Decentralized, Adaptive Coverage Control for Networked Robots

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

November, 2007. Revised July, 2008.

Abstract

A decentralized, adaptive control law is presented to drive a network of mobile robots to an optimal sensing configuration. The control law is adaptive in that it uses sensor measurements to learn the distribution of sensory information in the environment. It is decentralized in that it requires only information local to each robot. The controller is then improved upon by introducing a consensus algorithm to propagate sensory information from every robot throughout the network. Convergence and consensus of parameters is proven with a Lyapunov-type proof. The controller with and without consensus is demonstrated in numerical simulations. These techniques are suggestive of broader applications of adaptive control methodologies to decentralized control problems in unknown dynamic environments.

1 Introduction

We are interested in robot group control strategies that are fully decentralized over the group, adaptive to changes in the environment and the group, provably convergent, and practically feasible. This paper describes such a control strategy.

We present a decentralized controller that causes a network of robots to spread out over an environment while aggregating in areas of high sensory interest. Furthermore, the robots do not know beforehand where are the areas of sensory interest, but they learn this information on-line from sensor measurements. The control strategy can be thought of as proceeding simultaneously in two spaces. In the space of robot positions, the robots move to minimize a cost function representing the collective sensing cost of the network, similarly to the control law in [Cortés et al., 2004]. At the same time, in a high-dimensional parameter space, each robot adapts a parameter vector to learn[1] the distribution of sensory information in the environment. The robots eventually reach a near-optimal configuration, and if their paths are sufficiently rich, they reach an optimal configuration. An overview of the control strategy is shown in Figure 1.

Our controller can be used by groups of robots to carry out tasks such as environmental monitoring and clean-up, automatic surveillance of rooms, buildings, or towns, or search and rescue. For example, consider a team of water-borne robots charged with cleaning up an oil spill. Our controller allows the robots to distribute themselves over the spill, learn the areas where the spill is most severe and concentrate their efforts on those areas, without neglecting the areas where the spill is not as severe. Similarly, a group of aerial robots can be deployed with our controller to monitor human activity over a town. Using our controller, they can learn the areas where most human activity takes place (the market, for instance), and concentrate their coverage on those areas, while still providing coverage to less active areas. This behavior can be used to deliver adaptive wireless connectivity or mobile phone coverage to the people in the town. Any application in which a group of automated mobile agents is required to provide collective sensing over an environment can benefit from the proposed control law.

*Mac Schwager and Daniela Rus are with the Computer Science and Artificial Intelligence Lab, MIT, Cambridge, MA 02139, Email: schwager@mit.edu, rus@csail.mit.edu.

†Jean-Jacques Slotine is with the Nonlinear Systems Lab, MIT, Cambridge, MA 02139, Email: jjs@mit.edu.

[1]We will use the words learning and adaptation interchangeably. Learning and adaptation are specifically meant in the sense of parameter tuning, as in adaptive control, rather than the broader meaning often used in Biology and Bio-inspired applications.

The most important feature of our controller is that the robots learn a representation of the sensing environment in closed-loop, in a provably stable way. We first describe a learning law in which each robot uses only its own sensor measurements. We then include a consensus term in the learning law to couple the learning 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. All robots eventually learn the same function incorporating all the sensor measurements collected by all the robots.

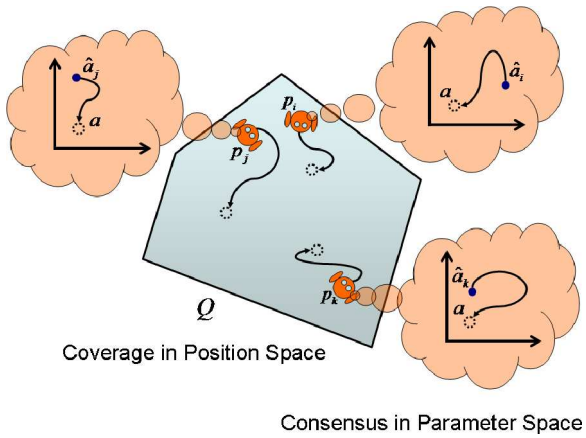


Figure 1: An overview of the decentralized control scheme is shown. The robots, at positions p_i, p_j, and p_k, spread out over the environment, Q, to reach optimal final positions. Simultaneously, each robot adapts a parameter vector ($\hat{a}_i$, $\hat{a}_j$, and $\hat{a}_k$) to build an approximation of the sensory environment. The parameter vectors for neighboring robots are coupled in such a way that their final value, a, is the same for all robots in the network.

1.1 Relation to Previous Work

We use the notion of an optimal sensing configuration developed in [Cortés et al., 2004]. This work itself adapted concepts from locational optimization [Weber, 1929, Drezner, 1995], which is the study of optimally placing industrial facilities. This, in turn, derives from a classical problem of finding geometric median points, which has been attributed to Fermat. Similar frameworks have been used for muli-robot problems in a stochastic setting, for example [Arsie and Frazzoli, 2007], however the problem we consider is a deterministic one. There are also a number of other notions of mult-robot sensory coverage (e.g. [Choset, 2001, Latimer IV et al., 2002, Butler and Rus, 2004, Ögren et al., 2004]), but we choose to adopt the locational optimization approach for its interesting possibilities for analysis and its compatibility with existing ideas in adaptive control [Narendra and Annaswamy, 1989, Sastry and Bodson, 1989, Slotine and Li, 1991]. Our emphasis in this paper is on incorporating learning to enable optimal coverage of an unfamiliar environment. [Arsie and Frazzoli, 2007] This is in contrast to [Cortés et al., 2004] and other papers that use the same optimization framework (e.g. [Salapaka et al., 2003, Cortés et al., 2005, Pimenta et al., 2008a]) in which the distribution of sensory information in the environment is required to be known *a priori* by all robots. This *a priori* requirement was first relaxed in [Schwager et al., 2006] by introducing a controller with a simple memoryless approximation from sensor measurements. The controller was demonstrated in hardware experiments, though a stability proof was not found. In the present work we remove this *a priori* requirement by introducing an adaptive controller inspired by the architecture in [Sanner and Slotine, 1992]. The results in this paper elaborate and improve upon our previous works [Schwager et al., 2007, Schwager et al., 2008b].

It is found that when each robot uses only its own sensor measurements to learn the distribution of sensory information, learning performance can be sluggish. We address this problem by including a consensus

term[2] 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 [Tsitsiklis, 1984, Tsitsiklis et al., 1986, Bertsekas and Tsitsiklis, 1989, Bertsekas and Tsitsiklis, 2007] in discrete time, and more recently in the controls community in continuous time [Jadbabaie et al., 2003, Olfati-Saber and Murray, 2004, Wang and Slotine, 2004, Blondel et al., 2005, Wang and Slotine, 2006, Cucker and Smale, 2007]. In the present work, consensus is used to learn the distribution of sensory information in the environment in a decentralized way by propagating sensor measurements of each robot around the network. This is similar to distributed filtering techniques that have recently been introduced, for example in [Lynch et al., 2008, Zhang and Leonard, 2008], though in contrast to those works, we are concerned with maintaining provable stability of the combined learning and control system. Consensus improves the quality and speed of learning, which in turn causes the robots to converge more quickly to their optimal positions.

In short, the main contributions of this work are to (1) provide a controller that uses parameter adaptation to accomplish coverage without *a priori* knowledge of the sensory environment, and (2) incorporate a consensus term within the parameter adaptation law to propagate sensory information among the robots in the network. Using a Lyapunov-like proof, we show that the control law causes the network to converge to a near-optimal sensing configuration (Theorems 1 and 2), and if the robots' paths are sufficiently rich, the network will converge to an optimal configuration (Corollaries 1 and 2). The dynamics and the environment are assumed to exist in a deterministic setting throughout this work, as is typical in adaptive control.

We set up the problem, provide some background on the results of locational optimization, and state the main assumptions and definitions in Section 2. We present the basic controller and prove convergence to a near-optimal coverage configuration in Section 3. Section 4 introduces the parameter consensus controller and Section 5 discusses and compares parameter convergence rates for the basic and consensus controllers. Alternative stable adaptation laws are discussed in Section 6. Section 7 describes the results of numerical simulations, and conclusions are given in Section 8. We prove two lemmas in Appendix A and give tables of mathematical symbols in Appendix B.

2 Problem Set-up

Let there be n robots in a bounded, convex environment $Q \subset \mathbb{R}^N$. An arbitrary point in Q is denoted q, the position of the i^{th} robot is denoted $p_i \in Q$. 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, $\|\cdot\|$ is used to denote the ℓ^2-norm). The robots are assumed to have some physical extent, therefore no two robots can be at the same position at the same time, $p_i(t) \neq p_j(t)\ \forall i \neq j$.

Define the sensory function to be a continuous function $\phi : Q \mapsto \mathbb{R}_{>0}$ (where $\mathbb{R}_{>0}$ is the set of strictly positive real numbers). The sensory function should be thought of as a weighting of importance over Q. We want to have many robots where $\phi(q)$ is large, and few where it is small. The function $\phi(q)$ is *not known* by the robots in the network, but the robots have sensors that can measure $\phi(p_i)$ at the robot's position p_i. The precise definition of the sensory function depends on the desired application. In an application in which a team of robots are used to clean up an oil spill, an appropriate choice for the sensory function would be the concentration of the oil as a function of position in the environment. For a human surveillance application in which robots use audio sensors, $\phi(q)$ may be chosen to be the intensity of the frequency range corresponding to the human voice.

Finally, let the *unreliability* of the robot's sensors 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.

[2]The phenomenon of decentralized consensus is known by many names including flocking, herding, swarming, agreement, rendezvous, gossip algorithms, and oscillator synchronization. All of these are, at root, the same phenomenon—convergence of the states of a group of dynamical systems to a common final vector (or manifold) through local coupling.

2.1 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 [Cortés et al., 2004, Schwager et al., 2006].

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

$$\mathcal{H}(p_1, \ldots, p_n) = \sum_{i=1}^{n} \int_{V_i} \frac{1}{2} \|q - p_i\|^2 \phi(q)\, dq. \tag{1}$$

Notice that unreliable sensing is expensive and high values of the sensory function, $\phi(q)$, are also expensive. Qualitatively, a low value of $\mathcal{H}$ corresponds to a good configuration for sensory coverage of the environment Q.

Next we define three properties analogous to mass-moments of rigid bodies. The mass, first moment, and centroid of a Voronoi region V_i are defined as

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

respectively. Note that $\phi(q)$ strictly positive imply both $M_{V_i} > 0$ and $C_{V_i} \in V_i \backslash \partial V_i$ (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. The moments and the Voronoi regions themselves depend on the robot positions. Remarkably, despite this dependency, a standard result from locational optimization [Drezner, 1995] 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 critical points 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. It is also known that all such points are local minima. This brings us to the concept of optimal coverage summarized in the following definition.

Definition 1 (Optimal Coverage Configuration) *A robot network is said to be in a (locally) optimal coverage configuration if every robot is positioned at the centroid of its Voronoi region, $p_i = C_{V_i}\ \forall i$.*

We restrict ourselves to looking at local minima of $\mathcal{H}$ in this paper. Global optimization of (1) is known to be difficult (NP-hard for a given discrete representation of $\phi(q)$) even in the centralized case. Thus when we refer to an optimal coverage configuration we mean a locally optimal one. Variations on the control law which attempt to find global minima through exploration are discussed in [Salapaka et al., 2003, Schwager et al., 2008a].

2.2 Assumptions and Definitions

In this section, we describe some of our implicit assumptions that are common in either mulit-robot control, or adaptive control. We comment on the restrictiveness of these assumptions before formally stating one main assumption and several definitions which will be leveraged in our analysis.

Firstly, we assume that the robots have integrator dynamics

$$\dot{p}_i = u_i, \tag{4}$$

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 (4). Higher order dynamics can be accommodated, but the resulting complication obscures the main result.

We assume the robots also are able to compute their own Voronoi cell, $V_i = \{q \mid \|q - p_i\| \leq \|q - p_j\|\}$. This is common in the literature [Salapaka et al., 2003, Cortés et al., 2004, Pimenta et al., 2008a], 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 conservative requirement for each robot to have a communication range as large as the diameter 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 [Cortés et al., 2005]. Indeed, the results of [Cortés et al., 2005] suggest that performance degrades gracefully with decreasing communication range among robots.

Also, as mentioned previously, we assume the sensory function $\phi(q)$ is static for the purposes of our analysis. The principle of timescale separation commonly used in control system analyses suggest that our results will apply to a $\phi(q)$ that varies slowly compared to the dynamics of the robots and the controller. Explicitly taking into account a time changing $\phi(q)$ results in significant analytical complications and is a topic of ongoing research. An analysis for a time-varying sensory function without adaptation is given in [Pimenta et al., 2008b].

We consider relaxing the above assumptions to be important future work, but it is tangential to the main contribution of this work. More central to this work, we make one important assumption regarding the sensory function $\phi(q)$. We use a basis function approximation scheme for each robot to represent the sensory function $\phi(q)$. Let $\mathcal{K} : Q \mapsto \mathbb{R}^m_{>0}$ be a vector of bounded, continuous basis functions (or features). Each robot has these functions available for computation. The sensory function approximation for robot i is given by $\hat{\phi}_i(q,t) = \mathcal{K}(q)^T \hat{a}_i(t)$, where $\hat{a}_i(t)$ is a parameter vector that is tuned according to an adaptation law which we will describe in Section 3. Figure 2 shows a graphical representation of this function approximation scheme. For our analysis, we require that the following assumption holds.

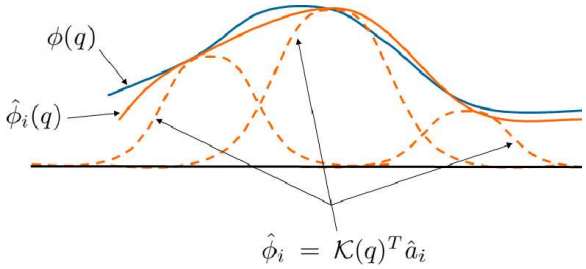


Figure 2: The sensory function approximation is illustrated in this simplified 2-D schematic. The true sensory function is represented by $\phi(q)$ and robot i's approximation of the sensory function is $\hat{\phi}_i(q)$. The basis function vector $\mathcal{K}(q)$ is shown as three Gaussians (dashed curves), and the parameter vector $\hat{a}_i$ denotes the weighting of each Gaussian.

Assumption 1 (Matching Conditions) *There exists and ideal parameter vector $a \in \mathbb{R}^m$ such that*

$$\phi(q) = \mathcal{K}(q)^T a, \tag{5}$$

and a is unknown to the robots. Furthermore,

$$a \geq \mathbf{1} a_{\min} \tag{6}$$

where $a_{\min} \in \mathbb{R}_{>0}$ is a lower bound known by each robot.

We mean the symbols $>, <, \geq$ and $\leq$ to apply element-wise for vectors, and $\mathbf{1}$ is the vector of 1's. 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 some set of basis functions [Sanner and Slotine, 1992]. In practice, however, designing a suitable set of basis functions requires application-specific expertise. We will comment further on this assumption in Section 3, Remark 1.

There is a variety of basis function families to chose from for $\mathcal{K}(q)$. We use Gaussians in our simulations, but other options include wavelets, sigmoids, and splines. Gaussian basis functions have a computational

advantage over non-local basis functions because, in any discrete representation, they have 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.

Next, let the moment approximations using $\hat{\phi}_i(q,t)$ be given by

$$\hat{M}_{V_i}(t) = \int_{V_i} \hat{\phi}_i(q,t)\, dq, \quad \hat{L}_{V_i}(t) = \int_{V_i} q\hat{\phi}_i(q,t)\, dq, \quad \text{and} \quad \hat{C}_{V_i}(t) = \hat{L}_{V_i}(t)/\hat{M}_{V_i}(t), \tag{7}$$

and, the parameter error by

$$\tilde{a}_i(t) = \hat{a}_i(t) - a, \tag{8}$$

and notice the relation

$$\hat{\phi}_i(q,t) - \phi(q) = \mathcal{K}(q)^T \tilde{a}_i(t). \tag{9}$$

In order to compress the notation, we introduce the shorthand $\mathcal{K}_i(t) = \mathcal{K}(p_i(t))$ for the value of the basis function vector at the position of robot i, and $\phi_i(t) = \phi(p_i(t))$ for the value of the sensory function at the position of robot i. As previously stated, robot i can measure ϕ_i with its sensors. We will also commonly refer to quantities without explicitly writing their arguments. However, we may include arguments in some instances to avoid ambiguity.

The function approximation framework described above brings us to another concept of optimality for coverage.

Definition 2 (Near-Optimal Coverage Configuration) *A robot network is said to be in a near-optimal coverage configuration if each robot is positioned at the estimated centroid of its Voronoi region, $p_i = \hat{C}_{V_i}\ \forall i$.*

Finally, we distinguish between two qualities of function approximations.

Definition 3 (Globally True Approximation) *A robot is said to have a globally true (or just true) approximation of the sensory function if its approximation is equal to the actual sensory function at every point of its domain, $\hat{\phi}_i(q) = \phi(q)\ \forall q \in Q$.*

Definition 4 (Locally True Approximation) *A robot is said to have a locally true approximation of the sensory function over a subset $\Omega \subset Q$ if its approximation is equal to the true function at every point in the subset, $\hat{\phi}_i(q) = \phi(q)\ \forall q \in \Omega$.*

In light of the above definitions, if the parameter error is zero, $\tilde{a}_i = 0$, then robot i has a *true* approximation of the sensory function. Also, if $\tilde{a}_i = 0\ \forall i$, then a *near-optimal* coverage configuration is also *optimal*. An overview of the geometrical objects involved in our set-up is shown in Figure 3.

3 Decentralized Adaptive Control Law

We want a controller to drive the robots to an *optimal* configuration, that is, we want to position them at their Voronoi centroids. We emphasize that it is not easy to position a robot at its Voronoi centroid because (1) the robot does not know the sensory function $\phi(q)$ which is required to calculate its centroid, and (2) the centroid moves as a nonlinear function of the robot's position. To overcome the first problem, our controller learns an approximation of the centroid on-line. To overcome the second problem, our controller causes each robot to pursue its estimated centroid. We will prove that the robots achieve a *near-optimal* configuration, and that every robot learns a *locally true* approximation of the sensory function (Theorem 1). Furthermore, if a robot's path is sufficiently rich, it achieves a *globally true* approximation, and if every robots' path is sufficiently rich, the robots reach an *optimal* configuration (Corollary 1).

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 exploration as in [Schwager et al., 2008a]. The environment Q

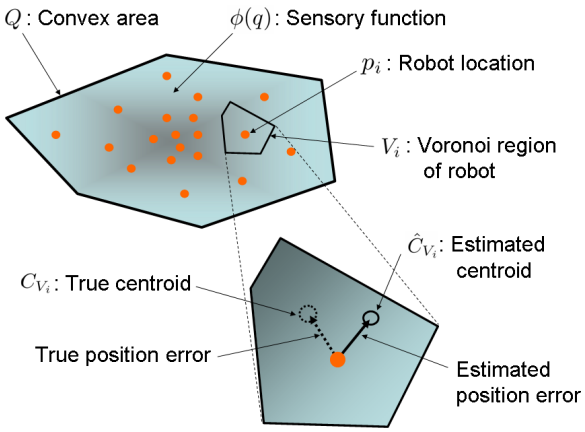


Figure 3: A graphical overview of the quantities involved in the controller is shown. The robots move to cover a bounded, convex environment Q their positions are p_i, and they each have a Voronoi region V_i with a true centroid C_{V_i} and an estimated centroid $\hat{C}_{V_i}$. The true centroid is 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 $\phi(q)$.

is required to be convex so that the control law is feasible, that is, the robots never attempt to cross the boundaries of Q. Since $\hat{C}_{V_i} \in V_i \subset Q$ and $p_i \in Q$, by convexity, the segment connecting the two is in Q. Since the robots have integrator dynamics (4), they will stay within the union of these segments over time, $p_i(t) \in \cup_{\tau>0}(C_{V_i}(\tau) - p_i(\tau))$, and therefore remain in the environment Q.

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. First, we define two quantities,

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

These 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 $w(t) \geq 0$ determines a data collection weighting. We require that it is integrable (belongs to L^1), and continuous (belongs to C^0). Define another quantity

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

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 the true parameter vector, a. The adaptation law for $\hat{a}_i$ is now defined as

$$\dot{\hat{a}}_{\text{pre}_i} = -F_i\hat{a}_i - \gamma(\Lambda_i\hat{a}_i - \lambda_i), \tag{13}$$

where $\gamma \in \mathbb{R}_{>0}$ is a learning rate gain. The two terms in (13) have 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 gradient descent interpretation is explored more in Section 6. We stress that a decentralized 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.

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 parameter projection [Slotine and Coetsee, 1986]

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

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

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

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

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

Theorem 1 (Basic Convergence) *Under Assumption 1, the network of robots with dynamics (4), control law (10), and adaptation law (13,14) converges to a* near-optimal *coverage configuration. Furthermore, each robot converges to a* locally true *approximation of the sensory function over the set* $\Omega_i = \{p_i(\tau) \mid \tau \geq 0, w(\tau) > 0\}$*, made up of all points on the robot's trajectory with positive weighting.*

Proof

We will define a lower-bounded, Lyapunov-like function and show that its time derivative is non-increasing. This will imply that it reaches a limit. Furthermore, the time derivative is uniformly continuous, so by Barbalat's lemma[3] [Barbălat, 1959, Popov, 1973] it approaches zero. The quantities $\|\hat{C}_{V_i}(t) - p_i(t)\|$ and $w(\tau)\tilde{\phi}_i(p_i(\tau), t)^2$, $0 \geq \tau \geq t$, will be included in the time derivative of this function, thereby implying $p_i(t) \to \hat{C}_{V_i}(t)$, and $\hat{\phi}_i(q,t) \to \phi(q)\ \forall q \in \Omega_i$ for all i.

Define a Lyapunov-like function

$$\mathcal{V} = \mathcal{H} + \sum_{i=1}^{n} \frac{1}{2}\tilde{a}_i^T \Gamma^{-1} \tilde{a}_i, \tag{16}$$

which incorporates the sensing cost $\mathcal{H}$, and is quadratic in the parameter errors $\tilde{a}_i$. Note that the sensing cost $\mathcal{H}$ is computed with the *actual* sensory function $\phi(q)$, so it inherently incorporates function approximation errors as well. $\mathcal{V}$ is bounded below by zero since $\mathcal{H}$ is a sum of integrals of strictly positive functions, and the quadratic parameter error terms are each bounded below by zero.

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

$$\dot{\mathcal{V}} = \sum_{i=1}^{n} \left[\frac{\partial \mathcal{H}}{\partial p_i}^T \dot{p}_i + \tilde{a}_i^T \Gamma^{-1} \dot{\tilde{a}}_i \right],$$

and substituting from (3) and noticing that $\dot{\tilde{a}}_i = \dot{\hat{a}}_i$ yields

$$\dot{\mathcal{V}} = \sum_{i=1}^{n} \left[-\int_{V_i} (q - p_i)^T \phi(q)\, dq \dot{p}_i + \tilde{a}_i^T \Gamma^{-1} \dot{\hat{a}}_i \right].$$

Using (9) to substitute for $\phi(q)$ gives

$$\dot{\mathcal{V}} = \sum_{i=1}^{n} \left[-\int_{V_i} (q - p_i)^T \hat{\phi}_i\, dq \dot{p}_i + \int_{V_i} \tilde{a}_i^T \mathcal{K}(q)(q - p_i)^T\, dq \dot{p}_i + \tilde{a}_i^T \Gamma^{-1} \dot{\hat{a}}_i \right].$$

Substituting for $\dot{p}_i$ with (4) and (10), and moving $\tilde{a}_i$ out of the second integral (since it is not a function of q) leads to

$$\dot{\mathcal{V}} = \sum_{i=1}^{n} \left[-\hat{M}_{V_i}(\hat{C}_{V_i} - p_i)^T K(\hat{C}_{V_i} - p_i) + \tilde{a}_i^T \int_{V_i} \mathcal{K}(q)(q - p_i)^T\, dq (\hat{C}_{V_i} - p_i) + \tilde{a}_i^T \Gamma^{-1} \dot{\hat{a}}_i \right].$$

[3] We cannot use the more typical LaSalle invariance theorem because our system is time-varying due to the data weighting function w(t).

Expanding $(\hat{C}_{V_i} - p_i)$ in the second term, and substituting for $\dot{\hat{a}}_i$ with (14) gives

$$\dot{\mathcal{V}} = \sum_{i=1}^{n} \left[-\hat{M}_{V_i} (\hat{C}_{V_i} - p_i)^T K (\hat{C}_{V_i} - p_i) + \tilde{a}_i^T F_i \hat{a}_i - \tilde{a}_i^T F_i \hat{a}_i - \tilde{a}_i^T \gamma (\Lambda_i \hat{a}_i - \lambda_i) - \tilde{a}_i^T I_{\text{proj}_i} \dot{\hat{a}}_{\text{pre}_i} \right].$$

Now we can expand $(\Lambda_i \hat{a}_i - \lambda_i)$, noting that $\lambda_i = \int_0^t w(\tau) \mathcal{K}_i (\mathcal{K}_i^T a)\, d\tau$, to get

$$\dot{\mathcal{V}} = -\sum_{i=1}^{n} \left[\hat{M}_{V_i} (\hat{C}_{V_i} - p_i)^T K (\hat{C}_{V_i} - p_i) + \tilde{a}_i^T \gamma \int_0^t w(\tau) \mathcal{K}_i \mathcal{K}_i^T \tilde{a}_i(t)\, d\tau + \tilde{a}_i^T I_{\text{proj}_i} \dot{\hat{a}}_{\text{pre}_i} \right],$$

and, finally, bringing $\tilde{a}_i^T$ inside the integral (it is not a function of τ, though it is a function of t) results in

$$\dot{\mathcal{V}} = -\sum_{i=1}^{n} \left[\hat{M}_{V_i} (\hat{C}_{V_i} - p_i)^T K (\hat{C}_{V_i} - p_i) + \gamma \int_0^t w(\tau) (\mathcal{K}_i(\tau)^T \tilde{a}_i(t))^2\, d\tau + \tilde{a}_i^T I_{\text{proj}_i} \dot{\hat{a}}_{\text{pre}_i} \right], \tag{17}$$

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

$$\tilde{a}_i(j) I_{\text{proj}_i}(j) \dot{\hat{a}}_{\text{pre}_i}(j). \tag{18}$$

From (15), 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 1). 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 (18) is non-negative, and all three terms inside the sum in (17) are non-negative. Thus $\dot{\mathcal{V}} \leq 0$.

We have that $\mathcal{V}$ is lower bounded and $\dot{\mathcal{V}} \leq 0$, so $\mathcal{V}$ approaches a limit. We establish the uniform continuity of $\dot{\mathcal{V}}$ in Lemma 1 in Appendix A, so by Barbalat's lemma $\lim_{t\to\infty} \dot{\mathcal{V}} = 0$. From (17), this implies $\lim_{t\to\infty} \|p_i(t) - \hat{C}_{V_i}(t)\| = 0\ \forall i$ from the first term in the sum, so the network converges to a *near-optimal* coverage configuration.

Furthermore, from $\mathcal{K}_i(\tau)^T \tilde{a}_i(t) = \hat{\phi}_i(p_i(\tau), t) - \phi(p_i(\tau))$, we have from the second term of(17)

$$\lim_{t\to\infty} \int_0^t w(\tau) (\hat{\phi}_i(p_i(\tau), t) - \phi(p_i(\tau)))^2\, d\tau = 0 \quad \forall i = 1, \ldots, n. \tag{19}$$

Now notice that the integrand in (19) is non-negative, therefore it must converge to zero for all τ except on a set of Lesbegue measure zero. Suppose the integrand is greater than zero at some point τ. The integrand is continuous (since $\mathcal{K}_i(t)$, $\hat{a}_i(t)$, and $\phi_i(t)$ are), so if it is greater than zero at τ, it is greater than zero in a neighborhood of non-zero measure around it, $(\tau - \epsilon, \tau + \epsilon)$, for some $\epsilon > 0$, which is a contradiction. Thus, we have $\hat{\phi}_i(q, t) \to \phi(q)\ \forall q \in \Omega_i$ and $\forall i$. □

In [Schwager et al., 2008a] the following extension to the above theorem was derived. We restate it here to give a more thorough characterization the controller's behavior.

Corollary 1 (Sufficient Richness for Basic Controller) *In addition to the conditions for Theorem 1, if the robots' paths are such that the matrix* $\lim_{t\to\infty} \Lambda_i(t)$ *is positive definite* $\forall i$*, the network converges to an* optimal *coverage configuration, and each robot converges to a* globally true *approximation of the sensory function,* $\phi(q)$*.*

Proof

Consider the second term in (17). Move the two $\tilde{a}_i(t)$ outside of the integral (since they are not a function of τ) to get

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

Since $\dot{\mathcal{V}} \to 0$, if $\lim_{t\to\infty} \Lambda_i(t)$ is positive definite (we know the limit exists because $\mathcal{K}(q)$ is bounded and $w(t) \in L^1$), then $\tilde{a}_i(t) \to 0$. This implies that robot i converges to a *globally true* approximation of the sensory function, $\phi(q)$. Furthermore, if $\lim_{t\to\infty} \Lambda_i(t) > 0 \ \forall i$, then $\hat{C}_{V_i} = C_{V_i} \ \forall i$, so the network converges to an *optimal* coverage configuration. □

Remark 1 *One may wonder how the controller will behave if Assumption 1 fails, so that there is no ideal parameter vector a that will* exactly *reconstruct $\phi(q)$ from the basis functions. Indeed, this will be the case in any real-world scenario. Such a question requires a robustness analysis that is beyond the scope of this paper, but analyses of robustness for centralized adaptive controllers can be found, for example, in [Sanner and Slotine, 1992] and most texts on adaptive control (e.g. [Narendra and Annaswamy, 1989, Sastry and Bodson, 1989, Slotine and Li, 1991]). It is observed in numerical simulations that the adaptation law finds a parameter to make $\hat{\phi}_i(q)$ as close as possible to $\phi(q)$, where closeness is measured by the integral of the squared difference, as described in Section 6.*

Remark 2 *One may also wonder how the controller behaves with time varying sensory functions $\phi(q,t)$. It can be expected from existing results for centralized adaptive controllers, that our controller will track sensory functions that change slowly with respect to the rate of adaptation of the parameters. The ability to track a time varying sensory function can be enhanced by using a forgetting factor in the data weighting function $w(t)$ as described in Section 6.3.*

Remark 3 *As an alternative to our controller, one may propose a two-part strategy in which robots explore until they have learned the sensory function well enough, and then move to cover the environment. This seems likely to work, but how would one implement it in a distributed fashion? Specifically, do all robots switch to coverage mode simultaneously? If so, how do they agree on a time? If not, how does the system behave when some robots are in explore mode and some in coverage mode? Our strategy essentially combines the exploration and coverage into one continuous controller avoiding these problems. We again highlight the fact that the control gain K can be designed with a time-varying component, as in [Schwager et al., 2008a], to encourage exploration and improve learning.*

4 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 (3) is modified so that the adaptation laws among Voronoi neighbors are coupled with a weighting proportional to the length of their shared Voronoi edge. Adaptation and consensus were also combined in [Ögren et al., 2004] and [Wang and Slotine, 2006], however in those works consensus was used to align the velocities of agents, not to help in the parameter adaptation process itself. Our use of consensus is more related to the recent algorithms for distributed filtering described in [Lynch et al., 2008] and [Zhang and Leonard, 2008].

4.1 Graph Laplacians

An undirected graph $G = (V, E)$ is defined by a set of indexed vertices $V = \{v_1, \ldots, v_n\}$ and a set of edges $E = \{e_1, \ldots, e_{n_E}\}$, $e_i = \{v_j, v_k\}$. 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. Consider a function $l : V \times V \mapsto \mathbb{R}$ such that $l(v_i, v_j) = 0 \ \forall (v_i, v_j) \notin E$ and $l(v_i, v_j) > 0 \ \forall (v_i, v_j) \in E$. We call $l(v_i, v_j)$ a weighting over the graph G, and for shorthand we write $l_{ij} = l(v_i, v_j)$. Next consider the weighted graph Laplacian matrix L, whose terms are given by

$$L(i,j) = \begin{cases} -l_{ij} & \text{for } i \neq j \\ \sum_{j=1}^{n} l_{ij} & \text{for } i = j. \end{cases}$$

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 [Godsil and Royle, 2004] that for a connected graph, the weighted graph Laplacian is positive semi-definite, $L \geq 0$, and it has exactly one zero eigenvalue, with the associated eigenvector $\mathbf{1} = [1, \ldots, 1]^T$. In particular, $L\mathbf{1} = \mathbf{1}^T L = 0$, and $x^T L x > 0$, $\forall x \neq c\mathbf{1}, c \in \mathbb{R}$. These properties will be important in what follows.

4.2 Consensus Learning Law

We add a term to the parameter adaptation law in (13) to couple the adaptation of parameters between neighboring agents. Let the new adaptation law be given by

$$\dot{\hat{a}}_{\mathrm{pre}_i} = -F_i \hat{a}_i - \gamma \left(\Lambda_i \hat{a}_i - \lambda_i\right) - \zeta \sum_{j=1}^{n} l_{ij}(\hat{a}_i - \hat{a}_j), \tag{20}$$

where l_{ij} is a weighting over the Delaunay communication graph between two robots i and j and $\zeta \in \mathbb{R}_{>0}$, is a positive gain. The projection remains the same as in (14), namely

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

A number of different weightings l_{ij} are conceivable, but here we propose that l_{ij} be equal to the length (area for $N = 3$, or volume for $N > 3$) of the shared Voronoi edge of robots i and j,

$$l_{ij} = |V_i \cap V_j|. \tag{21}$$

Notice that $l_{ij} \geq 0$ and $l_{ij} = 0$ if and only if i and j are not Voronoi neighbors, so l_{ij} is a valid weighting over the Delaunay communication graph as described in Section 4.1. This weighting is natural since one would want a robot to be influenced by its neighbor in proportion to its neighbor's proximity. This form of l_{ij} will also provide for a simple analysis since it maintains the continuity of the right hand side of (20), which is required for using Barbalat's lemma.

Theorem 2 (Convergence with Parameter Consensus) *Under the conditions of Theorem 1, using the parameter adaptation law (20), the network of robots converge to a* near-optimal *coverage configuration. Furthermore, each robot converges to a* locally true *approximation of the sensory function over the set all points on every robot's trajectory with positive weighting,* $\Omega = \cup_{j=1}^{n} \Omega_j$. *Additionally,*

$$\lim_{t \to \infty} (\hat{a}_i - \hat{a}_j) = 0 \quad \forall i, j \in \{1, \ldots, n\}. \tag{22}$$

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 (16), which leads to

$$\dot{\mathcal{V}} = -\sum_{i=1}^{n} \left[\hat{M}_{V_i} (\hat{C}_{V_i} - p_i)^T K (\hat{C}_{V_i} - p_i) + \gamma \int_0^t w(\tau) (\mathcal{K}_i(\tau)^T \tilde{a}_i(t))^2 \, d\tau + \tilde{a}_i^T I_{\mathrm{proj}_i} \dot{\hat{a}}_{\mathrm{pre}_i} \right] - \sum_{i=1}^{n} \tilde{a}_i^T \zeta \sum_{j=1}^{n} l_{ij}(\hat{a}_i - \hat{a}_j). \tag{23}$$

We have already shown that the three terms inside the first sum are non-negative. Now consider the parameter coupling term. We can rewrite this term using the graph Laplacian defined in Section 4.1 as

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

where $\alpha_j = a(j)\mathbf{1}$, $\hat{\alpha}_j = [\hat{a}_1(j) \quad \cdots \quad \hat{a}_n(j)]^T$, and $\tilde{\alpha}_j = \hat{\alpha}_j - \alpha_j$. Recall the ideal parameter vector $a = [a(1) \quad \cdots \quad a(j) \quad \cdots \quad a(m)]^T$, and the parameter estimate for each agent $\hat{a}_i = [\hat{a}_i(1) \quad \cdots \quad \hat{a}_i(j) \quad \cdots \quad \hat{a}_i(m)]^T$. We have simply regrouped the parameters by introducing the α_j notation. From Section 4.1 we saw that $\alpha_j^T L = a(j)\mathbf{1}^T L = 0$. 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 $L \geq 0$. Thus $\dot{\mathcal{V}} \leq 0$.

Lemma 2 establishes the uniform continuity of $\dot{\mathcal{V}}$ for this controller. We can therefore use Barbalat's lemma to conclude that $\dot{\mathcal{V}} \to 0$. As before this implies the two claims of Theorem 1. Since the graph Laplacian is positive semi-definite, and $\hat{a}_i(j) \geq a_{\min}$, $\lim_{t\to\infty} \hat{\alpha}_j^T L \hat{\alpha}_j = 0 \Rightarrow \lim_{t\to\infty} \hat{\alpha}_j = a_{\text{final}}(j)\mathbf{1}$ $\forall j \in \{1, \ldots, m\}$, 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. The consensus assertion (22) follows.

Finally, recall the fact that for robot j, $\hat{\phi}_j(q) \to \phi(q)$ over Ω_j, but $\hat{a}_i \to \hat{a}_j$, therefore $\hat{\phi}_i(q) \to \phi(q)$ over Ω_j. This is true for all robots i and j, therefore $\hat{\phi}_i \to \phi(q)$ over $\Omega = \cup_{j=1}^{n} \Omega_j$ for all i. □

Corollary 2 (Sufficient Richness for Consensus Controller) *In addition to the conditions for Theorem 2, if the robots' paths are such that $\int_\Omega \mathcal{K}(q)\mathcal{K}(q)^T dq$ is positive definite, the network converges to an* optimal *coverage configuration, and each robot converges to a* globally true *approximation of the sensory function, $\phi(q)$.*

Proof

Since $\hat{\phi}_i(q,t) \to \phi(q)$ over Ω, we have $\tilde{a}_i(\infty)^T \mathcal{K}(q)\mathcal{K}(q)^T \tilde{a}_i(\infty) = 0$ over Ω, where $\tilde{a}_i(\infty)$ is shorthand for $\lim_{t\to\infty} \tilde{a}_i(t)$. Then

$$0 = \int_\Omega \tilde{a}_i(\infty)^T \mathcal{K}(q)\mathcal{K}(q)^T \tilde{a}_i(\infty) dq = \tilde{a}_i(\infty)^T \int_\Omega \mathcal{K}(q)\mathcal{K}(q)^T dq\, \tilde{a}_i(\infty)$$

Therefore if $\int_\Omega \mathcal{K}(q)\mathcal{K}(q)^T dq > 0$, then $\tilde{a}_i(\infty) = 0$. This is true for all i □

Remark 4 *The condition of Corollary 2 is less strict than that of Corollary 1 because only the union of all the robots' paths has to be sufficiently rich, not each path individually. This means it is easier to achieve an* optimal *configuration with the consensus controller.*

Remark 5 *Another commonly used weighting for algorithms over communication graphs is*

$$l_{ij} = \begin{cases} 1 & \text{for } j \in \mathcal{N}_i \\ 0 & \text{for } j \notin \mathcal{N}_i, \end{cases}$$

where $\mathcal{N}_i$ is the set of indices of neighbors of i, as was proposed in [Schwager et al., 2008b]. In this case, stability can be proved, but with considerable complication in the analysis, since $\dot{\mathcal{V}}$ is not continuous. Even so, recent extensions of Barbalat's lemma to differential inclusions from [Ryan, 1998, Logemann and Ryan, 2004] (and applied to flocking systems in [Tanner et al., 2007]) can be used to prove the same result as in Theorem 2.

Remark 6 *Introducing parameter coupling increases parameter convergence rates and makes the controller equations better conditioned for numerical integration, as will be discussed in Section 7. However there is a cost in increased communication overhead. In a discrete-time implementation of the controller in which parameters and robot positions are represented finitely with b bits, a robot will have to transmit $(m+2)b$ bits and receive $|\mathcal{N}_i|(m+2)b$ bits per time step. While for the basic controller, each robot must transmit $2b$*

and receive $2|\mathcal{N}_i|b$ bits per time step. This may or may not represent a significant communication overhead, depending upon b and the speed of the control loop. In hardware experiments we have found this to be a negligible communication cost. Note that although discretization is necessary for a practical implementation, it does not affect the essential phenomenon of consensus, as shown in [Kashyap et al., 2007, Frasca et al., 2008].

5 Parameter Convergence Analysis

As a separate matter from the asymptotic convergence in Theorem 1 and Theorem 2, one may wonder *how quickly* parameters converge to their final values. In this section we show that parameter convergence is not exponential, though given sufficiently rich trajectories 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 non-smooth switching considerably complicates the convergence analysis.

From (13) and (14), neglecting the projection, but including the adaptation gain matrix Γ, we have

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

which can be written as

$$\dot{\tilde{a}}_i = -\Gamma\gamma\Lambda_i(t)\tilde{a}_i - \Gamma F_i\hat{a}_i, \tag{24}$$

leading to

$$\frac{d}{dt}\|\tilde{a}_i\| = -\frac{\gamma\tilde{a}_i^T\Gamma\Lambda_i(t)\tilde{a}_i}{\|\tilde{a}_i\|} - \frac{\tilde{a}_i^T\Gamma F_i\hat{a}_i}{\|\tilde{a}_i\|}. \tag{25}$$

Let $\lambda_{\min_i}(t) \geq 0$ be the minimum eigenvalue of $\Gamma\Lambda_i(t)$ (we know it is real-valued and non-negative since $\Lambda_i(t)$ is symmetric positive semi-definite). Then we have

$$\frac{d}{dt}\|\tilde{a}_i\| \leq -\gamma\lambda_{\min_i}(t)\|\tilde{a}_i\| + \|\Gamma F_i\hat{a}_i\|. \tag{26}$$

Now consider the signal $\|\Gamma F_i\hat{a}_i\|$. We proved in Theorem 1 that $\|\hat{C}_{V_i} - p_i\| \to 0$ and all other quantities in $\Gamma F_i\hat{a}_i$ are bounded for all i, therefore $\|\Gamma F_i\hat{a}_i\| \to 0$. Also, $\lambda_{\min_i}(0) = 0$, and $\lambda_{\min_i}(t)$ is a nondecreasing function of time. Suppose at some time T, robot i has a sufficiently rich trajectory (so that $\Lambda_i(T)$ is positive definite, as in Corollary 1), then $\lambda_{\min_i}(t) > \lambda_{\min_i}(T) > 0\ \forall t \geq T$. Then from (26), $\|\tilde{a}_i\|$ will decay faster than an exponentially stable first order system driven by $\|\Gamma F_i\hat{a}_i\|$. Finally, the gains Γ and γ can be set so that $\|\Gamma F_i\hat{a}_i\|$ is arbitrarily small compared to $\gamma\lambda_{\min_i}$ without affecting stability. Thus, if the robot's trajectory is sufficiently rich, exponentially fast convergence to an arbitrarily small parameter error can be achieved.

Now we consider a similar rate analysis for the controller with parameter consensus. In this case, because the parameters are coupled among robots, we must consider the evolution of all the robots' parameters together. Let

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

be a concatenated vector consisting of all the robots' parameter errors. 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 network can be written

$$\dot{\tilde{A}} = -(\gamma\Lambda + \zeta\mathcal{L})\tilde{A} - F\hat{A}, \tag{27}$$

with $\hat{A}$ defined in the obvious way. Notice the similarity in form between (24) and (27). Following the same type of derivation as before we find

$$\frac{d}{dt}\|\tilde{A}\| \leq -\lambda_{\min}(t)\|\tilde{A}\| + \|F\hat{A}\|, \tag{28}$$

where $\lambda_{\min}(t) \geq 0$ is the minimum eigenvalue of $\gamma\Lambda(t) + \zeta\mathcal{L}(t)$. Again, it is real-valued and non-negative since $\gamma\Lambda(t) + \zeta\mathcal{L}(t)$ is symmetric positive semi-definite.

As before, the signal $\|F\hat{A}\| \to 0$. If after some time T, $\text{mineig}(\Lambda(T)) > 0$ then $\lambda_{\min}(t) \geq \text{mineig}(\Lambda(t)) > 0$ $\forall t \geq T$ and the network's trajectory is sufficiently rich. Then from (25), $\|\tilde{A}\|$ will decay at least as fast as 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 $\gamma\Lambda(t) + \zeta\mathcal{L}(t)$ 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 for the whole network.

To compare with the performance of the basic controller consider that $\gamma\Lambda(t) \leq \gamma\Lambda(t) + \zeta\mathcal{L}(t)$. 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 Section 7, parameter convergence is orders of magnitude faster for the consensus controller.

6 Alternative Learning Laws

The adaptation law for parameter tuning (13) can be written more generally as

$$\dot{\hat{a}}_i = -F_i\hat{a}_i + f_i(p_i, V_i, \hat{a}_i, t), \tag{29}$$

where we have dropped the projection operation for clarity. There is considerable freedom in choosing the learning function $f_i(\cdot)$. We are constrained only by our ability to find a suitable Lyapunov-like function to accommodate Barbalat's lemma.

6.1 Gradient Laws

The form of $f_i(\cdot)$ chosen in Section 3 can be called a gradient law, since

$$f_i = -\frac{\partial}{\partial\hat{a}_i}\left[\frac{1}{2}\gamma\int_0^t w(\tau)(\hat{\phi}_i - \phi_i)^2\,d\tau\right]. \tag{30}$$

The parameter vector follows the negative gradient of the Least Squares cost function, seeking a minimum.

Another possible learning law is to follow the gradient, given by

$$\begin{aligned} f_i &= -\frac{\partial}{\partial\hat{a}_i}\left[\frac{1}{2}\gamma w(\tau)(\hat{\phi}_i - \phi_i)^2\right] \\ &= -\gamma w(t)\mathcal{K}_i(\mathcal{K}_i^T\hat{a}_i - \phi_i). \end{aligned} \tag{31}$$

Using the same Lyapunov function as before, it can be verified that this learning law results in a *near-optimal* coverage configuration.

These two gradient laws can be combined to give

$$f_i = -\gamma\left[w(t)\mathcal{K}_i(\mathcal{K}_i^T\hat{a}_i - \phi_i) + (\Lambda_i\hat{a}_i - \lambda_i)\right], \tag{32}$$

which is, in fact, equivalent to the first law with a weighting function $w_c(t,\tau) = \delta(t-\tau)w(t) + w(\tau)$, where $\delta(t-\tau)$ is the delta-Dirac function (we can make $w(\cdot)$ a function of t, and τ with minimal consequences to the convergence proof). The same Lyapunov-like function can be used, such that the resulting time derivative is

$$\dot{\mathcal{V}} = -\sum_{i=1}^{n} \Big[\hat{M}_{V_i}(\hat{C}_{V_i} - p_i)^T K(\hat{C}_{V_i} - p_i) + \tilde{a}_i^T I_{\text{proj}_i} \dot{\hat{a}}_{\text{pre}_i} + \gamma \tilde{a}_i^T \left[w(t)\mathcal{K}_i\mathcal{K}_i^T + \Lambda_i \right] \tilde{a}_i \Big],$$

leading to the same convergence claims as in Theorem 1 and Corollary 1.

6.2 Recursive Least Squares Laws

Another interesting possibility for a learning law is the continuous-time Recursive Least Squares method. This law can be interpreted as continuously solving the Least Squares minimization problem recursively as new data is acquired. Let

$$J = \frac{1}{2}\int_0^t w(\tau)(\hat{\phi}_i - \phi_i)^2 \, d\tau$$

be the standard Least Squares cost function with a data weighting function $w(\tau)$. Then, taking the gradient with respect to $\hat{a}_i$ and setting to zero we find

$$\Lambda_i(t)\hat{a}_i = \lambda_i(t).$$

If the matrix $\Lambda_i(t)$ is full rank, we can pre-multiply both sides by its inverse to solve the Least Squares problem. However, we seek a recursive expression, so taking the time derivative we obtain

$$\dot{\hat{a}}_i = -P_i(t)w(t)\mathcal{K}_i(\mathcal{K}_i^T\hat{a}_i - \phi_i), \text{ where } P_i(t) = \Lambda_i(t)^{-1}.$$

Using an identity from vector calculus, P_i can be computed differentially by $\dot{P}_i = -P_i w(t)\mathcal{K}_i\mathcal{K}_i^T P_i$, but the initial conditions are ill defined. Instead, we must use some nonzero initial condition, P_{i0}, with the differential equation $\dot{P}_i = -P_i w(t)\mathcal{K}_i\mathcal{K}_i^T P_i$, to give the approximation

$$P_i = \Lambda_i^{-1} + P_{i0}. \tag{33}$$

The initial condition can be interpreted as the inverse covariance of our prior knowledge of the parameter values. We should choose this to be small if we have no idea of the ideal parameter values when setting initial conditions.

Before we can apply the Recursive Least Squares law to our controller, there is one additional complication that must be dealt with. We can no longer use the same projection operator to prevent the singularity when $\hat{M}_{V_i} = 0$. However, it is possible to formulate a different stable controller that eliminates this singularity altogether. This formulation also has the advantage that it no longer requires $a(j) > a_{\min}\ \forall j$ in Assumption 1. We can use the controller

$$u_i = K(\hat{L}_{V_i} - \hat{M}_{V_i}p_i), \tag{34}$$

with the adaptation law

$$\dot{\hat{a}}_i = -P_i\left[\hat{M}_{V_i}F_i\hat{a}_i + w(t)\mathcal{K}_i(\mathcal{K}_i^T\hat{a}_i - \phi_i)\right] \tag{35}$$

to approximate the Recursive Least Squares law. Asymptotic convergence can be proven for this case by using the Lyapunov function

$$\mathcal{V} = \mathcal{H} + \sum_{i=1}^{n}\frac{1}{2}\tilde{a}_i^T P_i^{-1}\tilde{a}_i, \tag{36}$$

which leads to

$$\dot{\mathcal{V}} = -\sum_{i=1}^{n} \left[k\hat{M}_{V_i}^2 (\hat{C}_{V_i} - p_i)^T K (\hat{C}_{V_i} - p_i) + \frac{1}{2}\tilde{a}_i^T \left[w(t)\mathcal{K}_i \mathcal{K}_i^T \right] \tilde{a}_i \right],$$

Note that the only difference in the Lyapunov function is that Γ has been replaced with the time-varying quantity P_i.

We can also formulate a learning law analogous to the combined gradient law (32) as

$$\dot{\hat{a}}_i = -P_i \left(\hat{M}_{V_i} F_i \hat{a}_i + \frac{1}{2} w(t) \mathcal{K}_i (\mathcal{K}_i^T \hat{a}_i - \phi_i) + (\Lambda_i \hat{a}_i - \lambda_i) \right), \tag{37}$$

with Λ_i and λ_i defined as before. The same Lyapunov function can be used (36), resulting in

$$\dot{\mathcal{V}} = -\sum_{i=1}^{n} \left[k\hat{M}_{V_i}^2 (\hat{C}_{V_i} - p_i)^T K (\hat{C}_{V_i} - p_i) + \tilde{a}_i^T \Lambda_i \tilde{a}_i \right].$$

Interestingly, the integral terms (those involving Λ_i and λ_i) of the learning law in (37) have a gradient interpretation. Taking just those terms we have

$$\begin{aligned} f_i &= -P_i(\Lambda_i \hat{a}_i - \lambda_i) \\ &= -\tilde{a}_i + P_{i0}\Lambda_i \tilde{a}_i \\ &= -\frac{\partial}{\partial \hat{a}_i} \left(\frac{1}{2} \tilde{a}_i^T \tilde{a}_i \right) + P_{i0}\Lambda_i \tilde{a}_i, \end{aligned} \tag{38}$$

so the law approximates the gradient of the squared parameter error. The last term on the right hand side arises from the mismatch in initial conditions between P_i and Λ_i.

The combination of Least Squares and gradient learning apparent in this law is quite similar to the Composite Adaptation described in [Slotine and Li, 1989, Slotine and Li, 1991]. In fact, if one identifies the prediction error as $\mathcal{K}_i^T \hat{a}_i - \phi_i$ and the tracking error as $\Lambda_i - \lambda_i \phi_i$ we have composite adaptation (except, of course, for the term containing F_i, which is required for the stability proof).

Unfortunately, it is found that the equations resulting from the Least Squares formulation are difficult to solve numerically, often causing robots to jump outside of the environment Q, which then corrupts the Voronoi calculation. Alleviating this problem is a matter of ongoing research.

6.3 Data 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 $w(\tau) = 1/2(\mathrm{erf}(c-t)+1)$. Many other options exist.

One 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 Λ_i and λ_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 Λ_i, P_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. Forgetting factors can have other significant benefits such as improving parameter convergence rates and allowing the flexibility to reject certain frequencies of noise in the error signal. A thorough discussion of forgetting factors can be found in [Slotine and Li, 1991], Section 8.7.

7 Numerical Simulations

7.1 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 (10) and adaptation law in (14) both require the computation of integrals over V_i, thus robot i must be able to re-calculate its Voronoi region at each time step. Several algorithms exist for computing V_i in a distributed fashion, for example those given in [Cortés et al., 2004, Li and Rus, 2005].

Algorithm 1 Adaptive Coverage Control Algorithm

Require: Each robot can compute its Voronoi region
Require: $\phi(q)$ can be parameterized as in (5)
Require: a is lower bounded as in (6)
  Initialize Λ_i, λ_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 (7)
    Update $\hat{a}_i$ according to (14)
    Update Λ_i and λ_i according to (11)
    Apply control input $u_i = -K(\hat{C}_{V_i} - p_i)$
  end loop

Algorithm 1 is distributed, adaptive, and requires only communication between robots that are Voronoi neighbors. The time complexity of the distributed Voronoi region computation for one robot is linear in the number of robots, n. The time complexity of computing the discretized integral $\hat{C}_{V_i}$ can be found to be linear in the size of the discretization grid, and linear in the number of basis functions, m. Therefore, Algorithm 1 can be used on teams of large robots, on teams of small robots such as [McLurkin, 2004], or on mobile sensor network nodes with limited computation and storage capabilities such as the mobile Mica Motes described by [Sibley et al., 2002].

7.2 Implementation

Simulations were carried out in a Matlab environment. The dynamics in (4) with the control law in (10), and the adaptation laws in (14) (with (13) for the basic controller and (20) for the consensus controller) for a group of $n = 20$ robots were integrated forward in time. A numerical solver with a fixed-time-step of .01s was used to integrate the equations. The environment Q was taken to be the unit square. The sensory function, $\phi(q)$, was parameterized as a linear combination of nine Gaussians. In particular, for $\mathcal{K} = [\ \mathcal{K}(1) \quad \cdots \quad \mathcal{K}(9)\ ]^T$, each component, $\mathcal{K}(j)$, was implemented as

$$\mathcal{K}(j) = \frac{1}{2\pi\sigma_j^2} \exp -\frac{(q-\mu_j)^2}{2\sigma_j^2}, \tag{39}$$

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 nine 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 Gaussians 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 $K = 3I_2$, $\Gamma = I_9$, $\gamma = 300$ and $\zeta = 0$ for the basic controller, and $\gamma = 100$ and $\zeta = 50$ for the consensus controller. In practice, the first integral term in the adaptive law (13) seems to have little effect on the performance of the controller.

Choosing Γ small and γ 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 (7) and (13) required for the control law were computed 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 [Cortés et al., 2004].

7.3 Simulation Results

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.

Figure 5(a) shows that both controllers converge to a *near-optimal* configuration—one in which every robot is located at the estimated centroid of its Voronoi region, in accordance with Theorem 1. However, the true position error also converged to zero for the consensus controller, indicating that it achieved an *optimal* coverage configuration, as shown in Figure 5(b). The basic controller did not reach an *optimal* coverage 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 plot. Again, the somewhat jagged time history is a result of the discretized spatial integral computation over the Voronoi region.

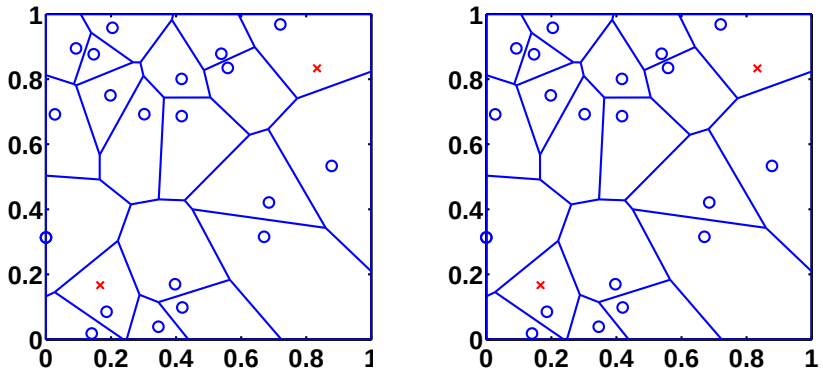
The Figure 6(a) demonstrates that a *locally true* sensory function approximation is achieved for each robot over $\Omega_i = \{p_i(\tau) \mid \tau \geq 0, w(\tau) > 0\}$, the set of points along the robot's trajectory with positive weighting. The plot shows the integral in (19) as a function of time averaged over all the robots in the network converging asymptotically to zero. The disagreement among the parameter values of robots is shown in the right of Figure 6(b). The parameters were initialized to $a_{\min}$ for all robots, so this value starts from zero in both cases. However, the consensus controller causes the parameters to reach consensus, while for the basic controller the parameters do not converge to a common value.

Figure 7(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. In fact, Figure 7(b) shows that the parameter errors $\|\tilde{a}_i\|$ actually converged to zero for the consensus controller, so the conditions for Corollary 2 were met. This was also evidenced in Figure 5(b) since the true position error converged to zero. For the basic controller, on the other hand, the parameters did not converge to the true parameters.

8 Conclusion

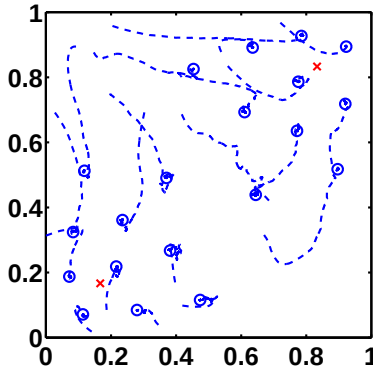
In this work we described a decentralized control law for positioning a network of robots optimally for sensing in their environment. The controller used an adaptive control architecture to learn a parameterized model of the sensory distribution in the environment while seeking an optimal coverage configuration. 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. Parameter coupling was introduced in the adaptation laws to increase parameter convergence rates and cause the robots' parameters to achieve a common final value. The control law was demonstrated in numerical simulations of a group of 20 robots sensing over an environment with a bimodal Gaussian distribution of sensory information.

We expect that the techniques 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 have compelling parallels with distributed learning mechanisms in biological systems. We hope that our approach will yield fruitful combinations of adaptive control and decentralized control to produce engineered agents that can cooperate with one another while gathering information from their environment to proceed toward a common goal.

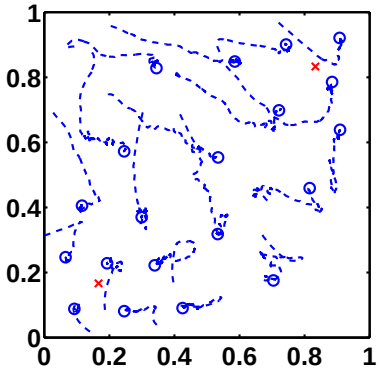


(a) Consensus Initial Config.

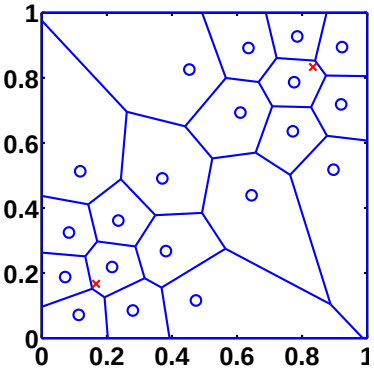
(b) Basic Initial Config.



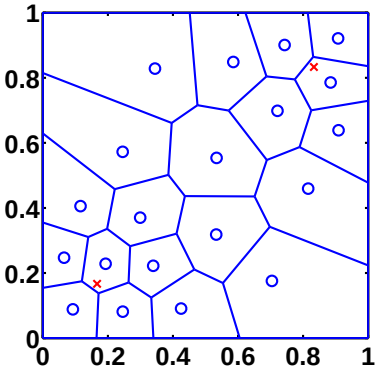
(c) Consensus Trajectories



(d) Basic Trajectories

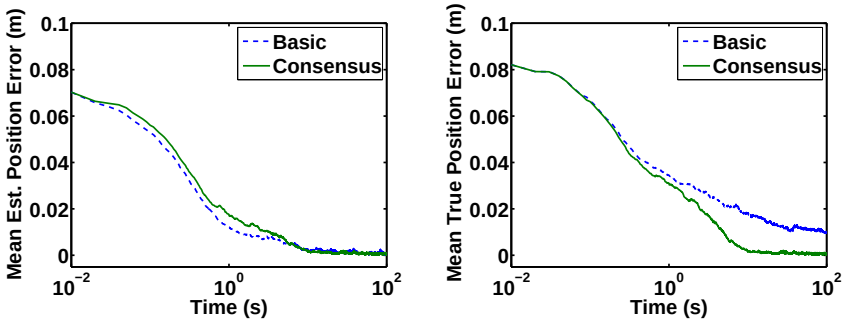


(e) Consensus Final Config.



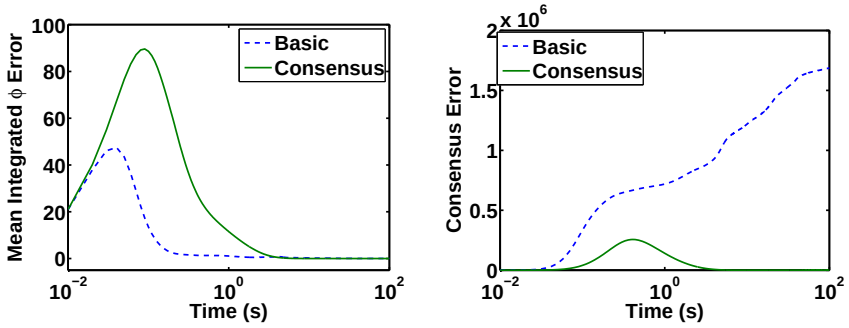
(f) Basic Final Config.

Figure 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 x's.



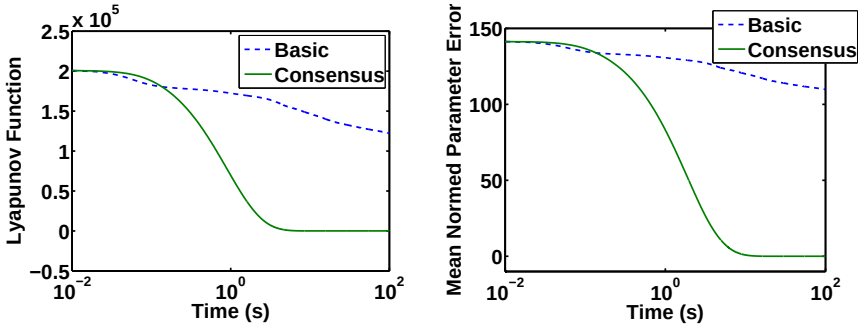
(a) Mean Estimated Position Error (b) Mean True Position Error

Figure 5: The estimated position error, $\|\hat{C}_{V_i} - p_i\|$, and the true position error, $\|C_{V_i} - p_i\|$ averaged over all the robots in the network is shown for the network of 20 robots for both the basic and parameter consensus 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) Mean Int. $\phi(q)$ Error (b) Consensus Error

Figure 6: The integrated sensory function error, namely $\int_0^t w(\tau)(\mathcal{K}_i \tilde{a}_i)^2 \, d\tau$, averaged over all the robots is shown for the basic and consensus controllers in 6(a). The plot demonstrates that each robot converges to a locally true function approximation over all points along its trajectory with positive weighting, $w(\tau) > 0$, as asserted in Theorem 1. The quantity $\sum_{i=1}^{n} \hat{a}_i^T \sum_{j=1}^{n} (\hat{a}_i - \hat{a}_j)$ is shown in 6(b), 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.



(a) Lyapunov Function

(b) Mean $\|\tilde{a}_i(t)\|$

Figure 7: The Lyapunov function is shown in 7(a) for both the basic and parameter consensus controllers. Notice that the parameter consensus controller results in a faster decrease and a lower final value of the function. The normed parameter error $\|\tilde{a}_i\|$ averaged over all robots is shown in 7(b). The parameter error converges to zero with the consensus controller indicating that the robot trajectories were sufficiently rich.

Appendix A

Lemma 1 (Uniform Continuity for Basic Controller) *For the basic controller, $\dot{\mathcal{V}}$ is uniformly continuous.*

Proof

We will bound the time derivatives of a number of quantities. A bounded derivative is sufficient for uniform continuity. Firstly, notice that $\hat{C}_{V_i}, p_i \in V_i \subset Q$, so $\hat{C}_{V_i}$ and p_i are bounded, which implies $\dot{p}_i = K(\hat{C}_{V_i} - p_i)$ is bounded. Consider terms of the form

$$\frac{d}{dt}\left(\int_{V_i} f(q,t)\,dq\right)$$

where $f(q,t)$ is a bounded function with a bounded time derivative $\frac{d}{dt}f(q,t)$. We have

$$\frac{d}{dt}\left(\int_{V_i} f(q,t)\,dq\right) = \int_{V_i} \frac{df(q,t)}{dt}\,dq + \int_{\partial V_i} f(q,t) n_{\partial V_i}^T \sum_{j=1}^{n} \frac{\partial(\partial V_i)}{\partial p_j}\dot{p}_j\,dq, \tag{40}$$

where ∂V_i is the boundary of V_i and $n_{\partial V_i}^T$ is the outward facing normal of the boundary. Now $\frac{\partial(\partial V_i)}{\partial p_j}$ is bounded for all j, $\dot{p}_j$ was already shown to be bounded, and $f(q,t)$ is bounded by assumption, therefore $d/dt(\int_{V_i} f(q,t)\,dq)$ is bounded.

Notice that $\dot{\hat{C}}_{V_i}$ is composed of terms of this form, so it is bounded. Therefore $\ddot{p}_i = K(\dot{\hat{C}}_{V_i} - \dot{p}_i)$ is bounded, and $\dot{p}_i$ is uniformly continuous.

Now consider

$$\dot{\mathcal{V}} = \sum_{i=1}^{n}\left[-\int_{V_i}(q-p_i)^T\phi(q)\,dq\,\dot{p}_i + \tilde{a}_i^T\Gamma^{-1}\dot{\hat{a}}_i\right].$$

The first term inside the sum is uniformly continuous since it is the product of two quantities which were already shown to have bounded time derivatives, namely $\int_{V_i}(q-p_i)^T\phi(q)\,dq$ (an integral of the form (40)) and $\dot{p}_i$. Now consider the second term in the sum. It is continuous in time since $\dot{\hat{a}}_i$ is continuous. Expanding it using (14) and (13) as

$$\tilde{a}_i^T\Gamma^{-1}(I_{\mathrm{proj}_i} - I)(F_i\hat{a}_i + \gamma(\Lambda_i\hat{a}_i - \lambda_i))$$

shows that it is not differentiable where the matrix I_{proj_i} switches. However, the switching condition (15) is such that $\dot{\hat{a}}_i(t)$ is not differentiable only at isolated points on the domain $[0, \infty)$. Also, at all points where it is differentiable, its time derivative is uniformly bounded (since $\dot{\hat{a}}_i$ and the integrands of Λ_i and λ_i are bounded, and F_i is composed of the kind of integral terms of the form (40)). This implies that $\tilde{a}_i^T \Gamma^{-1} \dot{\hat{a}}$ is uniformly continuous. We conclude that $\dot{\mathcal{V}}$ is uniformly continuous. □

Lemma 2 (Uniform Continuity for Consensus Controller) *For the consensus controller, $\dot{\mathcal{V}}$ is uniformly continuous.*

Proof

We have

$$\dot{\mathcal{V}} = \sum_{i=1}^{n} \left[-\int_{V_i} (q - p_i)^T \phi(q)\, dq\, \dot{p}_i + \tilde{a}_i^T \Gamma^{-1} \dot{\hat{a}}_i \right],$$

therefore the reasoning of the proof of Lemma 1 applies as long as $\dot{\hat{a}}_i$ can be shown to be uniformly continuous. But $\dot{\hat{a}}_i$ only differs from the basic controller in the presence of the term

$$\zeta \sum_{j=1}^{n} l_{ij}(\hat{a}_i - \hat{a}_j).$$

The Voronoi edge length, l_{ij}, is a continuous function of p_k, $k \in \{1, \ldots, n\}$. Furthermore, where it is differentiable, it has uniformly bounded derivatives. It was shown in the proof of Lemma 1 that $\dot{p}_k$ is bounded, so similarly to $\dot{\hat{a}}_i$, the points at which $l_{ij}(p_1(t), \ldots, p_n(t))$ is not differentiable are isolated points on $[0, \infty)$. Therefore l_{ij} is uniformly continuous in time. All other terms in $\dot{\hat{a}}_{pre_i}$ were previously shown to be uniformly continuous, so $\dot{\hat{a}}_{pre_i}$ is uniformly continuous. As shown in the proof of Lemma 1, the projection operation preserves uniform continuity, therefore $\dot{\hat{a}}_i$ is uniformly continuous. □

Appendix B

This section contains tables of the symbols used in this work.

Table of symbols of primary importance

Symbol	Description
Q	convex bounded environment which the robots are to cover
q	an arbitrary point in Q
p_i	position of robot i
V_i	Voronoi region of robot i
$\phi(q)$	sensory function
$\phi_i(t)$	the value of the sensory function at a robot position, $\phi(p_i(t))$
$\hat{\phi}_i(q,t)$	robot i's approximation of $\phi(q)$
$\mathcal{K}(q)$	vector of basis functions for the sensory function, $\phi(q) = \mathcal{K}(q)^T a$
$\mathcal{K}_i(t)$	the value of the basis functions at a robot position, $\mathcal{K}(p_i(t))$
a	ideal parameter vector for the sensory function, $\phi(q) = \mathcal{K}(q)^T a$
$a_{\min}$	a positive lower bound on the elements of a
$\hat{a}_i(t)$	robot i's parameters
l_{ij}	weighting between parameters for robots i and j
$\tilde{a}_i(t)$	robot i's parameter error, $\hat{a}_i(t) - a$
M_{V_i}	mass of V_i
$\hat{M}_{V_i}(t)$	approximation of M_{V_i}
L_{V_i}	first mass moment of V_i
$\hat{L}_{V_i}(t)$	approximation of L_{V_i}
C_{V_i}	centroid of V_i
$\hat{C}_{V_i}(t)$	approximation of C_{V_i}
$\mathcal{H}(p_1, \ldots, p_n)$	locational cost function
u_i	control input
Λ_i	weighted integral of basis functions vector over robot trajectory
λ_i	weighted integral of sensory measurements over robot trajectory
$w(t)$	data weighting function
$\mathcal{V}$	Lyapunov function
$\mathcal{N}_i$	neighbor set of robot i
L	graph Laplacian of the robot network

Table of symbols of secondary importance

Symbol	Description
n	number of robots in the network
N	number of dimensions of the space in which the robots exist
m	number of parameters
Ω_i	the set of points in the trajectory of $p_i(t)$ with positive weighting, $w(t) > 0$
Ω	union of all Ω_i
K	positive definite control gain matrix
F_i	term in Lyapunov proof resulting from imperfect sensory approximation
$\dot{\hat{a}}_{\mathrm{pre}_i}$	time derivative of robot i's parameters before projection
γ	adaptive gain for learning law
Γ	diagonal, positive definite adaptation gain matrix
I_{proj_i}	matrix for implementing parameter projection
G	a graph
V	vertex set of a graph
v_i	a vertex in a graph
E	edge set of a graph
e_i	an edge in a graph
n_E	number of edges in a graph
A	adjacency matrix of a graph
c	an arbitrary real constant
ζ	positive consensus gain
α_j	vector containing the jth parameter of each robot
T	some fixed time
λ_{min_i}	the minimum eigenvalue of $\Lambda_i(t)$
$\tilde{A}$	vector containing the parameter errors of all robots
$\hat{A}$	vector containing the parameter estimates of all robots
F	block diagonal matrix with ΓF_i on each block
Λ	block diagonal matrix with $\Gamma\Lambda_i$ on each block
$\mathcal{L}$	generalized graph Laplacian for the network of robots
$\lambda_{\min}$	minimum eigenvalue of $\gamma\Lambda + \zeta\mathcal{L}$
$f_i(p_i, V_i, \hat{a}_i, t)$	a general learning law
$f(q,t)$	a general function of position and time
P_i	an approximation of Λ_i^{-1}
P_{i0}	the initial condition for P_i
σ_j	standard deviation of the jth Gaussian basis function
μ_j	mean of the jth Gaussian basis function
∂V_i	boundary of V_i
$n_{\partial V_i}$	outward facing unit normal vector along ∂V_i
J	the least squares cost function

Acknowledgements

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. We thank the anonymous reviewers for their helpful comments.

References

[Arsie and Frazzoli, 2007] Arsie, A. and Frazzoli, E. (2007). Efficient routing of multiple vehicles with no explicit communications. *International Journal of Robust and Nonlinear Control*, 18(2):154–164.

[Barbălat, 1959] Barbălat, I. (1959). Systèmes d'équations différentielles d'oscillations non linéaires. *Revue de Mathématiques Pures et Appliquées*, 4:267–270.

[Bertsekas and Tsitsiklis, 1989] Bertsekas, D. P. and Tsitsiklis, J. N. (1989). *Parallel and Distributed Computation: Numerical Methods*. Prentice Hall.

[Bertsekas and Tsitsiklis, 2007] Bertsekas, D. P. and Tsitsiklis, J. N. (2007). Comments on "coordination of groups of mobile autonomous agents using nearest neighbor rules". *IEEE Transactions on Automatic Control*, 52(5):968–969.

[Blondel et al., 2005] Blondel, V. D., Hendrickx, J. M., Olshevsky, A., and Tsitsiklis, J. N. (2005). Convergence in multiagent coordination, consensus, and flocking. In *Proceedings of the Joint IEEE Conference on Decision and Control and European Control Conference*, pages 2996–3000, Seville, Spain.

[Butler and Rus, 2004] Butler, Z. J. and Rus, D. (2004). Controlling mobile sensors for monitoring events with coverage constraints. In *Proceedings of the IEEE International Conference of Robotics and Automation*, pages 1563–1573, New Orleans, LA.

[Choset, 2001] Choset, H. (2001). Coverage for robotics—A survey of recent results. *Annals of Mathematics and Artificial Intelligence*, 31:113–126.

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

[Cortés et al., 2004] Cortés, J., Martínez, S., Karatas, T., and Bullo, F. (2004). Coverage control for mobile sensing networks. *IEEE Transactions on Robotics and Automation*, 20(2):243–255.

[Cucker and Smale, 2007] Cucker, F. and Smale, S. (2007). Emergent behavior in flocks. *IEEE Transactions on Automatic Control*, 52(5):852–862.

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

[Frasca et al., 2008] Frasca, P., Carli, R., Fagnani, F., and Zampieri, S. (2008). Average consensus on networks with quantized communication. *Submitted*.

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

[Jadbabaie et al., 2003] Jadbabaie, A., Lin, J., and Morse, A. S. (2003). Coordination of groups of mobile autonomous agents using nearest neighbor rules. *IEEE Transactions on Automatic Control*, 48(6):988–1001.

[Kashyap et al., 2007] Kashyap, A., Başar, T., and Srikant, R. (2007). Quantized consensus. *IEEE Transaction on Automatic Control*, 43(7):1192–1203.

[Latimer IV et al., 2002] Latimer IV, D. T., Srinivasa, S., Shue, V., adnd H. Choset, S. S., and Hurst, A. (2002). Towards sensor based coverage with robot teams. In *Proceedings of the IEEE International Conference on Robotics and Automation*, volume 1, pages 961–967.

[Li and Rus, 2005] Li, Q. and Rus, D. (2005). Navigation protocols in sensor networks. *ACM Transactions on Sensor Networks*, 1(1):3–35.

[Logemann and Ryan, 2004] Logemann, H. and Ryan, E. P. (2004). Asymptotic behaviour of nonlinear systems. *The American Mathematical Monthly*, 111(10):864–889.

[Lynch et al., 2008] Lynch, K. M., Schwartz, I. B., Yang, P., and Freeman, R. A. (2008). Decentralized environmental modeling by mobile sensor networks. *IEEE Transactions on Robotics*, 24(3):710–724.

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

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

[Ögren et al., 2004] Ögren, P., Fiorelli, E., and Leonard, N. E. (2004). Cooperative control of mobile sensor networks: Adaptive gradient climbing in a distributed environment. *IEEE Transactions on Automatic Control*, 49(8):1292–1302.

[Olfati-Saber and Murray, 2004] Olfati-Saber, R. and Murray, R. R. (2004). Consensus problems in networks of agents with switching topology and time-delays. *IEEE Transactions on Automatic Control*, 49(9):1520–1533.

[Pimenta et al., 2008a] Pimenta, L. C. A., Kumar, V., Mesquita, R. C., and Pereira, G. A. S. (2008a). Sensing and coverage for a network of heterogeneous robots. In *Proceedings of the IEEE Conference on Decision and Control*, Cancun, Mexico.

[Pimenta et al., 2008b] Pimenta, L. C. A., Schwager, M., Lindsey, Q., Kumar, V., Rus, D., Mesquita, R. C., and Pereira, G. A. S. (2008b). Simultaneous coverage and tracking (SCAT) of moving targets with robot networks. In *Proceedings of the Eighth International Workshop on the Algorithmic Foundations of Robotics (WAFR)*, Guanajuato, Mexico.

[Popov, 1973] Popov, V. M. (1973). *Hyperstability of Automatic Control Systems*. Springer Verlag, New York.

[Ryan, 1998] Ryan, E. P. (1998). An integral invariance principle for differential inclusions with applications in adaptive control. *SIAM Journal of Control and Optimization*, 36(3):960–980.

[Salapaka et al., 2003] Salapaka, S., Khalak, A., and Dahleh, M. A. (2003). Constraints on locational optimization problems. In *Proceedings of the Conference on Decision and Control*, volume 2, pages 1430–1435, Maui, Hawaii, USA.

[Sanner and Slotine, 1992] Sanner, R. and Slotine, J. (1992). Gaussian networks for direct adaptive control. *IEEE Transactions on Neural Networks*, 3(6):837–863.

[Sastry and Bodson, 1989] Sastry, S. S. and Bodson, M. (1989). *Adaptive control: stability, convergence, and robustness*. Prentice-Hall, Inc., Upper Saddle River, NJ.

[Schwager et al., 2008a] Schwager, M., Bullo, F., Skelly, D., and Rus, D. (2008a). A ladybug exploration strategy for distributed adaptive coverage control. In *Proceedings of the International Conference on Robotics an Automation (ICRA 08)*, pages 2346–2353, Pasadena, CA.

[Schwager et al., 2006] Schwager, M., McLurkin, J., and Rus, D. (2006). Distributed coverage control with sensory feedback for networked robots. In *Proceedings of Robotics: Science and Systems II*, pages 49–56, Philadelphia, PA.

[Schwager et al., 2007] Schwager, M., Slotine, J.-J. E., and Rus, D. (2007). Decentralized, adaptive control for coverage with networked robots. In *Proceedings of the International Conference on Robotics and Automation (ICRA 07)*, pages 3289–3294, Rome.

[Schwager et al., 2008b] Schwager, M., Slotine, J. J. E., and Rus, D. (2008b). Consensus learning for distributed coverage control. In *Proceedings of the International Conference on Robotics and Automation (ICRA 08*, pages 1042–1048, Pasadena, CA.

[Sibley et al., 2002] Sibley, G. T., Rahimi, M. H., and Sukhatme, G. S. (2002). Robomote: A tiny mobile robot platform for large-scale sensor networks. In *Proceedings of the IEEE International Conference on Robotics and Automation*, volume 2, pages 1143–1148, Washington, DC.

[Slotine and Coetsee, 1986] Slotine, J. J. E. and Coetsee, J. A. (1986). Adaptive sliding controller synthesis for nonlinear systems. *International Journal of Control*, 43(6):1631–1651.

[Slotine and Li, 1989] Slotine, J. J. E. and Li, W. (1989). Composite adaptive control of robot manipulators. *Automatica*, 25(4):509–519.

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

[Tanner et al., 2007] Tanner, H. G., Jadbabaie, A., and Pappas, G. J. (2007). Flocking in fixed and switching networks. *IEEE Transactions on Automatic Control*, 52(5):863–868.

[Tsitsiklis, 1984] Tsitsiklis, J. N. (1984). *Problems in Decentralized Decision Making and Computation*. PhD thesis, Department of EECS, MIT.

[Tsitsiklis et al., 1986] Tsitsiklis, J. N., Bertsekas, D. P., and Athans, M. (1986). Distributed asynchronous deterministic and stochastic gradient optimization algorithms. *IEEE Transactions on Automatic Control*, 31(9):803–812.

[Wang and Slotine, 2004] Wang, W. and Slotine, J. J. E. (2004). On partial contraction analysis for coupled nonlinear oscillators. *Biological Cybernetics*, 23(1):38–53.

[Wang and Slotine, 2006] Wang, W. and Slotine, J. J. E. (2006). A theoretical study of different leader roles in networks. *IEEE Transactions on Automatic Control*, 51(7):1156–1161.

[Weber, 1929] Weber, A. (1929). *Theory of the Location of Industries*. The University of Chicago Press, Chicago, IL. Translated by Carl. J. Friedrich.

[Zhang and Leonard, 2008] Zhang, F. and Leonard, N. E. (2008). Cooperative filters and control for cooperative exploration. *IEEE Transactions on Automatic Control*, Submitted.