahimi, and G. S. Sukhatme, "Robomote: A tiny mobile robot platform for large-scale sensor networks," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2002.