

A Ladybug Exploration Strategy for Distributed Adaptive Coverage Control

Mac Schwager*, Francesco Bullo[†], David Skelly[‡], and Daniela Rus*

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

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

[†]Department of Mechanical Engineering
University of California at Santa Barbara
Santa Barbara, CA 93106

Email: bullo@engineering.ucsb.edu

[‡]Department of Ecology and Evolutionary Biology
Yale University, New Haven, CT 93106
Email: david.skelly@yale.edu

Abstract—A control strategy inspired by the hunting tactics of ladybugs is presented to simultaneously achieve sensor coverage and exploration of an area with a group of networked robots. The controller is distributed in that it requires only information local to each robot, and adaptive in that it modifies its behavior based on information in the environment. The ladybug controller is developed as a modification to a basic coverage control law, first for the non-adaptive case, then for the adaptive case. Stability is proven for both cases with a Lyapunov-type proof. Results of numerical simulations are presented.

I. INTRODUCTION

We present a decentralized adaptive control law that causes networked robots to explore an area while simultaneously searching for an optimal coverage configuration for sensing over the area. The exploration strategy is modeled on one used by ladybugs in their hunt for aphids (Fig. 1). Furthermore, the exploration strategy is provably stable and can be arbitrarily aggressive. The ladybug control law improves upon previous coverage controllers by avoiding locally optimal configurations and providing richer trajectories for better learning of the distribution of sensory information in the environment. Our controller would be useful in controlling teams of robots to carry out a number of tasks including search and rescue missions, environmental monitoring (e.g. for forest fires), automatic surveillance of rooms, buildings, or towns, or simulating collaborative predatory behavior. Virtually any application in which a group of automated mobile agents is required to both monitor and explore an area could benefit from the proposed controller.

The two notions we investigate in this paper, coverage and exploration, are often interchanged, and therefore it is important to clearly delineate what we mean by each. By

coverage, we refer specifically to the locational cost function introduced in [1]. A high value of this function corresponds to poor coverage and a low value to good coverage. Furthermore, the function has well-defined minima corresponding to static optimal coverage configurations. This is quite different from the other common notion of coverage defined as the total area swept out by a sensing disk surrounding an agent. In fact, the swept sensing disk notion is closely related to what we call exploration. Specifically, we define the degree to which a robot trajectory is exploratory by the minimum eigenvalue of a certain matrix, as defined in Section IV-A, and we show that this value is linked to the notion of a sweeping disk. In this work, our main objective is



Fig. 1. The exploration strategy that ladybugs use to hunt for aphids was used to inspire a decentralized controller for networked robots to perform exploration and coverage tasks.

coverage (minimizing the locational cost function), while exploration (maximizing the swept disk area) is considered as an additional desirable property. We show that a certain class of exploration, “ladybug exploration,” can proceed without disrupting the coverage objective. Other kinds of exploration, e.g. a random walk, may prevent convergence

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, and DEB-0640892.

to an optimal coverage configuration, and therefore subvert the main objective.

The controller that we present superimposes a ladybug exploration force on a basic coverage controller causing both exploration and coverage to proceed orthogonal to one another. Robots explore the whole area, at first aggressively, while they have little information about the sensing environment, and increasingly less aggressively as they collect information about the sensing environment and settle into an optimal configuration for coverage. The exploration force causes the robots to spiral around regions of high sensory interest, which is analogous to the way ladybugs turn more tightly around areas in which they encounter aphids [2], [3]. The robots asymptotically spiral in to fixed positions such that their density in different regions of the environment is directly related to the sensory importance of those regions. Thus regions of greater importance receive more concentrated sensor coverage than regions that are less important.

The control law is presented first in a non-adaptive form, then in an adaptive form. The main difference between the two forms is that for the non-adaptive controller, the robots are assumed to know *a priori* the distribution of sensory information in the environment, whereas in the adaptive form, they learn the distribution using an adaptation law. The non-adaptive controller is simpler, providing an intuitive platform from which to describe the ladybug exploration force. For the non-adaptive controller, the main effect of the ladybug exploration is to prevent the robots from getting trapped in locally optimal configurations. In this way it has an effect similar to simulated annealing strategies [4], [5], though without any stochasticity.

On the other hand, the main benefit of ladybug exploration for the adaptive controller is to improve parameter learning. The adaptive controller uses an adaptation law to learn the distribution of sensory information in the environment from collected sensor measurements. The fidelity of the learned distribution to the actual distribution is intimately related to the extent to which the robot has explored the environment. This fact is formalized in the persistent excitation conditions, which are discussed in Section IV-A. The ladybug controller, by driving each robot to explore the environment, provides markedly improved learning performance over the basic adaptive coverage controller. This leads to markedly improved final configurations and faster convergence.

A. Relation to Previous Work

The basic coverage controller relevant to this work was first introduced in [1] in a non-adaptive form, and in [6] in an adaptive form. The ladybug exploration term can be included in either controller in essentially the same way. It was shown in [1] that coverage control can be phrased as a locational optimization problem [7]. Many controllers using this paradigm have been proposed, for example [8]–[10]. Notably, in [8] a deterministic annealing approach was adapted to the locational optimization problem to find globally optimal solutions. The adaptive coverage controller presented in [6] combines techniques from adaptive control

[11]–[13] with the controller in [1] to learn the distribution of sensory information in the environment.

In this paper we use a deterministic framework, which is common in the coverage control and adaptive control literature. In Section II we set up the problem and introduce the necessary tools from locational optimization. In Section III we formulate the non-adaptive ladybug controller, and show that it can provide improved performance over the basic controller by avoiding local minima. In Section IV, we introduce the adaptive ladybug controller, formalize persistent excitation conditions, and show that it provides superior performance to the basic adaptive controller in both learning and final coverage configuration. Conclusions are given in Section V.

II. PROBLEM SET-UP

A group of n robots with positions $p_i \in \mathbb{R}^2$, where $i \in \{1, \dots, n\}$, move in a bounded, convex region $Q \subset \mathbb{R}^2$. The function $\phi : Q \mapsto \mathbb{R}_+$ (henceforth \mathbb{R}_+ refers to the strictly positive orthant), determines a weighting of importance of points $q \in Q$. Let $\{V_1, \dots, V_n\}$ be the Voronoi partition of Q , for which the robot positions are the generator points. Specifically,

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

We can formulate a function denoting the cost incurred by the group of robots sensing over the region Q as

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

The weighting $\frac{1}{2} \|q - p_i\|^2$ is included to capture the intuition that a robot located at p_i will sense information at a point q with less reliability the farther q is from p_i , thus points q that are farther away are more expensive. Notice also that high values of $\phi(q)$ are expensive. An optimal network configuration corresponds to a set of robot positions that minimizes (1). The subject dealing with optimization problems of this kind is called Locational Optimization. More thorough discussions can be found in [1], [7].

To solve this optimization problem, we define three properties analogous to mass-moments of rigid bodies. The mass, first moment, and centroid of V_i are defined respectively as

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

and $C_{V_i} = L_{V_i} / M_{V_i}, \quad (3)$

Note that $\phi(q)$ strictly positive imply both $M_{V_i} > 0 \forall V_i \neq \emptyset$ and $C_{V_i} \in V_i \setminus \partial V_i$ (C_{V_i} is in the interior of V_i). Thus M_{V_i} and C_{V_i} have properties intrinsic to physical masses and centroids. Finally, it is useful to define the position error

$$e_i = (C_{V_i} - p_i). \quad (4)$$

A standard result in locational optimization is that

$$\frac{\partial \mathcal{H}}{\partial p_i} = -M_{V_i} e_i. \quad (5)$$

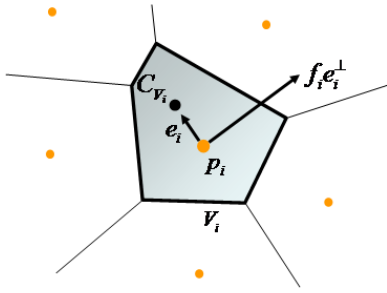


Fig. 2. A geometric representation of the non-adaptive ladybug controller is shown. The robot is at p_i , its Voronoi region is labelled V_i , with a centroid at C_{V_i} . The two components of the control law are shown as perpendicular vectors, e_i , the coverage force, and, $f_i e_i^\perp$, the exploration force.

Equation (5) implies that local minima of \mathcal{H} correspond to the configurations such that $p_i = C_{V_i} \forall i$, that is, each agent is located at the centroid of its Voronoi region. Such configurations are called centroidal Voronoi configurations, and it is known that several such configurations may exist. The optimization problem is to find a centroidal Voronoi configuration corresponding to the minimal value of \mathcal{H} .

III. NON-ADAPTIVE LADYBUG CONTROLLER

Let the robots have dynamics

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

where u_i is the control input. This may simply mean there is a low level controller in place to enforce integrator dynamics. Results can be extended to second order and nonholonomic dynamics. Such extension for the basic, non-adaptive controller are given in [1], and would likely apply straightforwardly to the ladybug controllers we discuss here.

We assume that the robots are able to compute their own Voronoi cell, $V_i = \{q \mid \|q - p_i\| \leq \|q - p_j\|\}$ by communicating their location among their Voronoi neighbors. Also, we require the follow assumption.

Assumption 1 (Sensory Function Knowledge): The sensory function $\phi(q)$ is available to all of the robots.

We can formulate the non-adaptive ladybug control law as

$$u_i = K e_i, \quad \text{where} \quad K = \begin{bmatrix} k & -f_i \\ f_i & k \end{bmatrix}, \quad (7)$$

and where $f_i \in \mathbb{R}$ is the exploration gain, and $k \in R_+$, is a control gain. A geometric schematic of the control law is shown in Fig. 2. The key attributes of the gain matrix K is that it is positive definite *and* it has a skew-symmetric component. The controller from [1] can be recovered simply by setting $f_i = 0 \forall i$.

Theorem 1 (Non-Adaptive Ladybug Convergence):

Under Assumption 1, for the system of agents with dynamics (6) and the control law (7),

$$\lim_{t \rightarrow \infty} e_i = 0 \quad \forall i \in \{1, \dots, n\} \quad (8)$$

Proof: Let \mathcal{H} be a Lyapunov-like function. Taking the time derivative of \mathcal{H} along the trajectories of the system yields

$$\dot{\mathcal{H}} = - \sum_{i=1}^n \left[M_{V_i} k e_i^T e_i + M_{V_i} f_i e_i^T e_i^\perp \right], \quad (9)$$

where e_i^\perp is the vector perpendicular to e_i of the same length. But $e_i^T e_i^\perp = 0$, resulting in $\dot{\mathcal{H}} \leq 0$. Also, the facts that u_i is continuous $\forall i$, \mathcal{H} has continuous first partial derivatives, \mathcal{H} is radially unbounded, and $\dot{\mathcal{H}} \leq 0$ imply that $\dot{\mathcal{H}}$ is uniformly continuous. Therefore, by Barbalat's lemma $\lim_{t \rightarrow \infty} \dot{\mathcal{H}} = 0$, which implies (8). ■

A. Ladybug Exploration

The exploration gain f_i can be chosen in a number of ways. In fact, it can be time varying and even stochastic (as long as it remains bounded). In any case, it is invisible to the convergence proof. Actual ladybugs use what appears to be a stochastic algorithm in which their turning frequency and amplitude increase with increased evidence of aphids. This causes them to stochastically spiral in and linger upon groups of aphids [2]. Furthermore, it has been found that individual ladybugs tend to show a significant bias toward turning either right or left [3]. These attributes might be captured in f_i in a variety of ways. We have chosen to use the most simplistic way as it tends to give good empirical performance despite its simplicity. In particular, we let $f_i = \pm f$ be a constant, the same magnitude for all i . We capture the right or left handedness of the ladybugs by assigning the sign of f_i randomly at initialization.

There is one subtle technicality that must be addressed. We desire that the robot trajectories remain inside Q for their entire trajectories. Indeed, this is not guaranteed by the convergence proof, but can be easily accomplished with a stable collision detection rule. Specifically, let

$$f_i(p_i) = \begin{cases} 0 & \text{if } p_i \in \partial Q \\ \pm f & \text{otherwise.} \end{cases} \quad (10)$$

This choice for f_i is bounded, therefore it does not affect the convergence result, and it is guaranteed to keep $p_i \in Q$. To see this, if p_i is on the boundary ∂Q , its velocity will be $\dot{p}_i = k(C_{V_i} - p_i)$. The centroid C_{V_i} , as was previously pointed out, is always in the interior of Q , thus p_i will be driven toward the interior preventing any escape from Q .

Unfortunately, for C_{V_i} to be on the interior of Q , Q must be convex (as was previously stated). This precludes environments with obstacles or "room and hallway" type structures. In practice the controller works well in some such environments and not well in others. Extending the controller to nonconvex environments is a topic of ongoing research [14].

B. Simulation Results

Simulations were carried out in a Matlab environment to compare the performance of the ladybug and basic controllers. The dynamics in (6) with the control law in (7)

with $k = 1$, for a group of $n = 10$ robots were integrated using a fixed time step solver. For the basic controller, the exploration gain $f_i = 0$, $\forall i$. For the ladybug controller, $f = 10$ was used with the sign determined randomly at initialization, and with the collision detection law described in (10). The region Q was taken to be the unit square. The sensory function $\phi(q)$ was constructed as a sum of two Gaussians,

$$\phi(q) = \sum_{j=1}^2 \frac{100}{\sigma_j \sqrt{2\pi}} \exp \left\{ -\frac{(q - \mu_j)^2}{2\sigma_j^2} \right\}, \quad (11)$$

where $\sigma_j = .18$. One of the Gaussian was centered at $\mu_1 = [1/6 \ 1/6]^T$ and the other was centered at $\mu_2 = [5/6 \ 5/6]^T$. The robots in the network were started from the same initial positions for the two controllers, confined to the lower-left $1/10$ square of Q . The spatial integrals in (2) required for the control law were approximated by discretizing each Voronoi region V_i into a 7×7 grid and summing contributions of the integrand over the grid. Voronoi polygons were computed for each robot using a distributed algorithm similar to the one described in [1]. We point out that each robot must compute its Voronoi region at each execution of the control loop, therefore controllers of this kind can be computationally expensive, though still well within the abilities of modern micro-controllers, as detailed in the robot experiments carried out in [10].

Figure 3 shows the results of numerical simulations for the ladybug controller (left column) and the basic controller (right column), with the Gaussian centers marked by red \times s. The figures show the explorative, spiralling behavior of the ladybug controller in comparison with the rather direct trajectories of the basic algorithm. The final configuration for the ladybug controller has an equal number of robots concentrated around the two Gaussians, whereas the basic controller has caused more robots to collect around the Gaussian close to where they started.

Figure 4 shows that both controllers cause the network to converge to a centroidal Voronoi configuration, as guaranteed by Theorem 1. However, Fig. 5 shows that the ladybug controller obtained a significantly lower cost configuration than the basic controller at a significantly faster convergence rate. This supports the assertion that exploration helps the ladybug controller to avoid local minima.

IV. ADAPTIVE LADYBUG CONTROLLER

The adaptive controller is significantly more complicated than its non-adaptive counterpart because it has to maintain a stable learning algorithm within the controller. Consider robots with the same dynamics as in (6).

We relax Assumption 1 with the following two less restrictive assumptions.

Assumption 2 (Sensory Function Measurement): The robots are able to measure $\phi(p_i)$. That is, they are equipped with sensors from which they can determine the value of ϕ at their own position p_i at any point in time.

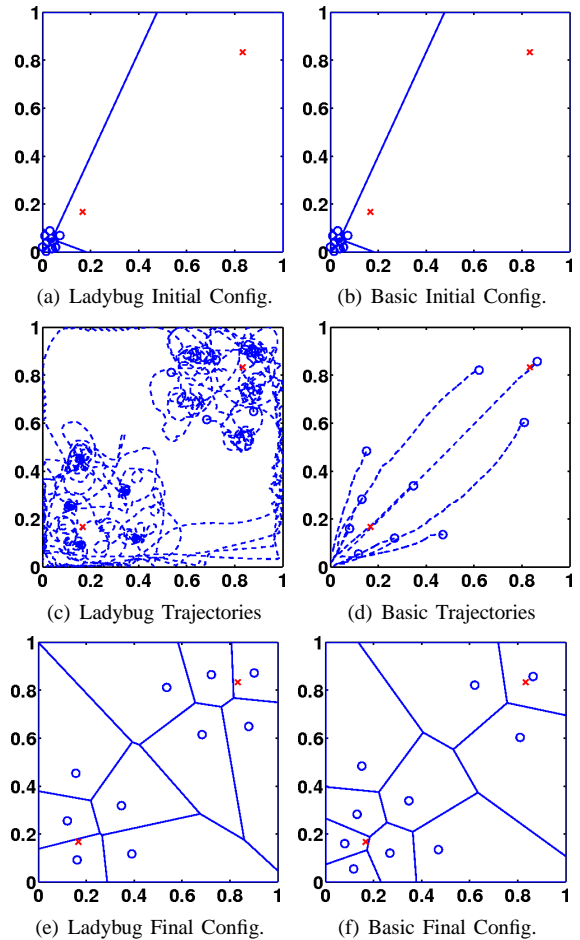


Fig. 3. The initial configuration, robot trajectories, and final configuration are shown for a network of robots with the non-adaptive ladybug controller on the left (3(a), 3(c), and 3(e)) and with the basic non-adaptive controller on the right (3(b), 3(d), and 3(f)). The Gaussian centers of $\phi(q)$ are marked by the red \times s.

Assumption 3 (Matching Conditions): $\exists a \in \mathbb{R}^m$ and $\mathcal{K} : Q \mapsto \mathbb{R}_+^m$, such that

$$\phi(q) = \mathcal{K}(q)^T a, \quad (12)$$

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

$$a(j) \geq a_{\min} \quad \forall j \in \{1, \dots, m\} \quad (13)$$

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

Requirements such as Assumption 3 are common for adaptive controllers. In theory, the assumption is not limiting since any function (with some smoothness requirements) over a bounded domain can be approximated arbitrarily well by a network of basis functions [15]. In practice, however, designing a suitable function approximation network requires application-specific expertise. We use Gaussian basis functions in our simulations, but there is a variety of other basis function families to choose from including, wavelets, sigmoids, and splines.

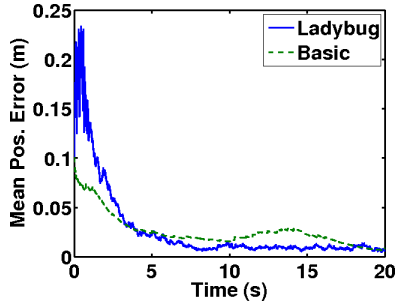


Fig. 4. The position error e_i averaged over the robots is shown for the ladybug and the basic non-adaptive controllers.

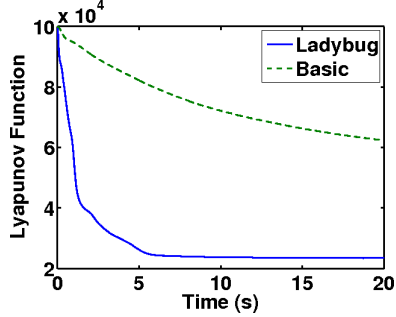


Fig. 5. The decreasing Lyapunov function is shown for both the ladybug and the basic non-adaptive controllers. The ladybug controller achieves a lower value.

Let $\hat{a}_i(t)$ be robot i 's approximation of the parameter vector. Naturally, $\hat{\phi}_i = \mathcal{K}(q)^T \hat{a}_i$ is robot i 's approximation of $\phi(q)$. Define the mass moment approximations

$$\hat{M}_{V_i} = \int_{V_i} \hat{\phi}_i dq, \quad \hat{L}_{V_i} = \int_{V_i} q \hat{\phi}_i dq, \quad (14)$$

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

Next, define the parameter error

$$\tilde{a}_i = \hat{a}_i - a, \quad (15)$$

and the sensory function error

$$\tilde{\phi}_i = \hat{\phi}_i - \phi = \mathcal{K}(q)^T \tilde{a}_i. \quad (16)$$

Also define the *estimated* position error vector

$$\hat{e}_i = (\hat{C}_{V_i} - p_i), \quad (17)$$

which is not to be confused with the actual position error defined previously, $e_i = (C_{V_i} - p_i)$. Most importantly, the robots do not know e_i , but they do know \hat{e}_i . Notice also that $\hat{a}_i = a$ implies $\hat{e}_i = e_i$.

Finally, in order to compress the notation, we introduce the shorthand $\mathcal{K}_i = \mathcal{K}(p_i(t))$ for the value of the basis function vector at the position of robot i , and $\phi_i = \phi(p_i(t))$ for the value of ϕ at the position of robot i . As previously stated in Assumption 2, robot i can measure ϕ_i with its sensors.

Consider a control law of a similar form to (7),

$$u_i = K \hat{e}_i, \quad (18)$$

with K defined as before. The parameters \hat{a}_i used to calculate \hat{e}_i are adjusted according to a set of adaptation laws which are introduced below.

Define two quantities,

$$\Lambda_i = \int_0^t w(\tau) \mathcal{K}_i(\tau) \mathcal{K}_i(\tau)^T d\tau, \quad (19)$$

and

$$\lambda_i = \int_0^t w(\tau) \mathcal{K}_i(\tau) \phi_i(\tau) d\tau. \quad (20)$$

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

Define another quantity

$$F_i = \frac{\int_{V_i} \mathcal{K}(q) (q - p_i)^T dq K \int_{V_i} (q - p_i) \mathcal{K}(q)^T dq}{\int_{V_i} \hat{\phi}_i dq}, \quad (21)$$

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

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

$$\dot{\hat{a}}_{\text{pre}_i} = -F_i \hat{a}_i - \gamma (\Lambda_i \hat{a}_i - \lambda_i), \quad (22)$$

$$\hat{a}_i = \Gamma(\hat{a}_{\text{pre}_i} - I_{\text{proj}_i} \hat{a}_{\text{pre}_i}) \quad (23)$$

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

$$I_{\text{proj}_i}(j) = \begin{cases} 0 & \text{for } \hat{a}_i(j) > a_{\min} \\ 0 & \text{for } \hat{a}_i(j) = a_{\min} \text{ and } a_{\text{pre}_i}(j) \geq 0 \\ 1 & \text{otherwise} \end{cases}, \quad (24)$$

where (j) denotes the j^{th} element for a vector and the j^{th} diagonal element for a matrix. Equations (23) and (24) implement a projection operation that prevents any element of \hat{a}_i from dropping below the lower bound a_{\min} . The controller and adaptation law from [6] can be recovered by setting $f_i = 0 \forall i$.

The terms in the adaptation law 22 have a natural interpretation. The term $-F_i \hat{a}_i$ compensates for uncertainty in the centroid position, and the term $-(\Lambda_i \hat{a}_i - \lambda_i)$ carries out a gradient descent to minimize the sensory function error $\tilde{\phi}_i(p_i)$ integrated over time. The projection is then required because the controller has a singularity at $\hat{M}_{V_i} = 0$. We could also add a consensus term to the adaptation law to speed convergence, as described in [16]. The controller and adaptation law cause the network of robots to converge to an advantageous configuration as formalized in the follow theorem.

Theorem 2 (Adaptive Ladybug Convergence): Under Assumptions 2 and 3, for the system of agents with dynamics (6) and the control law (18),

$$i) \quad \lim_{t \rightarrow \infty} \hat{e}_i = 0 \quad \forall i \in \{1, \dots, n\} \quad (25)$$

$$ii) \quad \lim_{t \rightarrow \infty} \mathcal{K}_i(\tau)^T \tilde{a}_i(t) = 0 \quad \forall \tau \mid w(\tau) > 0 \quad (26)$$

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

Proof: Let

$$\mathcal{V} = \sum_{i=1}^n \left(\int_{V_i} \frac{1}{2} \|q - p_i\|^2 \phi(q) dq + \frac{1}{2} \tilde{a}_i^T \Gamma^{-1} \tilde{a}_i \right). \quad (27)$$

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

$$\dot{\mathcal{V}} = - \sum_{i=1}^n \left[\hat{M}_{V_i} k \hat{e}_i^T \hat{e}_i + \hat{M}_{V_i} f_i \hat{e}_i^T \hat{e}_i^\perp + \tilde{a}_i^T I_{\text{proj}_i} a_{\text{pre}_i} + \gamma \int_0^t w(\tau) (\mathcal{K}_i^T(\tau) \tilde{a}_i(t))^2 d\tau \right], \quad (28)$$

For details of this derivation, please refer to [6]. As before $\hat{e}_i^T \hat{e}_i^\perp = 0$, resulting in

$$\dot{\mathcal{V}} = - \sum_{i=1}^n \left[\hat{M}_{V_i} k \hat{e}_i^T \hat{e}_i + \tilde{a}_i^T I_{\text{proj}_i} a_{\text{pre}_i} + \gamma \int_0^t w(\tau) (\mathcal{K}_i^T(\tau) \tilde{a}_i(t))^2 d\tau \right], \quad (29)$$

It can be shown that all terms inside the sum are positive, therefore $\dot{\mathcal{V}} \leq 0$. Also, the facts that u_i is continuous $\forall i$, \mathcal{V} has continuous first partial derivatives, \mathcal{V} is radially unbounded, and $\dot{\mathcal{V}} \leq 0$ imply that $\dot{\mathcal{V}}$ is uniformly continuous. Therefore, by Barbalat's lemma $\lim_{t \rightarrow \infty} \dot{\mathcal{V}} = 0$, which implies (8) from Theorem 2, and

$$\lim_{t \rightarrow \infty} \int_0^t w(\tau) (\mathcal{K}_i^T(\tau) \tilde{a}_i(t))^2 d\tau = 0 \quad \forall i \in \{1, \dots, n\}.$$

The integrand in (30) is non-negative, therefore it must converge to zero for all τ , which implies (26) from Theorem 2. ■

A. Persistent Excitation and Ladybug Exploration

The closeness of the learned sensory distribution to the true sensory distribution is intimately related to the extent to which a robot has explored the environment. Indeed, this was the main justification for introducing the ladybug force. The concept of persistent excitation (PE) from adaptive control captures this intuition precisely. In our case, conditions for PE fall out naturally, along with a metric for determining how explorative, or persistently exciting, a robot's trajectory is.

Formally, the assertion (26) of Theorem 2 states that the estimate of the sensory function $\hat{\phi}_i$ will converge asymptotically to the true sensory function ϕ for all points on the robot's trajectory with positive weighting $w(\tau)$. This does not, however, imply that $\hat{\phi}_i(q) \rightarrow \phi(q)$ everywhere in Q . This requires an extra condition.

Corollary 1 (Persistent Excitation): In addition to the requirements for Theorem 2, if for some $i \in \{1, \dots, n\}$

$$\int_0^t w(\tau) \mathcal{K}_i(\tau) \mathcal{K}_i^T(\tau) d\tau > 0, \quad (30)$$

then the following also occur

$$\lim_{t \rightarrow \infty} \hat{a}_i = a, \quad (31)$$

$$\lim_{t \rightarrow \infty} \hat{\phi}_i = \phi \quad \forall q \in Q, \quad (32)$$

$$\lim_{t \rightarrow \infty} e_i = 0. \quad (33)$$

Proof: Consider the last term in the sum from (29). Take the two \tilde{a}_i outside of the integral to give

$$\dot{\mathcal{V}} = - \sum_{i=1}^n \left(\dots + \gamma \tilde{a}_i^T \int_0^t w(\tau) \mathcal{K}_i(\tau) \mathcal{K}_i^T(\tau) d\tau \tilde{a}_i \right).$$

Since $\dot{\mathcal{V}} \rightarrow 0$, if $\int_0^t w(\tau) \mathcal{K}_i(\tau) \mathcal{K}_i^T(\tau) d\tau$ is positive definite for some i , $\tilde{a}_i \rightarrow 0$. Convergence of the sensory function (32) then follows directly from (16), and implies $\hat{C}_{V_i} \rightarrow C_{V_i}$ from (14), which in turn implies position error convergence (33). ■

What is more, even if PE is achieved, parameter convergence can be prohibitively slow. A useful metric for assessing parameter convergence rates is

$$\lambda_{\min_i}(t) = \text{mineig} \int_0^t w(\tau) \mathcal{K}_i(\tau) \mathcal{K}_i^T(\tau) d\tau, \quad (34)$$

where mineig denotes the minimum eigenvalue of a matrix. If this quantity is strictly positive, PE has been achieved, and the larger it is, the faster parameter convergence will take place. Intuitively, the more of Q visited by a robot, the larger λ_{\min_i} will be, and the faster parameter estimates will converge to the true parameters. The ladybug exploration gain f_i promotes exploration of the space and is therefore likely to produce trajectories that are *more* PE than the basic controller.

There is also an implicit trade-off between the richness of the basis function set and the difficulty of achieving a PE trajectory. For example, if $\mathcal{K}(q)$ consists of a single Gaussian function, all trajectories are PE. If $\mathcal{K}(q)$ consists of two Gaussian functions with radial symmetry, all trajectories are PE except those which are perpendicular to the line passing through the centers of the Gaussians. As the number of basis functions grows, the set of non-PE trajectories also grows. If the set of basis functions is too rich, it becomes very unlikely that PE will be achieved. Thus a balance must be found depending upon the requirements of the application. This is a manifestation of the well-known dichotomy of goodness-of-fit vs. generalizability.

B. Simulation Results

Simulations were carried out in the same environment as for the non-adaptive controllers, with the same initial configuration and exploration gain. The function $\phi(q)$ was constructed slightly differently to meet the requirements of Assumption 3. Specifically $\phi(q)$, was parameterized as a Gaussian network with 9 Gaussians and a constant offset. In particular, for $\mathcal{K} = [1 \quad \mathcal{K}(2) \quad \dots \quad \mathcal{K}(10)]^T$, each component, $\mathcal{K}(j)$ for $2 \leq j \leq 10$, was implemented as

$$\mathcal{K}(j) = \frac{1}{\sigma_j \sqrt{2\pi}} \exp \left\{ -\frac{(q - \mu_j)^2}{2\sigma_j^2} \right\}, \quad (35)$$

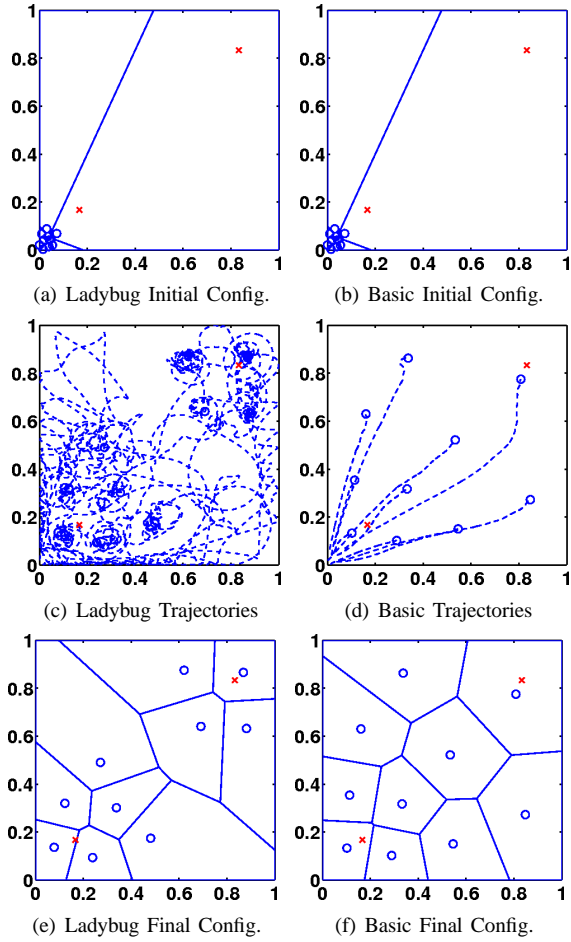


Fig. 6. The initial configuration, robot trajectories, and final configuration are shown for a network of robots with the adaptive ladybug controller on the left (6(a), 6(c), and 6(e)) and with the basic adaptive controller on the right (6(b), 6(d), and 6(f)). The Gaussian centers of $\phi(q)$ are marked by the red x's.

where $\sigma_j = .18$. The unit square was divided into an even 3×3 grid and each μ_j was chosen so that one of the 9 Gaussians was centered at the middle of each grid square. The parameters were chosen as $a = [a_{\min} \ 100 \ a_{\min} \ \dots \ a_{\min} \ 100]^T$, 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 essentially identical to the one used previously.

Each robot used a copy of the 9 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 $k = 1$, $\Gamma = I_{10}$, and $\gamma = 5$. The data weighting function was chosen to be $w(t) = \|\dot{p}_i\|^2$. In all other respects the simulations were identical to those described previously.

Figure 6 shows the results of numerical simulations for the adaptive ladybug controller (left column) and the basic controller (right column), with the Gaussian centers marked by red x's. As in the non-adaptive case, the figures show the spiraling behavior of the ladybug controller. Evidently, the

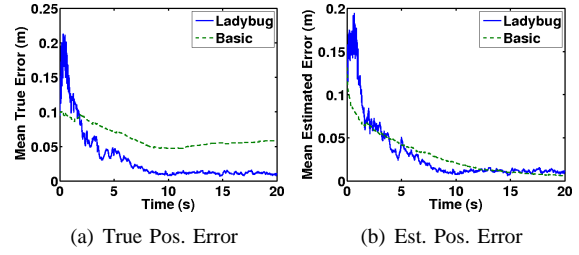


Fig. 7. The true position error e_i is shown on the left and the estimated position error \hat{e}_i on the right averaged over all of the robots for the ladybug and the basic adaptive controllers. The convergence of \hat{e}_i is guaranteed by Theorem 2 for both controllers, however the ladybug controller leads to a lower true position error.

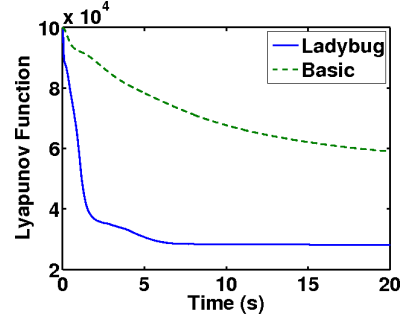


Fig. 8. The decreasing Lyapunov function is shown for both the ladybug and the basic adaptive controllers. The ladybug controller achieves a lower value indicating a better sensing configuration and a better sensory function approximation.

ladybug controller finds a better final configuration for sensing over the bi-modal distribution than the basic controller. Recall that the robots had no knowledge of the distribution before hand, but each robot learned an approximation of it during its trajectory.

The right side of Fig. 7 illustrates that the estimated position error converges to zero for both controllers, as asserted in Theorem 2. However, the true error (shown in the left of Fig. 7) indicates that the ladybug controller converged to a truly centroidal Voronoi configuration and the basic controller did not. Also, Fig. 8 shows that the ladybug controller obtained a lower Lyapunov function than the basic controller at a faster convergence rate, indicating both a

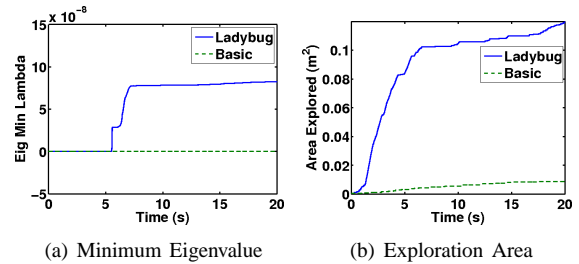


Fig. 9. The left plot shows a metric indicating the richness of the robot trajectories. A larger value denotes faster parameter convergence and also indicates more exploration. The ladybug controller gives richer trajectories than the basic controller. The plot on the right shows the minimum over all the robots of the area explored during a trajectory. Again, the ladybug controller causes the robots to explore more of the space than the basic one.

lower-cost configuration and a better function approximation.

Finally, Fig. 9 compares the richness of robot trajectories for the two controllers. The minimum over all the robots of λ_{\min_i} (as defined in (34)) is shown for the two controllers on the left (Fig. 9(a)). With the adaptive controller, PE was achieved for every robot in the network, thus satisfying the conditions for Corollary 1. For the basic controller, at least one of the robots (in fact 8 of them, though this is not evidenced in the plot) did not have PE trajectories, and therefore did not gather enough information to learning the true sensory distribution function. Also, the minimum over the robots of the area explored by a robot is shown on the right (Fig. 9(b)). This was computed by dividing the area Q into a 50×50 grid and summing up the grid areas visited by a robot. Even the least exploring robot visits 12% of the space with the ladybug controller, while for the basic controller, the least exploring robot visits less than 1% of the space.

V. CONCLUSION

In this work we proposed a decentralized controller for causing a network of robots to explore and cover an area. The exploration was inspired by the strategy used by ladybugs to hunt for aphids. A non-adaptive and an adaptive version of the controller was presented, with the ladybug exploration force appearing as an addition to a basic controller in each case. Convergence of the controller was proven with a Lyapunov-type proof. In both the adaptive and non-adaptive cases, the ladybug controller was shown to perform better than the basic controller. For the non-adaptive controller, this was attributed to the ladybug controller's ability to avoid locally optimal configurations. For the adaptive controller this was explained by showing that exploration leads to richer trajectories, causing better parameter learning, and eventually better coverage. In the future we would like to carry out experimental investigations with this control strategy on robot platforms, such as those used in [10]. Also, it would be interesting to look for adaptation rules to allow the positions and number of basis functions to adapt based on sensor measurements. We hope that the combination of animal behaviors with control theoretic concepts can lead to more creative and effective control strategies in the future.

REFERENCES

- [1] J. Cortés, S. Martínez, T. Karatas, and F. Bullo, "Coverage control for mobile sensing networks," *IEEE Transactions on Robotics and Automation*, vol. 20, no. 2, pp. 243–255, April 2004.
- [2] P. Kareiva and G. Odell, "Swarms of predators exhibit "preytaxis" if individual predators use area-restricted search," *The American Naturalist*, vol. 130, no. 2, pp. 233–270, August 1987.
- [3] R. Girling, M. Hassall, and J. Turner, "Do turning biases by the 7-spot ladybird, *Coccinella Septempunctata*, increase their foraging efficiency?" *Behaviour*, vol. 144, no. 2, pp. 143–163, February 2007.
- [4] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, no. 4598, 1983.
- [5] V. Cerny, "A thermodynamical approach to the travelling salesman problem: an efficient simulation algorithm," *Journal of Optimization Theory and Applications*, vol. 45, no. 1, pp. 41–51, 1985.
- [6] M. Schwager, J.-J. Slotine, and D. Rus, "Decentralized, adaptive control for coverage with networked robots," in *Proceedings of International Conference on Robotics and Automation*, Rome, April 2007.

- [7] Z. Drezner, *Facility Location: A Survey of Applications and Methods*, ser. Springer Series in Operations Research. New York: Springer-Verlag, 1995.
- [8] S. Salapaka, A. Khalak, and M. A. Dahleh, "Constraints on locational optimization problems," in *Proceedings of Conference on Decision and Control*, Maui, Hawaii, USA, December 2003.
- [9] S. Martínez, J. Cortés, and F. Bullo, "Motion coordination with distributed information," *IEEE Control Systems Magazine*, vol. 27, no. 4, pp. 75–88, 2007.
- [10] M. Schwager, J. McLurkin, and D. Rus, "Distributed coverage control with sensory feedback for networked robots," in *Proceedings of Robotics: Science and Systems*, Philadelphia, PA, August 2006.
- [11] J.-J. E. Slotine and W. Li, *Applied Nonlinear Control*. Upper Saddle River, NJ: Prentice-Hall, 1991.
- [12] K. S. Narendra and A. M. Annaswamy, *Stable Adaptive Systems*. Englewood Cliffs, NJ: Prentice-Hall, 1989.
- [13] P. A. Ioannou and J. Sun, *Robust Adaptive Control*. Englewood Cliffs, NJ: Prentice-Hall, 1996.
- [14] A. Ganguli, J. Cortés, and F. Bullo, "Maximizing visibility in nonconvex polygons: nonsmooth analysis and gradient algorithm design," in *Proceedings of the American Control Conference*, Portland, OR, June 2005, pp. 792–797.
- [15] R. Sanner and J. Slotine, "Gaussian networks for direct adaptive control," *IEEE Transactions on Neural Networks*, vol. 3, no. 6, 1992.
- [16] M. Schwager, J.-J. Slotine, and D. Rus, "Consensus learning for distributed coverage control," in *Proceedings of International Conference on Robotics and Automation*, Pasadena, CA, May 2008, to appear.