2008 IEEE International Conference on
Robotics and Automation
Pasadena, CA, USA, May 19-23, 2008

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 a near optimal sensing configuration, while simultaneously learning the distribution of sensory information in the environment. A consensus (or flocking) term is introduced in the learning law to allow sharing of parameters among neighbors, greatly increasing learning 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. Each robot simultaneously learns the distribution of sensory information in the environment while driving to its near optimal position. The distribution of sensory information in the environment is learned by adapting a parameter vector based on sensor measurements.

The controller improves upon the one we previously presented in [1] by including a consensus term 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 a configuration that minimizes a cost function relevant to the robots' collective sensing ability over the environment. We also prove that 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.

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, CNS-0520305, CNS-0707601, and EFRI-0735953.

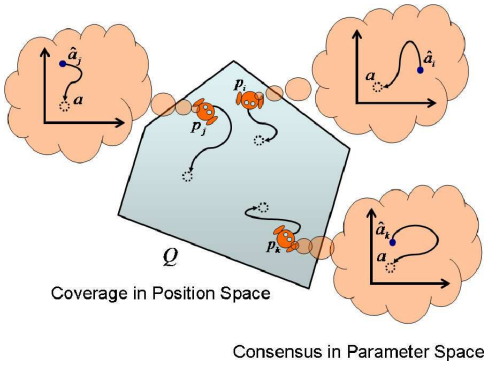


Fig. 1. A schematic of the overall control scheme is shown. The robots at $p_i, p_j,$ and p_k move to cover the area Q. Simultaneously, each robot adapts a parameter vector ($\hat{a}_i, \hat{a}_j,$ and $\hat{a}_k$) to build an approximation of the sensory environment. The parameter vectors are coupled 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], [5], 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 [6], though a stability proof was not found. In [1] we introduced an adaptive controller [7] with provable convergence properties in order to remove this requirement definitively.

However, the controller from [1] exhibited 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, 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 [8]–[11] in discrete time, and more recently in the controls community in continuous time [12]–[16]. 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, and provide the necessary background in Section II. We present the consensus controller and prove its convergence in Section III. In Section IV we discuss and compare parameter convergence rates for the consensus and basic controllers. The results of numerical simulations are described in Section V. Conclusions are given in Section VI.

II. Problem Set-up

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

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

(henceforth, $\|.\|$ is used to denote the ℓ^2-norm). We assume that the robots are able to compute their own Voronoi cell. This assumption is common in the literature [2], [4], 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. 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]. We will take this assumption on the communication range as implicit and leave the burden of relaxing this assumption for future work.

Next, define the sensory function as a map $\phi : Q \mapsto \mathbb{R}_+$ (where $\mathbb{R}_+$ refers to the strictly positive orthant) 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. The interpretation of the sensory function $\phi(q)$ is intentionally unspecific in order to suit a broad range of applications. For example, for a human surveillance application in which robots use audio sensors, $\phi(q)$ may be chosen to be the sound intensity of the frequency range corresponding to the human voice. In any case, $\phi(q)$ should be chosen to represent a weighting of importance over Q.

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.

A. Locational Optimization

Drawing from the field of locational optimization [3], we can formulate the cost incurred by the network sensing over the region Q as

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

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, first mass-moment, and centroid of V_i defined as

$$M_{V_i} = \int_{V_i} \phi(q)\, dq, \quad L_{V_i} = \int_{V_i} q\phi(q)\, dq, \tag{2}$$
$$\text{and } C_{V_i} = \frac{L_{V_i}}{M_{V_i}},$$

respectively. The moments and the Voronoi regions themselves depend on the robot positions. Remarkably, despite this dependency, a standard result [3] is that

$$\frac{\partial \mathcal{H}}{\partial p_i} = -\int_{V_i} (q - p_i)\phi(q)\, dq = -M_{V_i}(C_{V_i} - p_i). \tag{3}$$

Equation (3) 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.

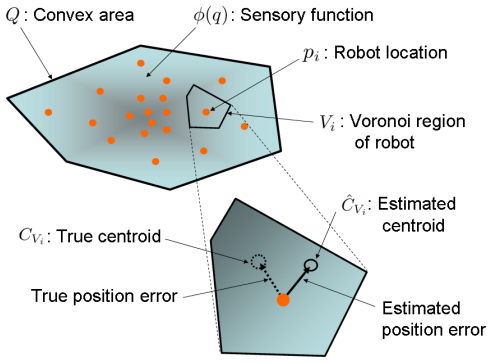


Fig. 2. A graphical overview of the quantities involved in the controller and environment is shown. The robots move to cover a bounded, convex area Q. Their positions are p_i, and they each have a Voronoi region V_i with a true centroid C_{V_i} and an estimated centroid $\hat{C}_{V_i}$. The true centroid is determined using a sensory function $\phi(q)$, which indicates the relative importance of points q in Q. The robots do not know $\phi(q)$, so they calculate an estimated centroid using an approximation $\hat{\phi}_i(q)$ learned from sensor measurements of $\phi(q)$.

B. Sensory Function Approximation

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 assumption.

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

$$\phi(q) = \mathcal{K}(q)^T a, \tag{4}$$

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

$$a(j) \geq a_{\min} \quad \forall j = 1, \ldots, m, \tag{5}$$

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

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 3 shows a graphical representation of this function approximation scheme.

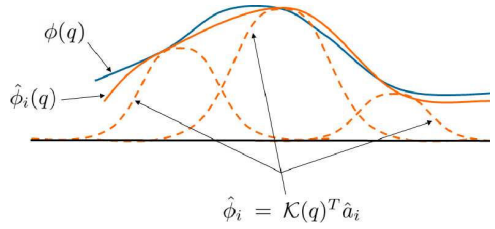


Fig. 3. The sensory function approximation is illustrated in this simplified 2-D schematic. The true sensory function is represented by ϕ (blue curve) and robot i's approximation of the sensory function is $\hat{\phi}_i$ (orange curve). The vector $\mathcal{K}(q)$ is shown as 3 Gaussians (dotted curves), 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.

Requirements such as Assumption 1 are common for adaptive controllers. In theory, the assumption is not limiting since any function (with some smoothness requirements) over a bounded domain can be approximated arbitrarily well by a network of basis functions [17]. In practice, however, designing a suitable function approximation network requires application-specific expertise. We use Gaussian basis functions in our simulations, but there is a variety of other basis function families to chose from including, wavelets, sigmoids, and splines. Gaussian basis functions have a computational advantage over non-local basis functions because they have nearly compact support. To compute the value of the network at a location $\hat{\phi}_i(q)$, or to tune the weights of the network $\hat{a}_i$ with new data, one has only to consider Gaussians in a region around the point of interest. Looking forward, we are also interested in adapting nonlinearly occurring parameters, such as Gaussian center locations and widths, similarly to [18].

Define the mass moment approximations

$$\hat{M}_{V_i} = \int_{V_i} \hat{\phi}_i \, dq, \quad \hat{L}_{V_i} = \int_{V_i} q \hat{\phi}_i \, dq, \tag{6}$$

$$\text{and } \hat{C}_{V_i} = \frac{\hat{L}_{V_i}}{\hat{M}_{V_i}}.$$

Next, define the parameter error

$$\tilde{a}_i = \hat{a}_i - a, \tag{7}$$

and the sensory function error

$$\tilde{\phi}_i = \hat{\phi}_i - \phi = \mathcal{K}(q)^T \tilde{a}_i. \tag{8}$$

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 ϕ at the position of robot i. As previously stated, robot i can measure ϕ_i with its sensors. Figure 2 illustrates the quantities involved in the problem.

C. Graph Laplacians

We here describe some of the basic properties of graph Laplacians, which will be used later to prove parameter convergence. A graph $G = (V, E)$ is defined by a set of indexed vertices $V = \{v_1, \ldots, v_n\}$ and a set of edges $E = \{e_1, \ldots, e_l\}$, $e_i = \{v_j, v_k\}$. 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. 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 [19] that for a connected graph $L \geq 0$ and L has exactly one zero eigenvalue, with the associated eigenvector $\mathbf{1} = [1, \ldots, 1]^T$. In particular, $L\mathbf{1} = \mathbf{1}^T L = 0$, $x^T L x \geq 0 \; \forall x$, and $x^T L x = 0$ imples $x = 0$ or $x = \mathbf{1}c$ for some $c \in \mathbb{R}$.

III. Adaptive Controller with Parameter Consensus

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, while causing the parameter vectors for all robots to converge to the same vector. The control law will integrate sensory measurements available to each robot to form an on-line approximation of the centroid of its Voronoi region. The controller is like the one from [1], with the key modification that the adaptation laws among Voronoi neighbors are coupled.

Let the robots have dynamics

$$\dot{p}_i = u_i, \tag{9}$$

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 (9). We propose to use the control law

$$u_i = K(\hat{C}_{V_i} - p_i), \tag{10}$$

where K is a (potentially time-varying) uniformly positive definite control gain matrix, which may have a skew-symmetric component to encourage other tasks, for example

exploration [20]. The parameters $\hat{a}_i$ used to calculate $\hat{C}_{V_i}$ are adjusted according to an adaptation law which is introduced below.

Define two quantities,

$$\Lambda_i = \int_0^t w(\tau)\mathcal{K}_i(\tau)\mathcal{K}_i(\tau)^T\, d\tau, \qquad (11)$$
$$\text{and} \quad \lambda_i = \int_0^t w(\tau)\mathcal{K}_i(\tau)\phi_i(\tau)\, d\tau.$$

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. The function $w(t) \in \mathcal{L}^1$, where $w(t) \geq 0$, determines a data collection weighting which will be discussed further in section III-A.

Also define

$$F_i = \frac{\int_{V_i} \mathcal{K}(q)(q-p_i)^T\, dq K \int_{V_i}(q-p_i)\mathcal{K}(q)^T\, dq}{\int_{V_i} \hat{\phi}_i(q)\, dq}, \qquad (12)$$

which can also be computed by robot i as it does not require any knowledge of a.

We use the quantities above to compute the parameter adaptation law

$$\dot{\hat{a}}_{\text{pre}_i} = -F_i\hat{a}_i - \gamma\,(\Lambda_i\hat{a}_i - \lambda_i) - \zeta \sum_{j\in\mathcal{N}_i}(\hat{a}_i - \hat{a}_j), \qquad (13)$$

where $\gamma, \zeta \in \mathbb{R}_+$ are scalar gains. Each term in (13) has an intuitive interpretation. The first term compensates for uncertainty in the centroid position estimate. The second term carries out a gradient descent to minimize the sensory function error $\tilde{\phi}_i(p_i)$ integrated over time. The last term is the consensus coupling term, which causes all of the robots' parameters to reach a common value. We stress that a distributed implementation requires that each robot adapts its own parameter vector using local information available to it. If one were interested, instead, in designing a *centralized* adaptation law, one could simply use a common parameter vector that is adapted using the information from all robots. We also point out that (13) will have discontinuities when robots join or leave the set of Voronoi neighbors of robot i, and our convergence analysis takes this into account.

Equation (13) is the main adaptation law, however the controller (10) has a singularity at $\hat{a}_i = 0$ (since $\hat{M}_{V_i}$ is in the denominator of $\hat{C}_{V_i}$). For this reason we prevent the parameters from dropping below $a_{\min} > 0$ using a projection law [21]

$$\dot{\hat{a}}_i = \Gamma(\dot{\hat{a}}_{\text{pre}_i} - I_{\text{proj}_i}\dot{\hat{a}}_{\text{pre}_i}), \qquad (14)$$

where $\Gamma \in \mathbb{R}^{m\times m}$ is a diagonal, positive definite adaptation gain matrix, and the diagonal matrix I_{proj_i} is defined element-wise as

$$I_{\text{proj}_i}(j) = \begin{cases} 0 & \text{for } \hat{a}_i(j) > a_{\min} \\ 0 & \text{for } \hat{a}_i(j) = a_{\min} \text{ and } \dot{\hat{a}}_{\text{pre}_i}(j) \geq 0 \\ 1 & \text{otherwise,} \end{cases} \qquad (15)$$

where (j) denotes the j^{th} element for a vector and the j^{th} diagonal element for a matrix.

The controller described above with $\zeta = 0$ will be referred to as the basic controller, and with $\zeta > 0$ as the consensus controller. The performance of the basic controller was described previously in [1]. The behavior of the consensus controller is formalized in the following theorem.

Theorem 1 (Convergence Theorem): Under Assumption 1, for the system of agents with dynamics (9) and the control law (10),

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

(ii) $\lim_{t\to\infty} \mathcal{K}(p_i(\tau))^T\tilde{a}_i(t) = 0 \quad \forall\, \tau \mid w(\tau) > 0$ and $\forall\, i \in \{1,\ldots,n\}$,

(iii) $\lim_{t\to\infty} \|\hat{a}_i - \hat{a}_j\| = 0 \quad \forall\, i,j \in \{1,\ldots,n\}$.

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 a consequence of Barbalat's lemma.

Let

$$\mathcal{V} = \mathcal{H} + \sum_{i=1}^n \frac{1}{2}\tilde{a}_i^T\Gamma^{-1}\tilde{a}_i. \qquad (16)$$

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

$$\dot{\mathcal{V}} = -\sum_{i=1}^n \Big[\hat{M}_{V_i}(\hat{C}_{V_i} - p_i)^T K(\hat{C}_{V_i} - p_i) + \gamma\int_0^t w(\tau)(\mathcal{K}_i^T(\tau)\tilde{a}_i(t))^2\, d\tau + \tilde{a}_i^T I_{\text{proj}_i}\dot{\hat{a}}_{\text{pre}_i}\Big] - \sum_{i=1}^n \tilde{a}_i^T\zeta\sum_{j\in\mathcal{N}_i}(\hat{a}_i - \hat{a}_j). \qquad (17)$$

In [1] we showed that all the terms within the large brackets are non-negative. Please refer to that work for details. Now consider the last term, which comes from the parameter consensus coupling. We can rewrite this term using the graph Laplacian defined in Section II-C as

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

where $\alpha_j = a(j)\mathbf{1}$, $\hat{\alpha}_j = [\hat{a}_1(j) \;\cdots\; \hat{a}_n(j)]^T$, and $\tilde{\alpha}_j = \hat{\alpha}_j - \alpha_j$. We simply have regrouped the parameters by introducing the α_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 II-C for all $t \geq 0$. From Section II-C we saw that $\alpha_j^T L(t) = a(j)\mathbf{1}^T L = 0$. This gives

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

since $L(t) \geq 0\ \forall t \geq 0$. Thus all terms in (17) are non-positive, and we have $\dot{\mathcal{V}} \leq 0$.

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 $\lim_{t\to\infty}\dot{\mathcal{V}}=0$. This implies (i) and (ii) from Theorem 1 (again, please refer to [1] for details).

Finally, recall that $x^TLx=0$ imples $x=0$ or $x=c\mathbf{1}$. Therefore $\lim_{t\to\infty}\hat{\alpha}_j^TL(t)\hat{\alpha}_j=0$ and $\hat{a}_i(j)\geq a_{\min}$ imply that $\lim_{t\to\infty}\hat{\alpha}_j=a_{\text{final}}(j)\mathbf{1}\ \forall j\in\{1,\ldots,m\}$, 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. This proves (iii) from Theorem 1. ∎

Remark 1: Theorem 1 (i) 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 sufficient richness condition must be satisfied.

Remark 2: Theorem 1 (ii) states that the sensory function estimate $\hat{\phi}_i$ will converge to the true sensory function ϕ for all points on the robot's trajectory with positive weighting $w(\tau)$. This does not, however, imply that $\hat{\phi}_i(q)\to\phi(q)$ $\forall q\in Q$. Again, this would require an extra sufficient richness condition as discussed in section IV.

Remark 3: Introducing parameter coupling greatly increases parameter convergence rates and makes the controller equations better conditioned for numerical integration, as will be discussed in Section V.

Remark 4: There are many other ways to couple the parameters among neighbors to cause consensus. For example the coupling strength can be a smoothly decaying function of distance to the neighbors [16] (which, among other things, avoids discrete jumps in the adaptation equation (13)), the coupling function can be nonlinear in the parameters with some contraction property [12], and the communication between neighbors can incorporate asymmetries, time delays, and significantly relaxed connectivity requirements [14].

A. Weighting Functions

The form of the function $w(\cdot)$ can be designed to encourage parameter convergence. One intuitive option is $w(\tau)=\|\dot{p}_i\|^2$. This weighting normalizes the effects of the rate of travel so that all new data is incorporated with equal weighting. Also, when the robot comes to a stop, the value of $\phi(p_i)$ at the stopped position does not overwhelm the learning law. However, there is an analytical technicality: to ensure that Λ_i and λ_i remain bounded we have to prove that $\dot{p}_i\in\mathcal{L}^2$. To get around this, we can set $w(\tau)=0$ after some fixed time.

We can also set $w(t,\tau)=\exp\{-(t-\tau)\}$, which turns the integrators Λ_i and λ_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 Convergence Analysis

As a separate matter from the convergence behavior in Theorem 1, one may wonder *how quickly* parameters converge to their final values, and under what conditions the final parameter values are the true parameters. In this section we show that parameter convergence is not exponential, though under some conditions it can be shown to converge exponentially to an arbitrarily small error. The rate of this convergence is shown to be faster for the controller with parameter consensus than for the basic controller. We neglect the projection operation, as the discrete switching considerably complicates the convergence analysis.

From (13) and (14), neglecting the projection, we have

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

To analyze parameter convergence, 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 $F=\text{diag}_{i=1}^n(\Gamma F_i)$, $\Lambda=\text{diag}_{i=1}^n(\Gamma\Lambda_i)$, 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 eigenvalues of $\mathcal{L}$ are the same as those of ΓL, but each eigenvalue has multiplicity m. As for a typical graph Laplacian, $\mathcal{L}$ is positive semi-definite.

The coupled dynamics of the parameters over the whole network can be written

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

with $\hat{A}$ defined in the obvious way. This leads to

$$\frac{d}{dt}\|\tilde{A}\|=-\frac{\tilde{A}^T(\gamma\Lambda+\zeta\mathcal{L})\tilde{A}}{\|\tilde{A}\|}-\frac{\tilde{A}^TF\hat{A}}{\|\tilde{A}\|},$$

Let $\lambda_{\min}(t)\geq 0$ be the minimum eigenvalue of $\gamma\Lambda(t)+\zeta\mathcal{L}(t)$. Then we have

$$\frac{d}{dt}\|\tilde{A}\|\leq-\lambda_{\min}(t)\|\tilde{A}\|+\|F\hat{A}\|. \tag{18}$$

Now consider the signal $\|F\hat{A}\|$. We proved in Theorem 1 that $\|\hat{C}_{V_i}-p_i\|\to 0$ and all other quantities in $F_i\hat{a}_i$ are bounded for all i, therefore $\|F\hat{A}\|\to 0$. Also, $\lambda_{\min}(0)=0$, and $\lambda_{\min}(t)$ is a nondecreasing function of time.

Suppose at some time T the robot network has a sufficiently rich trajectory (so that $\gamma\Lambda(T)+\zeta\mathcal{L}(T)$ is positive definite), then $\lambda_{\min}(t)>\lambda_{\min}(T)>0\ \forall t\geq T$. Then from (18), $\|\tilde{A}\|$ will behave like an exponentially stable first order system driven by $\|F\hat{A}\|$. Finally, the gains Γ, γ, and ζ can be set so that $\|F\hat{A}\|$ is arbitrarily small compared to $\gamma\Lambda+\zeta\mathcal{L}$ without affecting stability. Thus, if the robot network's trajectory is sufficiently rich, exponentially fast convergence to an arbitrarily small parameter error can be achieved.

To compare with the performance of the basic controller ($\zeta=0$) consider that $\gamma\Lambda\leq\gamma\Lambda+\zeta\mathcal{L}$. Therefore the minimum eigenvalue for the consensus controller is always at least as large as that for the basic controller implying convergence is at least as fast. In practice, as we will see in the next section, parameter convergence is orders of magnitude faster for the consensus controller.

V. Numerical Simulations

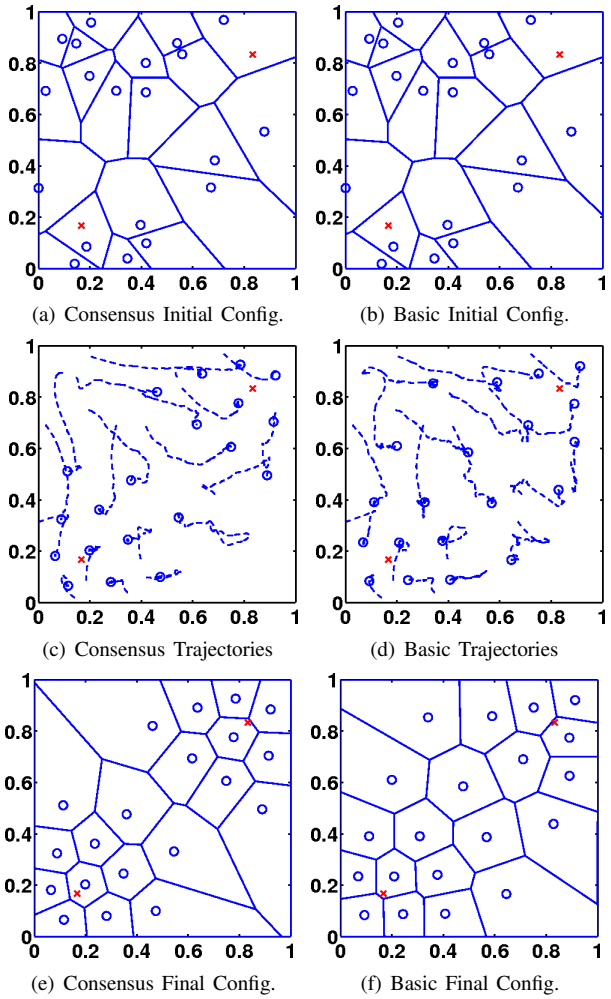
Simulations were carried out in a Matlab environment. The dynamics in (9) with the control law in (10), and the adaptation laws in (14) and (11) for a group of $n = 20$ robots were modeled as a system of coupled differential equations. The fixed-time-step numerical solver was used to integrate the equations 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

$$\mathcal{K}(j) = \frac{1}{\sigma_j\sqrt{2\pi}} \exp\left\{-\frac{(q-\mu_j)^2}{2\sigma_j^2}\right\}, \qquad (19)$$

where $\sigma_j = .18$. The unit square was divided into an even 3×3 grid and each μ_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 Λ_i and λ_i were each started at zero. The gains used by the robots were $K = 3I_2$, $\Gamma = I_{10}$, $\gamma = 1000$ for the basic controller, and $\gamma = 100$ and $\zeta = 5$ for the consensus controller. In practice, we find that choosing Γ small and γ comparatively large (putting more weight on the second term, which is responsible for integrating measurements of $\phi(p_i)$ into the parameters) improves performance. The spatial integrals in (2) and (13) were approximated by discretizing each Voronoi region V_i into a 7×7 grid and summing contributions of the integrand over the grid. Voronoi regions were computed using a decentralized algorithm similar to the one in [2].

Figure 4 shows the positions of the robots in the network over the course of a simulation run for the parameter 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.

The left of Fig. 5 shows that both controllers achieved a near optimal configuration—one in which the estimated error converges to zero, in accordance with (i) from 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 right of Fig. 5. The basic controller did not reach a true centroidal Voronoi configuration. Furthermore, convergence was so much faster for the consensus controller that we have to use a logarithmic time scale to display both curves on the same

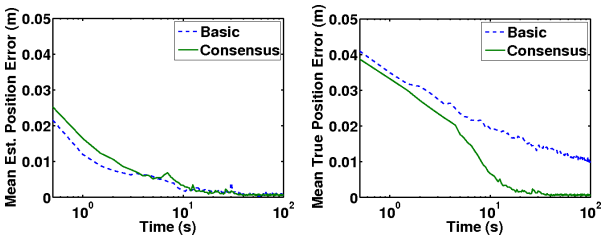


(a) Consensus Initial Config. (b) Basic Initial Config.
(c) Consensus Trajectories (d) Basic Trajectories
(e) Consensus Final Config. (f) Basic Final Config.

Fig. 4. Simulation results for the parameter consensus controller are shown in the left column (4(a), 4(c), and 4(e)), and for the basic controller in the right column (4(b), 4(d), and 4(f)). The Gaussian centers of $\phi(q)$ are marked by the red x's.

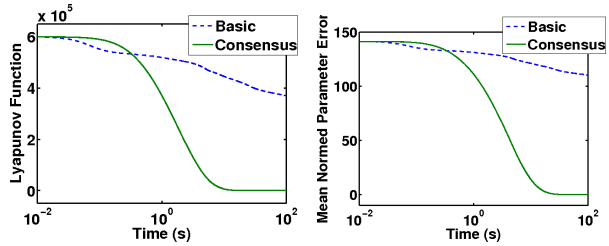
plot. Again, the somewhat jagged time history is a result of the discretized spatial integral computation over the Voronoi region.

Fig. 6(a) 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. Figure 6(b) shows the normed parameter error $\|\tilde{a}_i\|$ averaged over all of the robots. The parameter errors for the consensus controller all converge to zero, indicating that, in fact, the robots trajectories were sufficiently rich. This was also evidenced in Fig. 5(b). For the basic controller, on the other hand, the parameters did not converge to the true parameters. Finally, the quantity $\sum_{i=1}^{n} \tilde{a}_i^T \sum_{j\in\mathcal{N}_i}(\hat{a}_i - \hat{a}_j)$ representing the disagreement among the parameter values of robots is shown in Fig. 6(c). 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.

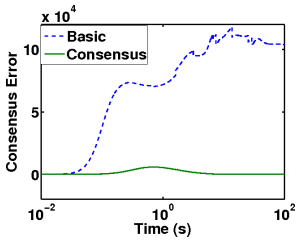


(a) Mean Estimated Position Error (b) Mean True Position Error

Fig. 5. 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, 5(b). However, in accordance with Theorem 1, the estimated error converges to zero in both cases, 5(a). Note the logarithmic time scale.



(a) Lyapunov Function (b) Mean Normed Parameter Error



(c) Consensus Error

Fig. 6. The value of the Lyapunov function is shown for both the basic and parameter consensus controllers on the top left, and the normed parameter error $\|\tilde{a}_i\|$ averaged over all robots is on the top right. The parameter error converges to zero for the consensus controller indicating that the robot trajectories were sufficiently rich. The bottom plot shows a quantity representing a measure of the disagreement of parameters among robots. The disagreement converges to zero for the consensus controller, as asserted in Theorem 1, but does not converge for the basic controller. Note the logarithmic time scale.

VI. Conclusion

In this work we introduced parameter coupling into an existing decentralized adaptive control law to drive a network of 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 parameter consensus among the robots. 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 fundamental and practical tools for enabling distributed learning, and have 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] 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.

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

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

[9] 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.

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

[11] D. P. Bertsekas and J. N. Tsitsiklis, "Comments on "coordination of groups of mobile autonomous agents using nearest neighbor rules"," *IEEE Transactions on Automatic Control*, vol. 52, no. 5, pp. 968–969, 2007.

[12] 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.

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

[14] 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.

[15] 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.

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

[17] R. Sanner and J. Slotine, "Gaussian networks for direct adaptive control," *IEEE Transactions on Neural Networks*, vol. 3, no. 6, 1992.

[18] M. Cannon and J. Slotine, "Space-frequency localized basis function networks for nonlinear estimation and control," *Neurocomputing*, vol. 9, no. 3, 1995.

[19] C. Godsil and G. Royle, *Algebraic Graph Theory*. New York: Springer, 2004.

[20] M. Schwager, F. Bullo, D. Skelly, and D. Rus, "A ladybug exploration strategy for distributed adaptive coverage control," in *Proceedings of International Conference on Robotics an Automation*, Pasadena, CA, May 2008.

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