Robust Adaptive Coverage for Robotic Sensor Networks

Mac Schwager, Michael P. Vitus, Daniela Rus, and Claire J. Tomlin

Abstract This paper presents a distributed control algorithm to drive a group of robots to spread out over an environment and provide adaptive sensor coverage of that environment. The robots use an on-line learning mechanism to approximate the areas in the environment which require more concentrated sensor coverage, while simultaneously exploring the environment before moving to final positions to provide this coverage. More precisely, the robots learn a scalar field, called the weighting function, representing the relative importance of different regions in the environment, and use a Traveling Salesperson based exploration method, followed by a Voronoi-based coverage controller to position themselves for sensing over the environment. The algorithm differs from previous approaches in that provable robustness is emphasized in the representation of the weighting function. It is proved that the robots approximate the weighting function with a known bounded error, and that they converge to locations that are locally optimal for sensing with respect to the approximate weighting function. Simulations using empirically measured light intensity data are presented to illustrate the performance of the method.

Mac Schwager
GRASP Lab, University of Pennsylvania, 3330 Walnut St, PA 19104, USA, and
Computer Science and Artificial Intelligence Lab, MIT, 77 Massachusetts Ave., Cambridge, MA, 02139, USA, e-mail: `schwager@mit.edu`

Michael P. Vitus
Aeronautics and Astronautics, Stanford University, Stanford, CA 94305, USA,
e-mail: `vitus@stanford.edu`

Daniela Rus
Computer Science and Artificial Intelligence Lab, MIT, 77 Massachusetts Ave., Cambridge, MA 02139, USA, e-mail: `rus@csail.mit.edu`

Claire J. Tomlin
Electrical Engineering and Computer Sciences, UC Berkeley, Berkeley, CA 94708, USA,
e-mail: `tomlin@eecs.berkeley.edu`

1 Introduction

In this paper we present a distributed control algorithm to command a group of robots to explore an unknown environment while providing adaptive sensor coverage of interesting areas within the environment. This algorithm has many applications in controlling teams of robots to perform tasks such as search and rescue missions, environmental monitoring, automatic surveillance of rooms, buildings, or towns, or simulating collaborative predatory behavior. As an example application, Japan was hit by a major earthquake on March 11, 2011 that triggered a devastating tsunami causing catastrophic damage to the nuclear reactors at Fukushima. Due to the risk of radiation exposure, humans could not inspect (or repair) the nuclear reactors, however, a team of robots could be used to monitor the changing levels of radiation. Using the proposed algorithm, the robots would concentrate on areas where the radiation was most dangerous, continually providing updated information on how the radiation was evolving. This information could be used to notify people in eminent danger of radiation exposure due to the changing conditions. Similarly, consider a team of waterborne 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.

Sensor coverage algorithms have been receiving a great deal of attention in recent years. Cortés et al. [Cortés et al., 2004] considered the problem of findin an optimal sensing configuration for a group of mobile robots. They used concepts from locational optimization [Weber, 1929, Drezner, 1995] to control the robots based upon gradient descent of a weighting function which encodes the sensing quality and coverage of the environment. This weighting function can be viewed as describing the importance of areas in the environment. The control law for each robot is distributed and only depends on the robot's position and the positions of its neighbors'. However, all robots are required to know the weighting function *a priori* which restricts the algorithm from being deployed in unknown environments. There have been several extensions to this formulation of coverage control. In [Cortés et al., 2005], the robots were assumed to have a limited sensing or communication range. Pimenta et al. [Pimenta et al., 2008a] incorporated heterogeneous robots, and extended the algorithm to handle nonconvex environments. The work [Martínez, 2010] used a distributed interpolation scheme to recursively estimate the weighting function. Similarly, [Schwager et al., 2009] removed the requirement of knowing the weighting function *a priori* by learning a basis function approximation of the weighting function on-line. This strategy has provable convergence properties, but requires that the weighting function lies in a known set of functions. The purpose of the present work is to remove this restriction, greatly broadening the class of weighting functions that can be approximated.

Similar frameworks have been used for multi-robot problems in a stochastic setting [Arsie and Frazzoli, 2007]. There are also a number of other notions of multi-robot sensor coverage (e.g. [Choset, 2001, Latimer IV et al., 2002] and [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].

As noted above, this work extends [Schwager et al., 2009] by removing restrictions on the weighting function, so that a much broader class of weighting functions can be provably approximated. Typically, the form of the weighting function is not known *a priori*, and if this is not accounted for directly then the learning algorithm could chatter between models or even become unstable. Also, in simulations performed with a realistic weighting function, the original algorithm only explores in a local neighborhood of the robots resulting in a poor approximation of the weighting function. However, the algorithm we propose here explores the entire space, successfully learning the weighting function with provable robustness. The robots first partition the environment and perform a Traveling Sales Person (TSP) based distributed exploration, so that the unknown weighting function can be adequately approximated. They then switch, in an asynchronous and distributed fashion, to a coverage mode in which they deploy over the environment to achieve positions that are advantageous for sensing. The robots use an on-line learning mechanism to approximate the weighting function. Since we do not assume the robots can perfectly approximate the weighting function, the parameter adaptation law for learning this function must be carefully constructed to be robust to function approximation errors.

Without specifically designing for such robustness, it is known that many different types of instability [Ioannou and Kokotovic, 1984] can occur. Several techniques have been proposed in the adaptive control literature to handle this kind of robustness, including using a dead-zone [Peterson and Narendra, 1982, Samson, 1983], the σ-modification [Ioannou and Kokotovic, 1984], and the e_1- modification [Narendra and Annaswamy, 1987]. We chose to adapt a dead-zone technique, and prove that the robots learn a function that has bounded difference from the true function, while converging to positions that are locally optimal for sensing with respect to the learned function.

The paper is organized as follows. In Section 2 we introduce notation and formulate the problem. In Section 3 we describe the function approximation strategy and the control algorithm, and we prove the main convergence result of the paper. Section 4 gives the results of a numerical simulation with a weighting function that was determined from empirical measurements of light intensity in a room. Finally, conclusions and future work are discussed in Section 5.

2 Problem Formulation

In this section we build a model of the multi-robot system, the environment, and the weighting function defining areas of importance in the environment. We then formulate the robust adaptive coverage problem with respect to this model.

Let there be n robots with positions $p_i(t)$ in a planar environment $Q \subset \mathbb{R}^2$. The environment is assumed to be compact and convex.[1] We call the tuple of all robot positions $P = (p_1, \ldots, p_n) \in Q^n$ the configuration of the multi-robot system, and we

[1] These assumptions can be relaxed to certain classes of nonconvex environments with obstacles [Breitenmoser et al., 2010, Pimenta et al., 2008a, Caicedo and Žefran, 2008].

assume that the robots move with integrator dynamics

$$\dot{p}_i = u_i, \tag{1}$$

so that we can control their velocities directly through the control input u_i. We define the Voronoi partition of the environment to be $V(P) = \{V_1(P), \dots, V_n(P)\}$, where

$$V_i(P) = \{q \in Q \mid \|q - p_i\| \le \|q - p_j\|, \forall j \neq i\},$$

and $\|\cdot\|$ is the ℓ^2-norm. We think of each robot i as being responsible for sensing in its associated Voronoi cell V_i. Next we define the communication network as an undirected graph in which all robots whose Voronoi cells touch share an edge in the graph. This graph is known as the Delaunay graph. Then the set of neighbors of robot i is defined as $\mathscr{N}_i := \{j \mid V_i \cup V_j \neq \emptyset\}$.

We now define a weighting function over the environment $\phi : Q \mapsto \mathbb{R}_{>0}$ (where $\mathbb{R}_{>0}$ denotes the strictly positive real numbers). This weighting function is *not known* by the robots. Intuitively, we want a high density of robots in areas where $\phi(q)$ is large and a lower density where it is small. Finally, suppose that the robots have sensors with which they can measure the value of the weighting function at their own position, $\phi(p_i)$ with very high precision, but that their quality of sensing at arbitrary points, $\phi(q)$, degrades quadratically in the distance between q and p_i. That is to say the cost of a robot at p_i sensing a point at q is given by $\frac{1}{2}\|q - p_i\|^2$. Since each robot is responsible for sensing in its own Voronoi cell, the cost of all robots sensing over all points in the environment is given by

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

This is the overall objective function that we would like to minimize by controlling the configuration of the multi-robot system.[2]

The gradient of $\mathscr{H}$ can be shown[3] to be given by

$$\frac{\partial \mathscr{H}}{\partial p_i} = -\int_{V_i(P)} (q - p_i)\phi(q)\, dq = -M_i(P)(C_i(P) - p_i), \tag{3}$$

where we define $M_i(P) := \int_{V_i(P)} \phi(q)\, dq$ and $C_i(P) := 1/M_i(P) \int_{V_i(P)} q\phi(q)\, dq$. We call M_i the mass of the Voronoi cell i and C_i its centroid, and for efficiency of notation we will henceforth write these without the dependence on P. We would like to control the robots to move to their Voronoi centroids, $p_i = C_i$ for all i, since from (3), this is a critical point of $\mathscr{H}$, and if we reach such a configuration using gradient descent, we know it will be a local minimum. Global optimization of $\mathscr{H}$ is known to be NP-hard, hence it is standard in the literature to only consider local optimality.

---

[2] We have pursued an intuitive development of this cost function, though more rigorous arguments can also be made [Schwager et al., 2011]. This function is known in several fields of study including the placement of retail facilities [Drezner, 1995] and data compression [Lloyd, 1982].

[3] The computation of this gradient is more complex than it may seem, because the Voronoi cells $V_i(P)$ depend on P, which results in extra integral terms. Fortunately, these extra terms all sum to zero, as shown in, e.g. [Pimenta et al., 2008b].

2.1 Approximate Weighting Function

Note that the cost function (2) and its gradient (3) rely on the weighting function $\phi(q)$, which is not known to the robots. In this paper we provide a means by which the robots can approximate $\phi(q)$ online in a distributed way and move to decrease (2) with respect to this approximate $\phi(q)$.

To be more precise, each robot maintains a separate approximation of the weighting function, which we denote $\hat{\phi}(q,t)$. These approximate weighting functions are generated from a linear combination of m static basis functions, $\mathscr{K}(q) = [\mathscr{K}_1(q) \cdots \mathscr{K}_m(q)]^{\mathrm{T}}$, where each basis function is a radially symmetric Gaussian of the form

$$\mathscr{K}_j(q) = \frac{1}{2\pi\sigma} \exp\left\{-\frac{\|q-\mu_j\|^2}{2\sigma^2}\right\}, \tag{4}$$

with fixed width σ and fixed center μ_j. Furthermore, the centers are arranged in a regular grid over Q. Each robot then forms its approximate weighting function as a weighted sum of these basis functions $\hat{\phi}_i(q,t) = \mathscr{K}(q)^{\mathrm{T}}\hat{a}_i(t)$, where $\hat{a}_i(t)$ is the parameter vector of robot i. Each element in the parameter vector is constrained to lie within some lower and upper bounds $0 < a_{\min} < a_{\max} < \infty$ so that $\hat{a}_i(t) \in [a_{\min}, a_{\max}]^m$. This function approximation scheme is illustrated in Fig. 1. Robot i's approximation of its Voronoi cell mass and centroid can then be defined as

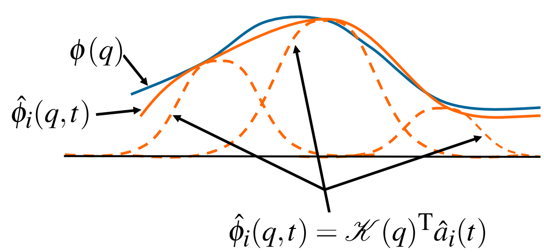

Fig. 1 The weighting function approximation is illustrated in this simplified 2-D schematic. The true weighting function $\phi(q)$ is approximated by robot i to be $\hat{\phi}_i(q,t)$. The basis function vector $\mathscr{K}(q)$ is shown as three Gaussians (dashed curves), and the parameter vector $\hat{a}_i(t)$ denotes the weighting of each Gaussian.

$\hat{M}_i(P,t) := \int_{V_i(P)} \hat{\phi}_i(q,t)\,dq$ and $\hat{C}_i(P,t) := 1/\hat{M}_i(P,t)\int_{V_i(P)} q\hat{\phi}_i(q,t)\,dq$, respectively. Again, we will drop the dependence of $\hat{M}_i$ and $\hat{C}_i$ on (P,t) for notational simplicity.

We measure the difference between an approximate weighting function and the true weighting function as the L^∞ function norm of their difference, so that the best approximation is given by

$$a := \underset{\hat{a}\in[a_{\min},a_{\max}]^m}{\arg\min}\ \max_{q\in Q} |\mathscr{K}(q)^{\mathrm{T}}\hat{a} - \phi(q)|, \tag{5}$$

and the optimal function approximation error is given by

$$\phi_\varepsilon(q) := \mathscr{K}(q)^{\mathrm{T}}a - \phi(q). \tag{6}$$

It will be shown in the proof of Theorem 1 that the L^∞ norm gives the tightest approximation bound with our proof technique. The only restriction that we put on

$\phi(q)$ is that it is bounded over the environment, or equivalently, the approximation error is bounded, $|\phi_\varepsilon(q)| \leq \phi_{\varepsilon_{\max}} < \infty$. We assume that the robots have knowledge of this bound, $\phi_{\varepsilon_{\max}}$. The theoretical analysis in the previous work [Schwager et al., 2009] was not robust to function approximation errors in that it required $\phi_\varepsilon(q) \equiv 0$. One of the main contributions here is to formulate an algorithm that is provably robust to function approximation errors. We only require that the robots have a known bound for the function approximation error, $\phi_{\varepsilon_{\max}}$.

Finally, we define the parameter error as $\tilde{a}_i(t) := \hat{a}_i(t) - a$. In what follows we describe an online tuning law by which robot i can tune its parameters, $\hat{a}_i$, to approach a neighborhood of the optimal parameters, a. Our proposed controller then causes the robots to converge to their approximate centroids, $p_i \to \hat{C}_i$ for all i. An overview of the geometrical objects involved in our set-up is shown in Figure 2.

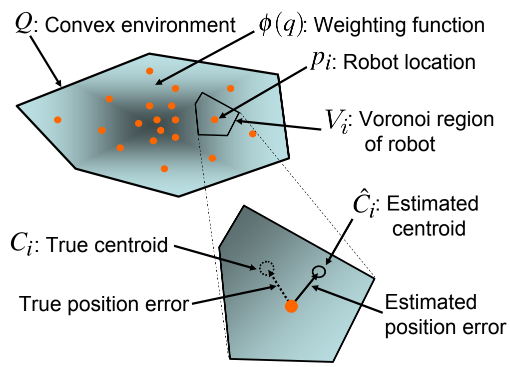

Fig. 2 A graphical overview of the quantities involved in the controller is shown. The robots move to cover a bounded, convex environment Q their positions are p_i, and they each have a Voronoi region V_i with a true centroid C_i and an estimated centroid $\hat{C}_i$. The true centroid is determined 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 $\phi(q)$.

3 Robust Adaptive Coverage Algorithm

In this section we describe the algorithm that drives the robots to spread out over the environment while simultaneously approximating the weighting function online. The algorithm naturally decomposes into two parts: (1) the parameter adaptation law, by which each robot updates its approximate weighting function, and (2) the control algorithm, which drives the robots to explore the environment before moving to their final positions for coverage. We describe these two parts in separate sections and then prove performance guarantees for the two working together.

3.1 Online Function Approximation

The parameters $\hat{a}_i$ used to calculate $\hat{\phi}_i(q,t)$ are adjusted according to a set of adaptation laws which are introduced below. First, we define two quantities,

$$\Lambda_i(t) = \Lambda_0 + \int_{s\in\Omega_i(t)} \mathscr{K}(s)\mathscr{K}(s)^{\mathrm{T}}\,ds, \text{ and } \lambda_i(t) = \int_{s\in\Omega_i(t)} \mathscr{K}(s)\phi(s)\,ds, \quad (7)$$

where $\Omega_i(t) = \{s \mid s = p_i(\tau) \text{ for some } \tau \in [0,t]\}$ is the set of points in the trajectory of p_i from time 0 to time t, and Λ_0 is a positive definite matrix. The quantities in (7) can be calculated differentially by robot i using $\dot{\Lambda}_i(t) = \mathscr{K}_i(t)\mathscr{K}_i(t)^{\mathrm{T}}|\dot{p}_i(t)|$ with initial condition Λ_0, and $\dot{\lambda}_i(t) = \mathscr{K}_i(t)\phi_i(t)|\dot{p}_i(t)|$ with zero initial conditions, where we introduced the shorthand notation $\mathscr{K}_i(t) := \mathscr{K}(p_i(t))$ and $\phi_i(t) := \phi(p_i(t))$. We require that $\Lambda_0 > 0$, though it can be arbitrarily small. This will ensure that $\Lambda_i(t) > 0$ for all time because $\int_{s\in\Omega_i(t)} \mathscr{K}(s)\mathscr{K}(s)^{\mathrm{T}}\,ds \geq 0$ and the sum of a positive semi-definite matrix and a positive definite matrix is positive definite. This, in turn, ensures that $\Lambda_i^{-1/2}$ always exists, which will be crucial in the control law and proof of convergence below. As previously stated, robot i can measure $\phi_i(t)$ with its sensors. Now we define another quantity

$$F_i = \frac{\int_{V_i} \mathscr{K}(q)(q-p_i)^{\mathrm{T}}\,dq \int_{V_i} (q-p_i)\mathscr{K}(q)^{\mathrm{T}}\,dq}{\int_{V_i} \hat{\phi}_i(q)\,dq}. \quad (8)$$

Notice that F_i can also be computed by robot i as it does not require any knowledge of the true weighting function, ϕ.

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

$$\dot{\hat{a}}_i^{\mathrm{pre}} = -\gamma B_{\mathrm{dz}}(\Lambda_i \hat{a}_i - \lambda_i) - \zeta \sum_{j\in\mathscr{N}_i} l_{ij}(\hat{a}_i - \hat{a}_j) - kF_i\hat{a}_i. \quad (9)$$

where γ, ζ, and k are positive gains, l_{ij} is the length of the shared Voronoi edge between robots i and j, and $B_{\mathrm{dz}}(\cdot)$ is a dead zone function which gives a zero if its argument is below some value. We will give B_{dz} careful attention in what follows as it is the main tool to ensure robustness to function approximation errors. Before describing the dead zone in detail, we note that the three terms in (9) have an intuitive interpretation. The first term is an integral of the function approximation error over the robot's trajectory, so that the parameter $\hat{a}_i$ is tuned to decrease this error. The second term is the difference between the robot's parameters and its neighbors' parameters. This term will be shown to lead to parameter consensus; the parameter vectors for all robots will approach a common vector. The third term compensates for uncertainty in the centroid position estimate, and will be shown to ensure convergence of the robots to their estimated centroids. A more in-depth explanation of each of these terms can be found in [Schwager et al., 2009].

Finally, we give the parameter adaptation law by restricting the "pre" adaptation law so that the parameters remain within their prescribed limits $[a_{\min}, a_{\max}]$ using a projection operator. We introduce a matrix I_{proj_i} defined element-wise as

$$I_{\mathrm{proj}_i} := \begin{cases} 0 \text{ for } a_{\min} < \hat{a}_i(j) < a_{\max} \\ 0 \text{ for } \hat{a}_i(j) = a_{\min} \text{ and } \dot{\hat{a}}_i^{\mathrm{pre}}(j) \geq 0 \\ 0 \text{ for } \hat{a}_i(j) = a_{\max} \text{ and } \dot{\hat{a}}_i^{\mathrm{pre}}(j) \leq 0 \\ 1 \text{ otherwise,} \end{cases} \quad (10)$$

where (j) denotes the j^{th} element for a vector and the j^{th} diagonal element for a matrix. The entries of I_{proj_i} are only nonzero if the parameter is about to exceed its bound. Now the parameters are changed according to the adaptation law

$$\dot{\hat{a}}_i = \Gamma(\dot{\hat{a}}_{\text{pre}_i} - I_{\text{proj}_i}\dot{\hat{a}}_{\text{pre}_i}), \tag{11}$$

where $\Gamma \in \mathbb{R}^{m\times m}$ is a diagonal, positive definite gain matrix. Although the adaptation law given by (11) and (9) is notationally complicated, it has a straightforward interpretation, it is of low computational complexity, and it is composed entirely of quantities that can be computed by robot i.

As mentioned above, the key innovation in this adaptation law compared with the one in [Schwager et al., 2009] is the dead zone function B_{dz}. We design this function so that the parameters are only changed in response to function errors that could be reduced with different parameters. More specifically, the minimal function error that can be achieved is ϕ_ε, as shown in (6). Therefore if the integrated parameter error $(\Lambda_i\hat{a}_i - \lambda_i)$ is less than ϕ_ε integrated over the robot's path, we have no reason to change the parameters. We will show that the correct form for the dead zone to prevent unnecessary parameter adaptation is

$$B_{\text{dz}}(x) = \begin{cases} 0 & \text{if } C(x) < 0 \\ \dfrac{x}{\|x\|}C(x) & \text{otherwise,} \end{cases} \tag{12}$$

where $C(x) := \|\Lambda_i^{1/2}\| \left(\|\Lambda_i^{-1/2}x\| - \|\Lambda_i^{-1/2}\beta_i\|\phi_{\varepsilon_{\max}} - \|\Lambda_i^{-1/2}\Lambda_0\|a_{\max}\right)$ and $\beta_i := \int_{s\in\Omega_i(t)} \mathscr{K}(s)\,ds$. This condition can be evaluated by robot i since $\beta_i(t)$ can be computed differentially from $\dot{\beta}_i = \mathscr{K}_i|\dot{p}_i|$ with zero initial conditions, we have already seen how to compute Λ_i, and $\phi_{\varepsilon_{\max}}$, $a_{\max}$, and Λ_0 are known.

3.2 Control Algorithm

We propose to use a control algorithm that is composed of a set of control modes, with switching conditions to determine when the robots change from one mode to the next. The robots first move to partition the basis function centers among one another, so that each center is assigned to one robot, then each robot executes a Traveling Salesperson (TSP) tour through all of the basis function centers that have been assigned to it. This tour will provide sufficient information so that the weighting function can be estimated well over all of the environment. Then the robots carry out a centroidal Voronoi controller using the estimated weighting function to drive to final positions. We call the first mode the "partitioning" mode, the second the "exploration" mode, and the third the "coverage" mode. This sequence of control modes is executed asynchronously in a distributed fashion, during which the function approximation parameters are updated continually with (11) and (9).

For each robot we define a mode variable $\mathscr{M}_{ii} \in \{\text{partition}, \text{explore}, \text{cover}\}$. In order to coordinate their mode switches, each robot also maintains an estimate of the modes of all the other robots, so that $\mathscr{M}_{ij}$ is the estimate by robot i of robot j's mode, and $\mathscr{M}_i := (\mathscr{M}_{i1}, \ldots, \mathscr{M}_{in})$ is an n-tuple of robot i's estimates of all robots' modes. Furthermore, the modes are ordered with respect to one another by partition $<$ explore $<$ cover, so that the $\max(\mathscr{M}_i, \mathscr{M}_j)$ function is the maximum between each element of the two mode estimate tuples, $\mathscr{M}_i$ and $\mathscr{M}_j$, according to this ordering. These mode estimates are updated using the flooding communication protocol described below. We first describe the algorithmic structure of the controller, then define the behavior within each mode, and finally prove the convergence of the coupled control algorithm and learning algorithm to a desirable final configuration.

The two algorithms below run concurrently in different threads. Algorithm 1 defines the switching conditions between control modes, and Algorithm 2 describes the flooding protocol that each robot uses to maintain its mode estimates.

Algorithm 1 Switching Control Algorithm (executed by robot i)

Require: Communication with Voronoi neighbors.
Require: Knowledge of position, p_i, in global coordinate frame.
Require: Knowledge of the total number of robots n.
Require: Knowledge of flooding algorithm (Algorithm 2) update period, T.
Require: Access to mode estimates $\mathscr{M}_i$ updated from Algorithm 2.

```
while M_i ≠ (explore,...,explore) do
    if M_ii == explore then
        u_i = [0, 0]^T
    else
        u_i = u_i^partition
    end if
    if Distance to mean of basis function centers < ε^partition and M_ii == partition then
        M_ii = explore
    end if
end while
Compute TSP tour through basis function centers N_i^μ
Wait for nT seconds with u_i = [0, 0]^T
Execute TSP tour
M_ii = cover
while M_ii == cover do
    u_i = u_i^cover
end while
```

Algorithm 2 Mode Estimate Flooding (executed by robot i)

Require: The network is connected.
Require: The robots have synchronized clocks with which they broadcast during a pre-assigned time slot.

```
Initialize M_i = (partition,...,partition)
while 1 do
    if Broadcast received from robot j then
        M_i = max(M_i, M_j) (where partition < explore < cover)
    end if
    if Robot i's turn to broadcast then
        Broadcast M_i
    end if
end while
```

The control laws within each mode are then defined as follows. In the partition mode, each robot uses the controller

$$u_i^{\text{partition}} = k\Big(\frac{1}{|\mathcal{N}_i^{\mu}|}\sum_{j\in\mathcal{N}_i^{\mu}} \mu_j - p_i\Big), \tag{13}$$

where $\mathcal{N}_i^{\mu} := \{\mu_j \mid \|\mu_j - p_i\| \leq \|\mu_j - p_k\| \forall k \neq i\}$ is the set of the closest basis function centers to robot i, μ_j are the basis function centers from (4), and $|\mathcal{N}_i^{\mu}|$ is the number of elements in $\mathcal{N}_i^{\mu}$. In the explore mode, each robot drives a tour through each basis function center in its neighborhood, μ_j for $j \in \mathcal{N}_i^{\mu}$. Any tour will do, but a good choice is to use an approximate TSP tour. Finally, for the "cover" mode, each robot moves toward the centroid of its Voronoi cell using

$$u_i^{\text{cover}} = k(\hat{C}_i - p_i), \tag{14}$$

where k is the same positive gain from (9).

Using the above control and function approximation algorithm, we can prove that all robots converge to the estimated centroid of their Voronoi cells, that all robots function approximation parameters converge to the same parameter vector, and that this parameter vector has a bounded error with the optimal parameter vector. This is stated formally in the following theorem.

Theorem 1 (Convergence). *A network of robots with dynamics (1) using Algorithm 1 for control, Algorithm 2 for communication, and (9) and (11) for online function approximation has the following convergence guarantees:*

$$\lim_{t\to\infty} \|p_i(t) - \hat{C}_i(P,t)\| = 0 \quad \forall i, \tag{15}$$

$$\lim_{t\to\infty} \|\hat{a}_i(t) - \hat{a}_j(t)\| = 0 \quad \forall i,j, \tag{16}$$

and

$$\lim_{t\to\infty} \|\hat{a}_i(t) - a\| \leq \frac{\sum_{j=1}^{n} 2\|\Lambda_j(t)^{1/2}\|\Big(\|\Lambda_j(t)^{-1/2}\beta_j(t)\|\phi_{\varepsilon_{\max}} + \|\Lambda_j(t)^{-1/2}\Lambda_0\|a_{\max}\Big)}{\text{mineig}(\sum_{j=1}^{n}\Lambda_j(t))} \quad \forall i. \tag{17}$$

Proof. The proof has two parts. The first part is to show that all robots reach "cover" mode and stay in "cover" mode. The second part uses a Lyapunov type proof technique similar to the one in [Schwager et al., 2009] to show that once all robots are in "cover" mode, the convergence claims of (15), (16), and (17) follow.

Firstly, the "partition" mode simply implements a K-means clustering algorithm [Duda et al., 2001], in which the basis function centers are the points to be clustered, and the robots move to the cluster means. This algorithm is well-known to converge in the sense that for any $\varepsilon^{\text{partition}}$ there exists a time $T_i^{\text{partition}}$ at which the distance between the robot and the centers' mean is less than $\varepsilon^{\text{partition}}$, therefore all robots will reach $\mathcal{M}_{ii} =$ explore at some finite time. After this time, according to Algorithm 1, a robot will remain stopped until all of its mode estimates have switched to "explore." Suppose the first robot to achieve $\mathscr{M}_i = (\text{explore}, \ldots, \text{explore})$ does so at time T_f. This means that at some time in the past all the other robots, j, have switched to $\mathscr{M}_{jj} = \text{explore}$ and stopped moving, but none of them have $\mathscr{M}_j = (\text{explore}, \ldots, \text{explore})$ (otherwise they would be the first). Therefore at T_f all robots are stopped. Suppose the last robot to achieve $\mathscr{M}_i = (\text{explore}, \ldots, \text{explore})$ does so at T_l. From the properties of Algorithm 2 we know that $T_l - T_f \leq nT$ (the maximum time between the first robot to obtain $\mathscr{M}_i = (\text{explore}, \ldots, \text{explore})$ and the last robot to do so is nT). At time T_l, the first robot to have $\mathscr{M}_i = (\text{explore}, \ldots, \text{explore})$ will still be stopped, because it waits for nT seconds after achieving $\mathscr{M}_i = (\text{explore}, \ldots, \text{explore})$, hence when any robot obtains $\mathscr{M}_i = (\text{explore}, \ldots, \text{explore})$, all other robots are stopped. Even though the robots may compute their TSP tours at different times, they are all at the same positions when they do so. Therefore, each basis function center is in at least one robot's TSP tour. Consequently, when all robots have completed their TSP tours, $\text{mineig}\big(\sum_{i=1}^n \Lambda_i\big)$ will be similar in size to $\text{maxeig}\big(\sum_{i=1}^n \Lambda_i\big)$, making the bound in (17) small.

Each tour is finite length, so it will terminate in finite time, hence each robot will eventually enter the "cover" mode. Furthermore, when some robots are in "cover" mode, and some are still in "explore" mode, the robots in "cover" mode will remain inside the environment Q. This is because Q is convex, and since $\hat{C}_i \in V_i \subset Q$ and $p_i \in Q$, by convexity, the segment connecting the two is in Q. Since the robots have integrator dynamics (1), they will stay within the union of these segments over time, $p_i(t) \in \cup_{\tau>0}(C_i(\tau) - p_i(\tau))$, and therefore remain in the environment Q. Thus at some finite time, T^{cover}, all robots reach "cover" mode and are at positions inside Q.

Now, define a Lyapunov-like function

$$\mathscr{V} = \mathscr{H} + \frac{1}{2}\sum_i \tilde{a}_i^{\mathrm{T}} \Gamma^{-1} \tilde{a}_i, \tag{18}$$

which incorporates the sensing cost $\mathscr{H}$, and is quadratic in the parameter errors $\tilde{a}_i$. We will use Barbalat's lemma to prove that $\dot{\mathscr{V}} \to 0$ and then show that the claims of the theorem follow. Barbalat's lemma requires that $\mathscr{V}$ is lower bounded, non-increasing, and uniformly continuous. $\mathscr{V}$ is bounded below by zero since $\mathscr{H}$ is a sum of integrals of strictly positive functions, and the quadratic parameter error terms are each bounded below by zero.

Now we will show that $\dot{\mathscr{V}} \leq 0$. Taking the time derivative of $\mathscr{V}$ along the trajectories of the system and simplifying with (3) gives

$$\dot{\mathscr{V}} = \sum_i \left(-\|\hat{C}_i - p_i\|^2 k\hat{M}_i + \tilde{a}_i^{\mathrm{T}} kF_i\hat{a}_i + \tilde{a}_i^{\mathrm{T}}\Gamma^{-1}\dot{\hat{a}}_i\right).$$

Substituting for $\dot{\hat{a}}_i$ with (11) and (9) gives

$$\dot{\mathscr{V}} = -\sum_i \Big(\|\hat{C}_i - p_i\|^2 k\hat{M}_i + \gamma\tilde{a}_i^{\mathrm{T}} B_{\mathrm{dz}}(\Lambda_i\tilde{a}_i + \lambda_{\varepsilon_i}) + \zeta\tilde{a}_i^{\mathrm{T}} \sum_{j\in\mathscr{N}_i} l_{ij}(\hat{a}_i - \hat{a}_j) + \tilde{a}_i^{\mathrm{T}} I_{\mathrm{proj}_i}\dot{\hat{a}}_{\mathrm{pre}_i}\Big),$$

where $\lambda_{\varepsilon_i} := \int_{s\in\Omega_i(t)} \mathscr{K}(s)\phi_\varepsilon(s)\,ds + \Lambda_0 a$. Rearranging terms we get

$$\dot{\mathscr{V}} = -\sum_i \left(\|\hat{C}_i - p_i\|^2 k\hat{M}_i + \gamma \tilde{a}_i^{\mathrm{T}} B_{\mathrm{dz}}(\Lambda_i \tilde{a}_i + \lambda_{\varepsilon_i}) + \tilde{a}_i^{\mathrm{T}} I_{\mathrm{proj}_i} \dot{\hat{a}}_{\mathrm{pre}_i}\right) - \zeta \sum_{j=1}^{m} \hat{\alpha}_j^{\mathrm{T}} L(P) \hat{\alpha}_j, \tag{19}$$

where $\hat{\alpha}_j := [\hat{a}_{1j} \cdots \hat{a}_{mj}]^{\mathrm{T}}$ is the j^{th} element in every robot's parameter vector, stacked into a vector, and $L(P) \geq 0$ is the graph Laplacian for the Delaunay graph which defines the robots communication network (please refer to the proof of Theorem 2 in [Schwager et al., 2009] for details). The first term inside the first sum is the square of a norm, and therefore is non-negative. The third term in the first sum is non-negative by design (please see the proof of Theorem 1 from [Schwager et al., 2009] for details). The second sum is nonnegative because $L(P) \geq 0$. Therefore, the term with the dead-zone operator B_{dz} is the only term in question, and this distinguishes the Lyapunov construction here from the one in the proof of Theorem 1 in [Schwager et al., 2009].

We now show that the dead-zone term is also non-negative by design. Suppose the condition $C(\Lambda_i \tilde{a}_i + \lambda_{\varepsilon_i}) < 0$ from (12). Then $B_{\mathrm{dz}}(\Lambda \tilde{a}_i + \lambda_{\varepsilon_i}) = 0$ and the term is zero. Now suppose $C(\Lambda_i \tilde{a}_i + \lambda_{\varepsilon_i}) \geq 0$. In that case we have

$$0 \leq \|\Lambda_i^{1/2}\| \left(\|\Lambda_i^{-1/2}(\Lambda_i \tilde{a}_i + \lambda_{\varepsilon_i})\| - \|\Lambda_i^{-1/2}\beta_i\|\phi_{\varepsilon_{\max}} - \|\Lambda_i^{-1/2}\Lambda_0\| a_{\max} \right),$$

which implies

$$\begin{aligned}
0 &\leq \|\Lambda_i^{-1/2}(\Lambda_i \tilde{a}_i + \lambda_{\varepsilon_i})\| - \|\Lambda_i^{-1/2}\beta_i\|\phi_{\varepsilon_{\max}} - \|\Lambda_i^{-1/2}\Lambda_0\| a_{\max} \\
&\leq \|\Lambda_i^{1/2}\tilde{a}_i + \Lambda^{-1/2}\lambda_{\varepsilon_i}\| - \left\| \Lambda_i^{-1/2} \left(\int_{\Omega_i(t)} \mathscr{K}(s)\phi_\varepsilon(s)\,ds + \Lambda_0 a \right) \right\| \\
&\leq \|\Lambda_i^{1/2}\tilde{a}_i + \Lambda_i^{-1/2}\lambda_{\varepsilon_i}\|^2 - (\Lambda_i^{-1/2}\lambda_{\varepsilon_i})^{\mathrm{T}}(\Lambda_i^{1/2}\tilde{a}_i - \Lambda_i^{-1/2}\lambda_{\varepsilon_i}) \\
&= (\Lambda_i^{1/2}\tilde{a}_i)^{\mathrm{T}}(\Lambda_i^{1/2}\tilde{a}_i + \Lambda_i^{-1/2}\lambda_{\varepsilon_i}) \\
&= \tilde{a}_i^{\mathrm{T}}(\Lambda_i \tilde{a}_i + \lambda_{\varepsilon_i}).
\end{aligned} \tag{20}$$

Then from the definition of the dead-zone operator, $B_{\mathrm{dz}}(\cdot)$, we have

$$\tilde{a}^{\mathrm{T}} B_{\mathrm{dz}}(\Lambda_i \tilde{a}_i + \lambda_{\varepsilon_i}) = \frac{\tilde{a}^{\mathrm{T}}(\Lambda_i \tilde{a}_i + \lambda_{\varepsilon_i})}{\|\Lambda_i \tilde{a}_i + \lambda_{\varepsilon_i}\|} C(\Lambda_i \tilde{a}_i + \lambda_{\varepsilon_i}) \geq 0,$$

since the numerator was shown to be non-negative in (20), and $C(\cdot)$ is non-negative by supposition. Therefore the dead-zone term is non-negative in this case as well. We conclude therefore that $\dot{\mathscr{V}} \leq 0$.

The final condition to prove that $\dot{\mathscr{V}} \to 0$ is that $\dot{\mathscr{V}}$ must be uniformly continuous, which is a technical condition that was shown in [Schwager et al., 2009], Lemmas 1 and 2. These lemmas also apply to our case, since our function $\dot{\mathscr{V}}$ is the same as in that case except for the dead-zone term. Following the argument of those lemmas, the dead-zone term has a bounded derivative everywhere, except where it is non-differentiable, and these point of non-differentiability are isolated. Therefore, it is Lipschitz continuous and hence uniformly continuous, and we conclude by Barbalat's lemma that $\dot{\mathscr{V}} \to 0$.

Now we show that $\dot{\mathscr{V}} \to 0$ implies the convergence claims stated in the theorem. Firstly, since all the terms in (19) are non-negative, each one must separately approach zero. The first term approaching zero gives the position convergence (15), the last term approaching zero gives parameter consensus (16) (again, see [Schwager et al., 2009] for more details on these). Finally, we verify the parameter error convergence (17). We know that the dead-zone term approaches zero, therefore either $\lim_{t\to\infty} \tilde{a}_i^{\mathrm{T}}(\Lambda_t \tilde{a}_i + \lambda_{\varepsilon_i}) = 0$, or $\lim_{t\to\infty} C(\Lambda_i \tilde{a}_i + \lambda_{\varepsilon_i}) \le 0$. We already saw from (20) that $\tilde{a}_i^{\mathrm{T}}(\Lambda_t \tilde{a}_i + \lambda_{\varepsilon_i}) = 0$ implies $C(\Lambda_i \tilde{a}_i + \lambda_{\varepsilon_i}) \le 0$, thus we only consider this later case. To condense notation at this point, we introduce $d_i := \|\Lambda_i^{1/2}\|\big(\|\Lambda_i^{-1/2}\beta_i\|\phi_{\varepsilon_{\max}} - \|\Lambda_i^{-1/2}\Lambda_0\|a_{\max}\big)$. Then from $\lim_{t\to\infty} C(\Lambda_i \tilde{a}_i + \lambda_{\varepsilon_i}) \le 0$ we have

$$0 \ge \lim_{t\to\infty}\big(\|\Lambda_i^{1/2}\|\|\Lambda_i^{-1/2}(\Lambda_i\tilde{a}_i + \lambda_{\varepsilon_i})\| - d_i\big) \ge \lim_{t\to\infty}\big(\|\Lambda_i\tilde{a}_i + \lambda_{\varepsilon_i}\| - d_i\big),$$

and because of parameter consensus (16), $0 \ge \lim_{t\to\infty}\big(\|\Lambda_j\tilde{a}_i + \lambda_{\varepsilon_j}\| - d_j\big)$ for all j. Then summing over j, we have

$$\begin{aligned}
0 &\ge \lim_{t\to\infty}\Big(\sum_{j=1}^n \|\Lambda_j\tilde{a}_i + \lambda_{\varepsilon_j}\| - \sum_{j=1}^n d_j\Big) \\
&\ge \lim_{t\to\infty}\Big(\|\sum_{j=1}^n \Lambda_j\tilde{a}_i + \sum_{j=1}^n \lambda_{\varepsilon_j}\| - \sum_{j=1}^n d_j\Big) \\
&\ge \lim_{t\to\infty}\Big(\Big|\|\sum_{j=1}^n \Lambda_j\tilde{a}_i\| - \|\sum_{j=1}^n \lambda_{\varepsilon_j}\|\Big| - \sum_{j=1}^n d_j\Big).
\end{aligned}$$

The last condition has two possibilities; either $\lim_{t\to\infty}\big(\|\sum_{j=1}^n \Lambda_j\tilde{a}_i\| > \|\sum_{j=1}^n \lambda_{\varepsilon_j}\|\big)$, in which case

$$0 \ge \lim_{t\to\infty}\Big(\|\sum_{j=1}^n \Lambda_j\tilde{a}_i\| - \|\sum_{j=1}^n \lambda_{\varepsilon_i}\| - \sum_{j=1}^n d_j\Big) \ge \lim_{t\to\infty}\Big(\|\sum_{j=1}^n \Lambda_j\tilde{a}_i\| - 2\sum_{j=1}^n d_j\Big), \quad (21)$$

where the last inequality uses the fact that $\big\|\sum_{j=1}^n \lambda_{\varepsilon_i}\big\| \le \sum_{j=1}^n d_j$. Otherwise, $\lim_{t\to\infty}\big(\|\sum_{j=1}^n \Lambda_j\tilde{a}_i\| < \|\sum_{j=1}^n \lambda_{\varepsilon_j}\|\big)$, which implies $\lim_{t\to\infty}\big(\|\sum_{j=1}^n \Lambda_j\tilde{a}_i\| < \sum_{j=1}^n d_j\big)$ which in turn implies (21), thus we only need to consider (21). This expression then leads to $0 \ge \lim_{t\to\infty}\big(\mathrm{mineig}\big(\sum_{j=1}^n \Lambda_j\big)\|\tilde{a}_i\| - 2\sum_{j=1}^n d_j\big)$, and dividing both sides by $\mathrm{mineig}\big(\sum_{j=1}^n \Lambda_j\big)$ (which is strictly positive since $\sum_{j=1}^n \Lambda_j > 0$), gives (17). □

4 Simulation Results

The proposed algorithm is tested using the data collected from previous experiments [Schwager et al., 2008] in which two incandescent office lights were placed at the position (0,0) of the environment, and the robots used on-board light sensors to measure the light intensity. The data collected during these previous experiments was used to generate a realistic weighting function which cannot be reconstructed exactly by the chosen basis functions. In the simulation, there are 10 robots and the basis functions are arranged on a 15×15 grid in the environment.

The proposed robust algorithm was compared against the standard algorithm from [Schwager et al., 2009], which assumes that the weighting function can be matched exactly. The true and optimally reconstructed (according to (5)) weighting functions are shown in Figure 3(a) and (b), respectively. As shown in Figure 3(c) and (d), with the robust and standard algorithm respectively, the proposed robust algorithm significantly outperforms the standard algorithm and reconstructs the true weighting function well. The robot trajectories for the robust and standard adaptive coverage algorithm are shown in Figure 4(a) and (b), respectively. Since the standard algorithm doesn't include the exploration phase, the robots get stuck in a local area around their starting position which causes the robots to be unsuccessful in learning an acceptable model of the weighting function. In contrast, the robots with the robust algorithm explore the entire space and reconstruct the true weighting function well.

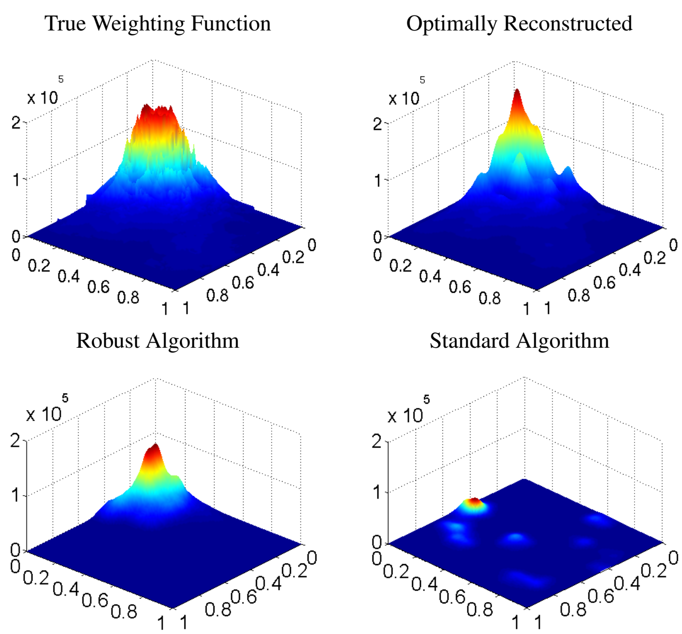

Fig. 3 A comparison of the weighting functions. (a) The true weighting function. (b) The optimally reconstructed weighting function for the chosen basis functions. (c) The weighting function for the proposed algorithm with deadzone and exploration. (d) The previously proposed algorithm without deadzone or exploration.

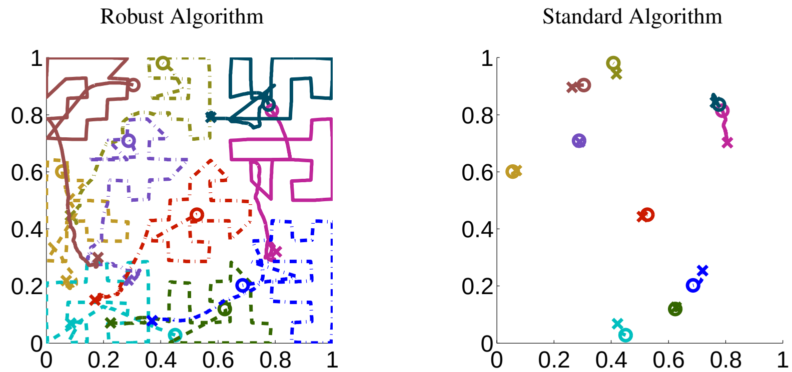

Fig. 4 The vehicle trajectories for the different control strategies. The initial and final vehicle position is marked by a circle and cross, respectively.

5 Conclusions

In this paper we formulated a distributed control and function approximation algorithm for deploying a robotic sensor network to adaptively monitor an environment. The robots robustly learn a weighting function over the environment representing where sensing is most needed. The function learning is enabled by the robots exploring the environment in a systematic and distributed way, and is provably robust to function approximation errors. After exploring the environment, the robots drive to positions that locally minimize a cost function representing the sensing quality of the network. The performance of the algorithm is proven in a theorem, and demonstrated in a numerical simulation with an empirical weighting function derived from light intensity measurements in a room. The authors are currently working toward hardware experiments to prove the practicality of the algorithm.

Acknowledgements This work was funded in part by ONR MURI Grants N00014-07-1-0829, N00014-09-1-1051, and N00014-09-1-1031, and the SMART Future Mobility project. We are grateful for this financial support.

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.

[Breitenmoser et al., 2010] Breitenmoser, A., Schwager, M., Metzger, J. C., Siegwart, R., and Rus, D. (2010). Voronoi coverage of non-convex environments with a group of networked robots. In *Proceedings of the International Conference on Robotics and Automation (ICRA 10)*, pages 4982–4989, Anchorage, Alaska, USA.

[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.

[Caicedo and Žefran, 2008] Caicedo, C. and Žefran, M. (2008). Performing coverage on nonconvex domains. In *Proceedings of the 2008 IEEE Multi-Conference on Systems and Control*, pages 1019–1024.

[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.

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

[Duda et al., 2001] Duda, R., Hart, P., and Stork, D. (2001). *Pattern Classification*. Wiley and Sons, New York.

[Ioannou and Kokotovic, 1984] Ioannou, P. and Kokotovic, P. V. (1984). Instability analysis and improvement of robustness of adaptive control. *Automatica*, 20(5):583–594.

[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.

[Lloyd, 1982] Lloyd, S. P. (1982). Least squares quantization in pcm. *IEEE Transactions on Information Theory*, 28(2):129–137.

[Martínez, 2010] Martínez, S. (2010). Distributed interpolation schemes for field estimation by mobile sensor networks. *IEEE Transactions on Control Systems Technology*, 18(2):419–500.

[Narendra and Annaswamy, 1987] Narendra, K. and Annaswamy, A. (1987). A new adaptive law for robust adaptation without persistent excitation. *IEEE Transactions on Automatic Control*, 32(2):134 – 145.

[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.

[Peterson and Narendra, 1982] Peterson, B. and Narendra, K. (1982). Bounded error adaptive control. *IEEE Transactions on Automatic Control*, 27(6):1161 – 1168.

[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 08)*, Guanajuato, Mexico.

[Samson, 1983] Samson, C. (1983). Stability analysis of adaptively controlled systems subject to bounded disturbances. *Automatica*, 19(1):81 – 86.

[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., 2008] Schwager, M., McLurkin, J., Slotine, J. J. E., and Rus, D. (2008). From theory to practice: Distributed coverage control experiments with groups of robots. In *Experimental Robotics: The Eleventh International Symposium*, volume 54, pages 127–136. Springer-Verlag.

[Schwager et al., 2009] Schwager, M., Rus, D., and Slotine, J. J. (2009). Decentralized, adaptive coverage control for networked robots. *International Journal of Robotics Research*, 28(3):357–375.

[Schwager et al., 2011] Schwager, M., Rus, D., and Slotine, J. J. (2011). Unifying geometric, probabilistic, and potential field approaches to multi-robot deployment. *International Journal of Robotics Research*, 30(3):371–383.

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

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