Learning Coverage Control of Mobile Sensing Agents in One-Dimensional Stochastic Environments

Jongeun Choi and Roberto Horowitz
Department of Mechanical Engineering
Michigan State University
Technical Report: CRL 2009-7-11

Jongeun Choi (corresponding author) is with the Department of Mechanical Engineering, Michigan State University jchoi@egr.msu.edu
Roberto Horowitz is with the Department of Mechanical Engineering, University of California at Berkeley horowitz@me.berkeley.edu

Abstract

This paper presents learning coverage control of mobile sensing agents without a priori statistical information regarding random signal locations in a one-dimensional space. In particular, the proposed algorithm controls the usage probability of each agent in a network while simultaneously satisfying an overall network formation topology. The distributed learning coordination algorithm is rather direct, not involving any identification of an unknown probability density function associated to random signal locations. The control algorithm will be synthesized based on diffeomorphic function learning with kernels. The almost sure convergence properties of the proposed coordination algorithm are analyzed using the ODE approach. Numerical simulations for different scenarios demonstrate the effectiveness of the proposed algorithm.

I. Introduction

Coordination of mobile autonomous agents and distributed mobile sensor networks have increasingly drawn the attention of engineers and scientists [1], [2], [3], [4], [5], [6], [7]. Mobile sensing agents form an ad-hoc wireless communication network in which each agent operates usually under a short communication range, a limited memory storage, and limited computational power. Sensing agents are often spatially distributed in an uncertain surveillance environment, can sense, communicate, and take control actions locally. To perform various tasks such as exploration, surveillance, and environmental monitoring, a group of mobile sensing agents require distributed coordination to adapt to unknown and non-stationary environments for a global goal. Cortes and Bullo [2] designed and analyzed a collection of such distributed control laws for mobile sensors whose closed-loop systems are nonsmooth gradient systems. One of the challenging problems in the coordination of sensing agents is to allocate coverage regions to agents optimally, which will be referred to as the coverage control problem. In [3], [4], a distributed coordination algorithm for sensing agents was derived and analyzed based on the classic Lloyd algorithm [8], which requires the knowledge of the probability density function associated to random signal locations. However, in practice, the exact knowledge of a statistical distribution of signal locations including its support may not be available a priori. This coverage control strategy is extended by [9] using a deterministic adaptive control approach assuming that the true density related to the cost function can be measured by sensing agents. Dynamic vehicle routing problems were studied in [6], [7], in which mobile agents in a fixed known convex region must visit event points generated by an (unknown) spatial-temporal Poisson point process. Arsie and Frazzoli [7] introduced strategies to minimize the expected time between the appearance of a target point and the time it is visited by one of the agents. The policy was similar to the MacQueen's [10] learning vector quantization algorithm thus, it does not rely on the knowledge of the underlying stochastic process.

Due to recent advances in micro-electro-mechanical systems (MEMS) technology [11], each agent can afford a particular set of sensors among different types such as acoustic, vibration, acceleration, infrared, magnetic, temperature, and biochemical sensors. Sensing agents in a sensor network are able to locally carry out simple computations to fuse collected measurements for the goal of the sensor network [11]. Measurements from heterogeneous sensors in different locations will provide statistically rich information in the sense of redundancy and complementarity [12], [13]. Such collective measurements along with multisensor fusion algorithms [14] will improve the performance of the sensor network significantly regarding estimation, prediction and tracking of a process of interest. A process with higher dimensional features can only be detected by using complimentary multiple sensors in different locations, rather than using a single sensor

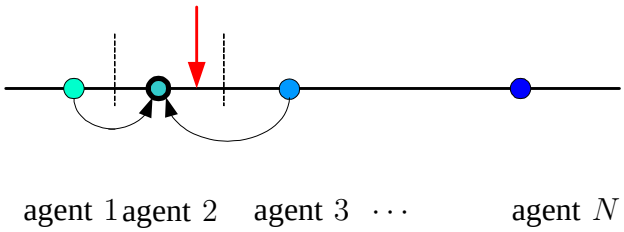

Fig. 1. Mobile agents with heterogenous sensors are distributed on a one-dimensional stochastic environment. Agent 2 measures a signal (denoted by a red arrow) in its coverage region while neighboring agents (agents 1 and 3) are supporting agent 2 and/or transmitting complimentary measurements of the same signal to agent 2.

[11], [12]. Equipping every robot with every sensor for possible signals and every localization capability may be too expensive. In [15], heterogeneous robots with different configurations use their special capabilities collaboratively to accomplish localization and mapping tasks.

Motivated by aforementioned research trends, we propose a class of self-organizing sensing agents with the following properties. First, a network of sensing agents should perform the coverage control without the statistical knowledge of random signal locations. Second, frequencies of random events or signal occurrences covered by agents are to be controlled according to each agent's limited capability and resources. To this end, we introduce a concept of the usage frequency of an agent, which will be referred to as *the usage probability* of the agent. Finally, the formation topology of the sensor network should be controlled so that each sensing agent can select specific neighbors equipped with functionally complementary sensor configurations to its own configuration.

Hence, this paper tackles the learning coverage control problem for agents to achieve a *desired usage probability* in one-dimensional stochastic environments without a priori statistical information regarding random signal locations. In particular, we present a new learning algorithm that controls the usage probability of each agent in a network while simultaneously satisfying an overall network formation order with respect to the signal locational space. The distributed coordination algorithm is rather direct, not involving any identification of an unknown probability density function associated to random signal locations. The proposed approach stems from the unsupervised kernel learning algorithm developed by the authors [16]. Based on this new approach, surprisingly, the learning coordination problem can be turned into a problem of designing a recursive diffeomorphic function learning algorithm using kernels. In this paper, as illustrated in Section V, we have modified the learning algorithm developed in [16] such that the agent's *usage probability estimates* are not necessary for the coordination algorithm, making the new algorithm distributed and more efficient.

Our approach can be illustrated by the following scenario. A network of mobile sensing agents with a limited communication range observes signals or events occurring at stationary random points on the one-dimensional space. The one-dimensional stochastic environment is illustrated by the solid line in Fig. 1. The probability density function that generates random signal locations is *not known* a priori. A sensing agent observes a signal if the signal's location is close to it as compared to neighboring agents. As more and more signals are observed, the mobile agents need to locally and iteratively adjust their relative positions over time so that asymptotically, each sensing agent ends up observing a desired fraction of the signals and therefore, dealing with a desired fraction of necessary tasks for the observed signals. In addition, the formation topology of the network will be organized in order for agents to utilize heterogeneous sensors.

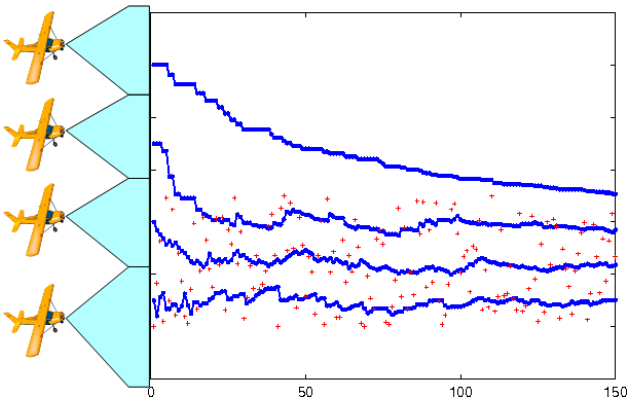

Fig. 2. A group of UAVs with a unit speed is scanning an environment for possible events. Kohonen's self-organizing map (SOM) algorithm in Eqs. (3) and (4) is used for this adaptive coverage control. Trajectories of agents are plotted in blue lines. Red crosses denote locations of events. The initial positions of agents are given by $q(0) = [1\,4\,7\,10]^T$.

For instance, as depicted in Fig. 1, agent 2 measures a signal (denoted by a red arrow) in its coverage region while neighboring agents (agents 1 and 3) are supporting agent 2 and/or relaying complimentary measurements of the same signal (from complimentary sensors) to agent 2.

This paper is organized as follows. We start with motivating examples in Section II. Here we explain how our approach was initiated. In Section III, we formulate a learning coverage control problem that achieves the desired usage probability of each sensing agent without a priori statistical knowledge of random signal locations. Section IV explains our scheme in which the adaptive network converges to a reference diffeomorphism which can be parameterized using local kernels. Section V describes learning coverage control for sensing agents. The main result for convergence analysis is presented in Section VI. In Section VII, we apply the new algorithm to different scenarios to demonstrate the usefulness of the proposed scheme.

Standard notation is used throughout the paper. Let $\mathbb{R}, \mathbb{R}_{\geq 0}, \mathbb{R}_{>0}, \mathbb{Z}, \mathbb{Z}_{\geq 0},$ and $\mathbb{Z}_{>0}$ denote, respectively, the set of real, non-negative real, positive real, integer, non-negative integer, and positive integer numbers. The positive definiteness (respectively, semi-definiteness) of a matrix A is denoted by $A \succ 0$ (respectively, $A \succeq 0$). The relative complement of a set A in a set B is denoted by $B \setminus A := B \cap A^c$, where A^c is the complement of A. The derivative of a column vector $y \in \mathbb{R}^m$ with respect to a column vector $x \in \mathbb{R}^n$ is defined by the matrix $\frac{\partial y}{\partial x} \in \mathbb{R}^{n \times m}$ whose (i,j)-th entry is given by $\frac{\partial y_i}{\partial x_j}$. Other notation will be explained in due course.

II. Motivating Examples

We introduce algorithms for the following scenario. As shown in Fig. 2, a group of unmanned aerial vehicles (UAVs) with a unit speed is scanning a two-dimensional space for possible events or targets of interest. In each iteration, an event occurs at a stochastic location with some probability density function (pdf). The set of identities of agents is denoted by $\mathcal{I} = \{1, \cdots, N\}$. Let $q_\gamma(t)$ be the location of agent γ at time t. The collection of agent's locations is denoted by

$$q(t) = [q_1(t) \cdots q_N(t)]^T \in \mathbb{R}^N. \tag{1}$$

Each sensing agent will detect signals and the corresponding location $u(t) \in \mathcal{U}$ at time index t over the surveillance region in charge. We assume that the agent γ takes charge of measuring

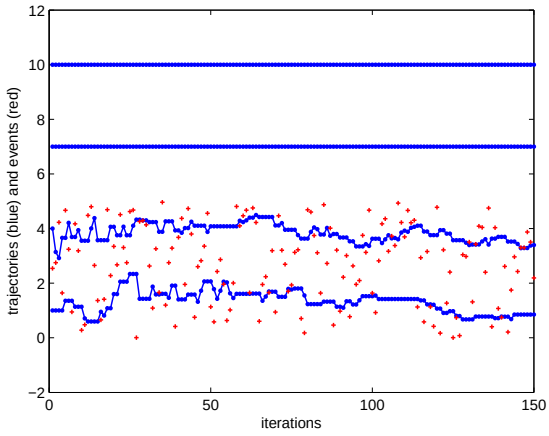

Fig. 3. A learning vector quantization (LVQ) algorithm in Eqs. (3) and (5) is used for this adaptive coverage control. Trajectories of agents are plotted in blue lines. Red crosses denote locations of events. The initial positions of agents are given by $q(0) = [1\,4\,7\,10]^T$.

signals and getting necessary tasks done in its *coverage region* R_γ determined by the nearest neighbor rule [3], [4]. The coverage region R_γ is given by the Voronoi cell [17] of agent γ

$$R_\gamma := \{u \in \mathcal{U} \mid |u - q_\gamma| \leq |u - q_i|,\, \forall i \neq \gamma\}, \tag{2}$$

where $|\cdot|$ is the Euclidean norm, and u is the location of the signal. For the given configuration $q(t)$ and the signal location $u(t)$, the *winning index* $w(\cdot,\cdot) : \mathbb{R}^N \times \mathcal{U} \to \mathcal{I}$ is defined by

$$w(q(t), u(t)) := \arg\left\{\min_{i\in\mathcal{I}} |u(t) - q_i(t)|\right\}. \tag{3}$$

When there are multiple minimizers in Eq. (3), the function will select the smallest index. Throughout the paper, the winning index in Eq. (3) will be often written as $w(u(t))$, or $w(t)$ for notational simplicity in different contexts. Each of UAVs should detect events occurring in its coverage region for optimizing their limited resources.

This adaptive coverage problem may be addressed using unsupervised competitive learning algorithms [10], [18], [19], [20]. Using Kohonen's self-organizing map (SOM) algorithm [20], [21] directly positions of agents can be updated by

$$q_i(t+1) = q_i(t) + \epsilon(t)\mathcal{K}(i, w(t))(u(t) - q_i(t)),\, \forall i \in \mathcal{I}, \tag{4}$$

where $\mathcal{K}(\cdot,\cdot)$ is a kernel function (e.g., a radial basis function) and it returns a smaller value as index i moves further away from the winning index $w(t)$. $\epsilon(t) > 0$ is a monotonically decreasing sequence such that $\epsilon(t) \to 0$ as $t \to \infty$. As can be seen in Fig. 2, the formation order[1] is preserved during the operation. Moreover, agents cover the support of the locational pdf, which tends to manage the limited resources efficiently. The almost sure convergence of one-dimensional Kohonen's algorithm with a class of nonuniform stimuli distributions was proved by Sadeghi [22] using the stability results on cooperative dynamical systems developed by Hirsch [23]. However, it is not clear what is actually optimized during the operation of this algorithm.

[1]Kohonen's algorithm guarantees either an orientation preserving or a reversing feature map [20].

A cost function (e.g., quantization error variance) to be minimized can be identified [18], [19] when we replace the kernel function $\mathcal{K}(\cdot,\cdot)$ in Eq. (4) with the Kronecker delta function:

$$\delta_{ij} = \begin{cases} 1, & \text{if } i = j; \\ 0, & \text{if } i \neq j. \end{cases}$$

In this case, the following learning vector quantization (LVQ) algorithm can be obtained:

$$q_i(t+1) = q_i(t) + \epsilon(t)\delta_{i,w(t)}(u(t) - q_i(t)),\ \forall i \in \mathcal{I}. \tag{5}$$

However, it is also well known that this LVQ algorithm suffers from local minima and so called unused "dead" agents [18], [19]. Fig. 3 shows trajectories of agents under this LVQ algorithm. As shown in Fig. 3, two agents (agents 3 and 4) that started initially away from the support of the pdf ($[0,\ 5]$), are not attracted to the support of the pdf at all. Hence, two agents will not be utilized for all future time. A way to avoid these "dead" agents is to change the problem by assuming the complete knowledge of the support, which however defeats the purpose of this paper. In the next section, we cope with this dilemma by introducing a concept of the usage probability of an agent.

III. Problem Statement

We refer to the environment in which events or signals of interest occur, as the signal locational space given by $\mathcal{U} = (u_{min}, u_{max}) \subset \mathbb{R}$. Suppose that agent γ can collect time samples of signal locations in R_γ, generated by the stationary random process $u : \mathbb{Z}_{>0} \to \mathcal{U}$ with an associated pdf $f_U : \mathcal{U} \to \mathbb{R}_{>0}$. The sequence $w(t)$ in Eq. (3) is then a random sequence with a discrete probability distribution that is induced by the pdf f_U. A vector $p = [p_1 \cdots p_N]^T \in \mathbb{R}^N_{\geq 0}$ is referred to as the *usage probability distribution* whose elements are defined by the *usage probabilities* of agents:

$$\begin{aligned} p_\gamma &:= \int_{u \in R_\gamma} f_U(u)du \\ &= \int_{u \in \mathcal{U}} \delta_{\gamma, w(q,u)} f_U(u)du \\ &= \Pr[w(q, u(t)) = \gamma] \geq 0,\ \forall \gamma \in \mathcal{I}, \end{aligned} \tag{6}$$

where $\sum_{\gamma=1}^N p_\gamma = 1$. Hence, p_γ provides a direct measure of how frequently agent γ is being used. For instance, agents 3 and 4 in Fig. 3 are not utilized at all since $p_3 = p_4 = 0$ for all time.

Assume that there exists a cost function J of the sensor network, which may be related to the functional lifetime, limited resources, and/or capabilities of sensing agents. Suppose that $J : \mathbb{R}^N_{\geq 0} \to \mathbb{R}_{\geq 0}$ is a function of the usage probability distribution p. Then, the optimal (or target) usage probability distribution that minimizes J, will be denoted by

$$p^o = [p^o_1 \cdots p^o_N]^T \in \mathbb{R}^N_{>0}, \qquad \sum_{\gamma=1}^N p^o_\gamma = 1. \tag{7}$$

p^o will be generated by Eq. (6) with the pdf f_U and optimal locations[2] of agents, denoted by $\bar{q}^o = [q^o_1 \cdots q^o_N]^T \in \mathcal{U}^N$. The problem is then formulated as follows.

[2]The map from a signal location u to the location of the responsible sensing agent q_w can be viewed as a *quantizer* in quantization theory [8]. p^o must be produced by a quantizer with an optimal location codebook vector $\bar{q}^o$.

Problem: Consider the random variable of signal locations generated by a pdf f_U. We assume that f_U is continuous and its support is connected and finite. However, f_U and its support are not known a priori. Assume that agents sense signals and their locations based on the nearest neighbor rule in Eq. (2). For a given p^o in Eq. (7), design a learning coverage algorithm that coordinates sensing agents to achieve the following asymptotic properties:

$$\begin{aligned} &\lim_{t\to\infty} p_\gamma(q(t)) = p^o_\gamma, \quad \forall \gamma \in \mathcal{I}, \\ &\text{subject to} \quad \lim_{t\to\infty} q_1(t) < \lim_{t\to\infty} q_2(t) < \cdots < \lim_{t\to\infty} q_N(t). \end{aligned} \tag{8}$$

Remark 1: The constraint on the *formation order* of agents in Eq. (8) will predecide the neighbors of each agent, since some agents prefer particular agents to be its neighbors among agents equipped with heterogeneous sensors. For a given decoder, the optimal encoder is the nearest neighbor rule [8]. Hence, if we set a codebook of a quantizer to the location vector q that satisfies Eq. (8), the nearest neighbor winning rule in Eq. (3) achieves the minimum variance distortion with respect to the stochastic signal locations $u(t)$ [8]. The variance distortion was often used for measuring the performance of a sensor network [3], [4]. In general, the centroid condition [8] may not be satisfied by our formulation since it may conflict with p^o.

IV. Diffeomorphism Learning with Kernels

In this section, we explain a diffeomorphism that maps a domain that contains the indices of agents to the signal locational space. This map plays a central role in the sense that it provides a structure in our learning coverage algorithm. We introduce a fictitious random sequence $x : \mathbb{Z}_{>0} \to \mathcal{X} \subset \mathbb{R}$, where $\mathcal{X} = (x_{\min}, x_{\max})$ is a *specified* finite open interval. We set $\mathcal{X} = (1/2, N+1/2)$, so that $\mathcal{I} \subset \mathcal{X}$. Let $u : \mathcal{X} \to \mathcal{U}$ be a mapping from $\mathcal{X}$ to the locational space $\mathcal{U}$. We now assume that u is actually a diffeomorphic function of x, i.e., $u : \mathcal{X} \to \mathcal{U}$ is a differentiable bijection and has a differentiable inverse such that the time samples of the locational random variable, $u(t)$, are generated by $u(t) = u(x(t))$. Thus, the pdf of x, $f_X : \mathcal{X}_{\mathbb{R}} \to \mathbb{R}_{>0}$, is given by

$$f_X(x) = \left|\frac{du}{dx}\right| f_U(u(x)) \quad \text{for all } x \in \mathcal{X}. \tag{9}$$

The diffeomorphism $u : \mathcal{X} \to \mathcal{U}$, induces the pdf f_X from the unknown pdf f_U via Eq. (9). This map will be subsequently referred to as the reference or the target map. Since $u : \mathcal{X} \to \mathcal{U}$ is a diffeomorphism, the collection of optimal sensor locations in Eq. (8) becomes

$$\bar{q}^o = [q^o_1 \cdots q^o_N]^T = [u(1) \cdots u(N)]^T \in \mathbb{R}^N. \tag{10}$$

Suppose that the target map $u(x)$ can be obtained by solving an integral equation of the first kind with a known smooth scalar symmetric kernel $\mathcal{K}(\cdot,\cdot) : \mathcal{X}^e \times \mathcal{X}^e \to \mathbb{R}_{\geq 0}$ and unknown influence coefficients c^o_ν's $\in \mathbb{R}$ that satisfy

$$u(x) = \sum_{\nu \in \mathcal{I}^e} \mathcal{K}(x,\nu)\, c^o_\nu, \tag{11}$$

$$\begin{aligned} \text{where } \mathcal{I}^e &= \{-(m-1), \cdots, N+m\}, \\ \mathcal{X}^e &= (-1/2-(m-1), N+m+1/2), \end{aligned} \tag{12}$$

for some integer $0 < m \ll N$. To obtain a distributed coordination algorithm, the support of the kernel has to be finite. We assume that the kernel in Eq. (11) is a radial basis function that

is (at least) $\mathcal{C}^1$ differentiable with a compact support. For instance, we may use a local kernel with a compact support proposed by Storkey [24].

$$\mathcal{K}(x,\nu) = \begin{cases} \frac{(2\pi-\Delta)(1+(\cos\Delta)/2)+\frac{3}{2}\sin\Delta}{3\pi}, & \text{for } \Delta < 2\pi, \\ 0, & \text{otherwise,} \end{cases}$$

where $\Delta = \beta|x-\nu|$, for $\beta > 0$. Then the resulting $u(x)$ is a $\mathcal{C}^1$ diffeomorphism. The elements of the vector

$$c^o \;\; = [c^o_{-(m-1)} \cdots c^o_{N+m}]^T \in \mathbb{R}^{N+2m}$$

are the *unknown* optimal influence coefficients that satisfy $q^o_\gamma = u(\gamma)$ for all $\gamma \in \mathcal{I}$. Hence, the logical approach to deal with our problem is to coordinate sensing agents according to the diffeomorphism function learning with kernels. The time varying outputs of the learning algorithm will directly update the locations of agents given as

$$q_\gamma(t) = q(\gamma, t), \qquad \forall\, \gamma \in \mathcal{I}^e, \tag{13}$$

where $q(x,t)$ is produced by the estimates of influence coefficients $\hat{c}_\nu(t)$:

$$q(x,t) = \sum_{\nu \in \mathcal{I}^e} \mathcal{K}(x,\nu)\, \hat{c}_\nu(t), \text{for all } x \in \mathcal{X}^e. \tag{14}$$

Here $\{q(x,t)\}$ in Eq. (14) is a parameterized family of smooth functions that contains the diffeomorphism of interest in Eq. (11). For given time t, we define the extended locational space $\mathcal{U}^e$ by the union of $\mathcal{U}$ and the range space of $q(x,t)$ (denoted by $\mathscr{R}(q(\mathcal{X}),t)$), i.e., $\mathcal{U}^e := \mathcal{U} \cup \mathscr{R}(q(\mathcal{X},t))$. We define the influence coefficient estimate vector by

$$\hat{c}(t) := [\hat{c}_{-(m-1)}(t) \cdots \hat{c}_{N+m}(t)]^T \in \mathbb{R}^{N+2m}. \tag{15}$$

Eq. (13) can then be rewritten as

$$q^e(t) := [q_{-(m-1)} \cdots q_{N+m}(t)]^T = K\,\hat{c}(t) \in \mathbb{R}^{N+2m}, \tag{16}$$

where $K \in \mathbb{R}^{(N+2m)\times(N+2m)}$ is the kernel matrix with (i,j) element $K_{i,j} := \mathcal{K}(i-m, j-m)$, which must be rank $N+2m$. For the function $q(x,t)$ in Eq. (14) to converge to an orientation preserving diffeomorphism, it is necessary to have

$$\lim_{t\to\infty} q'(x,t) = \lim_{t\to\infty} \frac{\partial}{\partial x} q(x,t) > 0.$$

Define the vector of partial derivatives of $q(x,t)$ with respect to x evaluated at $\gamma \in \mathcal{I}^e$ by

$$\begin{aligned} q'^e(t) &= \left[\; q'(-(1-m),t) \;\; \cdots \;\; q'(N+m,t) \;\right]^T \\ &= K'\hat{c}(t), \end{aligned} \tag{17}$$

where $K' \in \mathbb{R}^{(N+2m)\times(N+2m)}$ is the matrix whose (i,j) element is given by $K'_{i,j} := \partial\mathcal{K}(x,\lambda)/\partial x|_{x=i-m,\lambda=j-m}$. K (respectively K') is the collection of kernel vectors k_γ (respectively k'_γ) as defined by

$$\begin{aligned} K^T &= \left[\; k_{-(1-m)} \;\; \cdots \;\; k_{N+m} \;\right], \\ K'^T &= \left[\; k'_{-(1-m)} \;\; \cdots \;\; k'_{N+m} \;\right]. \end{aligned} \tag{18}$$

These local kernel vectors k_γ and k'_γ for $\gamma = 10$ are plotted in Fig. 4. Therefore, for $\gamma \in \mathcal{I}^e$, we have

$$\begin{aligned} q_\gamma(t) &= k_\gamma^T \hat{c}(t), \\ q'_\gamma(t) &= \frac{\partial q(x,t)}{\partial x}\Big|_{x=\gamma} = k_\gamma'^T \hat{c}(t). \end{aligned}$$

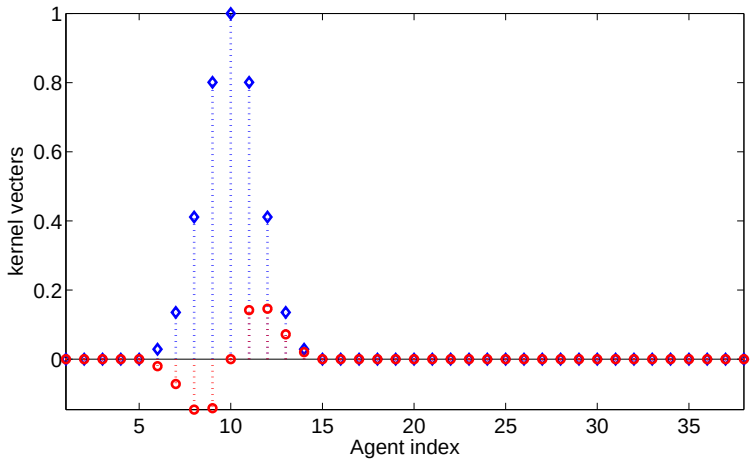

Fig. 4. Local kernel vectors k_γ (blue diamonds $\diamond$) and k'_γ (red circles $\circ$) for $\gamma = 10$.

V. Learning Coverage Control

The following discrete-time control system describes the motion of mobile agents:

$$q_i(t+1) = q_i(t) + \zeta_i(t), \tag{19}$$

where $q_i(t)$ and $\zeta_i(t)$ are, respectively, the position and the control of agent i at time index $t \in \mathbb{Z}_{\geq 0}$. For a sampled input location $u(t)$ at time t, the control of each sensing agent will have the form of

$$\zeta_i(t) = \alpha(t) k_i^T [-\delta\hat{c}_1(t) - \delta\hat{c}_2(t)], \, \forall i \in \mathcal{I}^e, \tag{20}$$

where $\alpha(t)$ is the monotonically decreasing gain sequence often used in stochastic approximation theory [25], [26] and it satisfies the following properties:

$$\alpha(t) > 0, \quad \sum_{t=1}^{\infty} \alpha(t) = \infty, \quad \sum_{t=1}^{\infty} \alpha^2(t) < \infty. \tag{21}$$

This gain sequence $\alpha(t)$ is introduced to guarantee that sensing agents converge to an optimal configuration in spite of stochastic locational signals. This sufficiently slow vanishing rate of the sequence is a key mechanism to ensure the almost sure convergence of states by slowly attenuating the effects of randomness [25], [26]. $\delta\hat{c}_1(t)$ and $\delta\hat{c}_2(t)$ in Eq. (20) will be provided shortly. To parameterize a family of slopes of $q(x,t)$ properly at the boundary of $\mathcal{X}$, agent 1 (respectively, agent N) needs to memorize and update the positions of fictitious agents $-(m-1)\cdots 0$ (respectively, agents $(N+1)\cdots(N+m)$) according to Eq. (20). These fictitious agents do not have sensors and only be passively updated by either agents 1 or N.

A. The Target Probability Tracking Law

We first define some notation. Let $\overline{\partial}x$ and $\underline{\partial}x$ be the indices associated to the extremum values of $\{q_\gamma \,|\, \gamma \in \mathcal{I}\}$ defined respectively by

$$\overline{\partial}x := \arg\left\{\max_{\gamma\in\mathcal{I}} q_\gamma\right\}, \quad \underline{\partial}x := \arg\left\{\min_{\gamma\in\mathcal{I}} q_\gamma\right\}. \tag{22}$$

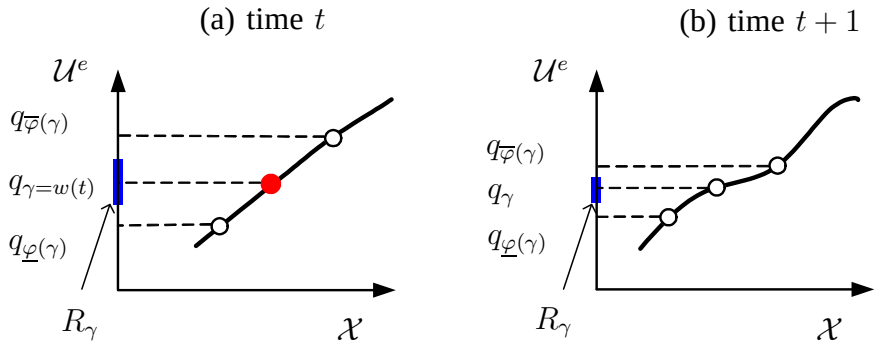

Fig. 5. The illustration of the target probability tracking law: (a) The configuration of agents at time t. The red dot denotes the location of the winning agent w at time t with respect to $\mathcal{X}$ and $\mathcal{U}^e$ spaces. (b) The updated configuration at time $t+1$. The learning rule decreases the slope of the function $q(x,t)$ at $x = w(t) = \gamma$ to reduce p_γ and the size of R_γ.

The indices of local neighbors in $\mathcal{U}^e$, $\overline{\varphi} : \mathcal{I} \to \mathcal{I}$ and $\underline{\varphi} : \mathcal{I} \to \mathcal{I}$ (see Fig. 5) are defined respectively by

$$\overline{\varphi}(w) := \arg \min_{\gamma \in \mathcal{I} \setminus \{w\}} \{|q_\gamma - q_w|\}, \text{ subject to } \quad (q_\gamma - q_w) \geq 0, \tag{23}$$

$$\underline{\varphi}(w) := \arg \min_{\gamma \in \mathcal{I} \setminus \{w\}} \{|q_\gamma - q_w|\}, \text{ subject to } \quad (q_w - q_\gamma) \geq 0. \tag{24}$$

The first term $\delta \hat{c}_1$ in Eq. (20) is designed for the usage probability $\{p_\gamma(t) \,|\, \gamma \in \mathcal{I}\}$ to track the target usage probability $\{p^o_\gamma \,|\, \gamma \in \mathcal{I}\}$. $\delta \hat{c}_1$ is given by

$$\delta \hat{c}_1(t) = \beta_1 \frac{\delta \hat{c}_1^{X(w(t))}}{p^o_{w(t)}}, \tag{25}$$

where $\beta_1 > 0$ is a gain and $p^o_{w(t)}$ is the target usage probability of the winning index w at time t given by Eqs. (3) and (7). The function $\delta \hat{c}_1^{X(\cdot)} : \mathcal{I} \to \mathbb{R}^{N+2m}$ is defined by

$$\delta \hat{c}_1^{X(w)} := \begin{cases} (k_{\overline{\varphi}(w)} - k_{\underline{\varphi}(w)})/2, & \text{if } w \in \mathcal{I} \setminus \{\overline{\partial} x, \underline{\partial} x\}, \\ (k_w + k_{\overline{\varphi}(w)})/2, & \text{if } w = \underline{\partial} x, \\ -(k_w + k_{\underline{\varphi}(w)})/2, & \text{if } w = \overline{\partial} x. \end{cases} \tag{26}$$

To demonstrate the mechanism of our learning algorithm, an illustration of the control action is provided as in Fig. 5. The configuration of agents at time t is depicted in Fig. 5-(a). The red dot denotes the location of the winning agent w at time t. Consider the map $q(\cdot, t) : \mathcal{X} \to \mathcal{U}^e$ defined in Eq. (14). Suppose that the slope of the function at $x = w(t)$ is positive and $w(t) \notin \{\overline{\partial} x, \underline{\partial} x\}$. The learning rule will then decrease the slope of the map $q(x,t)$ at $x = w(t) = \gamma$ in order to reduce the usage probability of agent γ (Fig. 5-(b)). This control action reduces the size of the coverage region R_γ for agent γ, which decreases the associated usage probability ($p_\gamma = \int_{R_\gamma} f_U(\sigma) d\sigma$) as well. Moreover, this control action is inversely proportional to p^o_γ as in Eq. (25) in order to guarantee that $p_\gamma(t) \to p^o_\gamma$ as $t \to \infty$.

Notice that the probability tracking law in Eq. (25) no longer requires the estimation of the usage probability $\{\hat{p}_\gamma \,|\, \gamma \in \mathcal{I}\}$ as compared to the learning algorithms derived in [16]. This enables us to develop the *distributed and efficient* learning algorithm.

B. The Orientation Preserving Update Law

The second term $\delta\hat{c}_2$ in Eq. (20) controls the orientation of the map $q(\cdot,t)$ in Eq. (14), and it is given by

$$\delta\hat{c}_2(t) := \beta_2\, k'_{w(t)}\, \left(\left|q'_{w(t)}\right| - q'_{w(t)}\right)\, \frac{\left[\text{sign}(q'_{w(t)}) - 1\right]}{p^o_{w(t)}}, \tag{27}$$

where $\beta_2 > 0$ is a gain and $q'_{w(t)} = q'(w(t),t)$ as defined in Eq. (17). $\text{sign}(\cdot)$ is defined by

$$\text{sign}(y) := \begin{cases} 1, & \text{if } y > 0, \\ 0, & \text{if } y = 0, \\ -1, & \text{if } y < 0. \end{cases}$$

To calculate $\delta\hat{c}_2(t)$ in Eq. (27), agent γ should update the slope of the map $q(x,t)$ at $x = \gamma$ and keep the updated slope for the next iteration. Hence, for agent γ, the proposed learning coordination algorithm is summarized as follows:

$$\begin{aligned} q_\gamma(t+1) &= q_\gamma(t) + \alpha(t) k_\gamma^T[-\delta\hat{c}_1(t) - \delta\hat{c}_2(t)], \forall\gamma \in \mathcal{I}^e, \\ q'_\gamma(t+1) &= q'_\gamma(t) + \alpha(t) {k'_\gamma}^T[-\delta\hat{c}_1(t) - \delta\hat{c}_2(t)], \forall\gamma \in \mathcal{I}. \end{aligned} \tag{28}$$

Remark 2: The proposed learning coverage control algorithm in Eq. (28) contains an update rule for $q'_\gamma(t)$, which is unique as compared to other algorithms [3], [10], [20].

C. The Distributed and Scalable Algorithm

In this section, we discuss the distributed nature of the algorithm. The learning algorithm can be either centralized or decentralized. Agent γ becomes a winner $w(t) = \gamma$ if an input signal occurs in its coverage region R_γ as defined in Eq. (2). In order to implement the tracking law in Eq. (25), the winning agent should determine whether its index is $\underline{\partial}x$, $\overline{\partial}x$, or neither of the two. This can be done by exchanging information between the winning agent and its neighbors in $\mathcal{U}^e$. If the winning agent is not surrounded by its two neighbors in $\mathcal{U}^e$, then the winning index w belongs to $\{\underline{\partial}x, \overline{\partial}x\}$. $\overline{\varphi}(w)$ and $\underline{\varphi}(w)$ in Eqs. (23) and (24) are local neighbors in $\mathcal{U}^e$. $q_{\underline{\varphi}(i)}$ (respectively, $q_{\overline{\varphi}(i)}$) is the lower (respectively, upper) closest neighbor to q_i, in which the closeness was meant in $\mathcal{U}^e$. All these operations are distributed. As shown in Eqs. (25) and (27), $\delta\hat{c}_1(t)$ and $\delta\hat{c}_2(t)$ contain either $k_{w(t)}$ or $k'_{w(t)}$. Notice that there exist positive numbers r_1, r_2, and r_3 (see Fig. 4) such that

$$\begin{aligned} r_1 &= \min\{r \mid k_\gamma^T k_w = 0,\ |\gamma - w| > r\,, \forall\gamma, w \in \mathcal{I}^e\}, \\ r_2 &= \min\{r \mid {k'_\gamma}^T k_w = 0,\ |\gamma - w| > r\,, \forall\gamma, w \in \mathcal{I}^e\}, \\ r_3 &= \min\{r \mid {k'_\gamma}^T k'_w = 0,\ |\gamma - w| > r\,, \forall\gamma, w \in \mathcal{I}^e\}. \end{aligned} \tag{29}$$

Eq. (29) shows that only the states of the winning agent and its neighboring indices in $\mathcal{I}$ are updated. The size of the neighborhood in $\mathcal{X}$ is determined by the support size of the kernel used in Eq. (11). For our adaptation rule, only local information about $w(t)$ and $q'_w(t)$, is needed as shown in Eqs. (25) and (27). Now we explain that the neighborhood in $\mathcal{X}$ is equivalent to the neighborhood in $\mathcal{U}^e$. Since $q(x,t)$ is a differentiable function with respect to x, and for a bounded $\hat{c}$, $|q'(x,t)| \le L$ for all $x_1, x_2 \in \mathcal{B}_{w,\delta} := \{x \in \mathcal{X} \mid |x - w| < \delta := \max\{r_1, r_2, r_3\}\}$, we have $|q(x_1,t) - q(x_2,t)| \le L|x_1 - x_2|$. Hence, if indices of agents in $\mathcal{X}$ belong to a ball $\mathcal{B}_{w,\delta}$, the

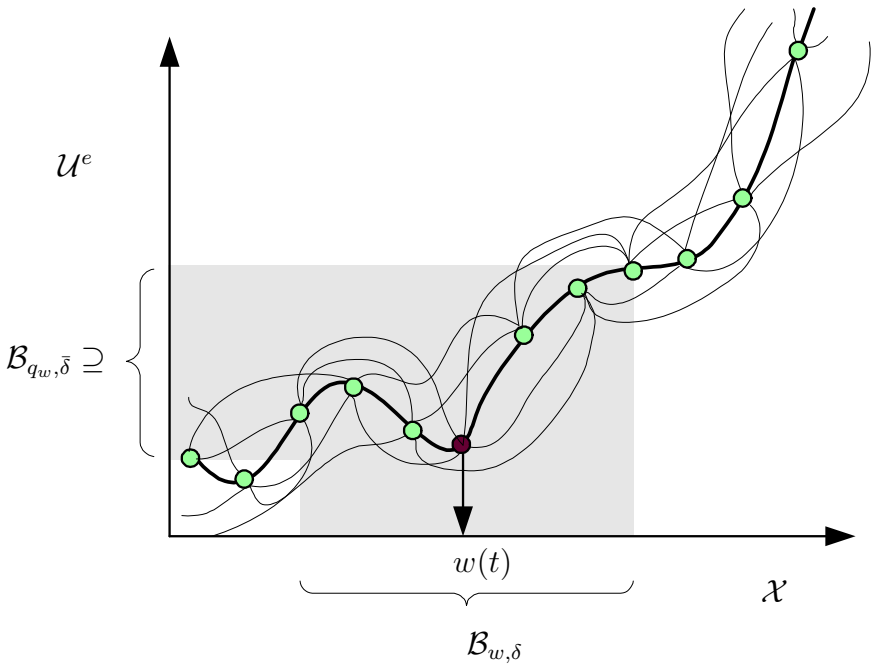

Fig. 6. The thick line represents the map $q(x,t)$ at a fixed time t. Circles represent distributed agents with respect to $\mathcal{U}^e$ and $\mathcal{X}$ spaces. Lines denote possible communication links between agents. The black dot represents the winning agent at time t. $\mathcal{B}_{w,\delta}$ is the neighborhood in $\mathcal{X}$ that needs an update. $\mathcal{B}_{q_w,\bar{\delta}}$ is the neighborhood in the physical space $\mathcal{U}^e$.

corresponding positions should then belong to the neighborhood $\mathcal{B}_{q_w,\bar{\delta}} := \{u \in \mathcal{U}^e \,|\, |u - q_w| < \bar{\delta}\}$, where information is actually communicated as illustrated in Fig. 6. We assume that the transmission range T_w can be adjusted according to the size of the image set of $\mathcal{B}_{w(x),\delta}$ under $q(x,t)$. T_w has to be adaptive since the length of a support of the signal locational distribution is not known a priori at the initial stage. After $q(x,t)$ achieves the correct formation ordering, i.e., $q(x,t)$ becomes an orientation preserving map, a vicinity in $\mathcal{X}$ implies a vicinity in $\mathcal{U}^e$, and vice versa. Each agent can then maintain a minimal transmission range. The learning coordination algorithm is scalable as the number of agents increases.

D. The Centralized Formulation

Since $q^e(t) = K\hat{c}(t)$ and K is bijective, it is easy to see that the overall dynamics of agents in Eq. (28) can be reformulated by

$$\hat{c}(t+1) = \hat{c}(t) + \alpha(t)[-\delta\hat{c}_1(t) - \delta\hat{c}_2(t)], \tag{30}$$

where $\hat{c}(t)$ is the influence coefficient estimate defined in Eq. (15). For convergence analysis, we will consider the learning coordination algorithm in the form of the centralized adaptation in Eq. (30).

VI. The Main Result

We use Ljung's ordinary differential equation (ODE) approach developed in [25], [27] to analyze the convergence properties of our new learning algorithm. Eq. (30) can be expressed as

$$\hat{c}(t+1) = \hat{c}(t) + \alpha(t)F(t, \hat{c}(t), u(t)), \tag{31}$$

where $F(t, \hat{c}(t), u(t)) := -\delta\hat{c}_1(\hat{c}(t), u(t)) - \delta\hat{c}_2(\hat{c}(t), u(t))$. The ODE associated to Eq. (31) is

$$\begin{aligned}\dot{\hat{c}}(\tau) &= f(\tau, \hat{c}(\tau)) = \mathbb{E}^{\hat{c}(\tau)}\{F(\hat{c}(\tau), u(t))\} \\ &= \int_{\mathcal{X}} F(\hat{c}(\tau), u(x)) f_X(x) dx,\end{aligned} \tag{32}$$

where $\hat{c}(\tau)$ is kept constant at the frozen time τ in the calculation of $\mathbb{E}^{\hat{c}(\tau)}\{F(\hat{c}(\tau), u(t))\}$. Two of the nontrivial sufficient conditions for the ODE [25] approach to be applicable are that $F(t, \hat{c}, u)$ must be Lipschitz continuous in $\hat{c}$ and u (B.3 in [25]), and the Lipschitz constants must be Lipschitz continuous in $\hat{c}$ and u (B.4 in [25]). These conditions are verified by the following Lemma.

Lemma 3: For the input signal u, let $w(\hat{c}, u)$ be the value determined by Eq. (3) and $q(x, t)$ that builds on $\hat{c}$ as in Eq. (14). Given the function $w(\hat{c}, u)$, except for a set $\{\hat{c}, u\}$ of measure zero, there exists a sufficiently small $\delta > 0$ such that for any bounded $\hat{c}_1, \hat{c}_2$ and u_1, u_2, if $\|\hat{c}_1 - \hat{c}_2\|_\infty < \delta$ and $|u_1 - u_2| < \delta$ then $|w(\hat{c}_1, u_1) - w(\hat{c}_2, u_2)| = 0$.

Proof: See Appendix I-A.

Following the ODE approach, we freeze the parameter $\hat{c}(\tau)$ at time τ and take the average of $F(\hat{c}(\tau), u(t))$ over the random variable $u(t)$ as in Eq. (32). For this purpose, the winning index $w(\cdot)$ in Eq. (3) will be represented as $w(x, \tau)$ which is a function of the random variable x and the frozen time τ (and $\hat{c}(\tau)$). However, we will often omit τ for convenience. Even though we sample the random variable $u(t)$, we would take average with respect to the random variable x, using the reference diffeomorphism $u(x)$ as in Eq. (9). For instance, we define $\Delta\hat{c}_1(\tau)$ and $\Delta\hat{c}_2(\tau)$ respectively by

$$\begin{aligned}\Delta\hat{c}_1(\tau) &:= \int_{\mathcal{X}} \delta\hat{c}_1(x, \tau) f_X(x) dx, \\ \Delta\hat{c}_2(\tau) &:= \int_{\mathcal{X}} \delta\hat{c}_2(x, \tau) f_X(x) dx.\end{aligned} \tag{33}$$

We summarize sufficient conditions for the correct convergence of the learning coverage control algorithm.

- A.1 p^o in Eq. (7), random signal locations $u(t) \in \mathcal{U}$ with an associated unknown pdf f_U, and the kernel function $\mathcal{K}$ in Eq. (14) are compatible in the sense that the family of smooth functions in Eq. (14) contains an optimal configuration in Eq. (10). Moreover, if $q'_\gamma > 0$, $\forall \gamma \in \mathcal{I}$, $q(\cdot, t)$ is an orientation preserving map.
- A.2 p^o, f_U, the kernel function $\mathcal{K}$, β_1 in Eq. (25) and β_2 in Eq. (27) satisfy that $\Delta\hat{c}_1 \neq -\Delta\hat{c}_2$, for any $\Delta\hat{c}_2 \neq 0$ where $\Delta\hat{c}_1$ and $\Delta\hat{c}_2$ are defined in Eq. (33).

Remark 4: The condition A.1 could be a limitation of our approach. Since we are using the fixed, finite number of kernels, the family of smooth functions generated from these kernels may not contain a configuration for a particular combination of p^o and f_U. By the unique structures of $\delta\hat{c}_1$ and $\delta\hat{c}_2$ in Eqs. (25) and (27) respectively, A.2 can be satisfied. A.2 may be checked analytically or numerically by using the following formulae for given a set of p^o, f_U, kernel

vectors, β_1 and β_2:

$$\begin{aligned}\Delta\hat{c}_1 &= \int_{\mathcal{X}} \beta_1 \frac{\delta\hat{c}_1^{X(w(x))}}{p_w^o(x)} f_X(x)dx \\ &= \sum_{\gamma\in\mathcal{I}\setminus\left\{\underline{\partial}x,\overline{\partial}x\right\}} \beta_1 \frac{(k_{\overline{\varphi}(\gamma)} - k_{\underline{\varphi}(\gamma)})}{2p_\gamma^o} p_\gamma \\ &\quad + \beta_1 \frac{k_\gamma + k_{\overline{\varphi}(\gamma)}}{2p_\gamma^o} p_\gamma\Big|_{\gamma=\underline{\partial}x} - \beta_1 \frac{k_\gamma + k_{\underline{\varphi}(\gamma)}}{2p_\gamma^o} p_\gamma\Big|_{\gamma=\overline{\partial}x},\end{aligned} \tag{34}$$

$$\begin{aligned}\Delta\hat{c}_2 &= \int_{\mathcal{X}} \beta_2 \frac{k'_{w(x)}(|q'_{w(x)}| - q'_{w(x)})[\mathrm{sign}(q'_{w(x)}) - 1] f_X(x)}{p^o_{w(x)}} dx \\ &= \sum_{\gamma\in\mathcal{I}} \beta_2 \frac{k'_\gamma(|q'_\gamma| - q'_\gamma)[\mathrm{sign}(q'_\gamma) - 1] p_\gamma}{p_\gamma^o},\end{aligned} \tag{35}$$

where $p_\gamma = \int_{R_\gamma} f_U(u)du$. Notice that $\Delta\hat{c}_1$ and $\Delta\hat{c}_2$ in Eqs. (34) and (35) build on different kernel vectors k_γ and k'_γ, respectively (see Fig. 4).

Remark 5: In general, $q_\gamma(0)$ for all $\gamma \in \mathcal{I}$ may be initially far away from the support of the pdf f_U. However, it is straightforward to see that the algorithm attracts the agents into the support of the pdf f_U. Thus, in the following argument, we assume that positions of winning agents whose indices are not extremum values ($w \in \mathcal{I}\setminus\{\underline{\partial}x, \overline{\partial}x\}$) are contained in the support of the pdf f_U, i.e., $q_w \in \mathrm{Supp}(f_U)$, where $w \in \mathcal{I}\setminus\{\underline{\partial}x, \overline{\partial}x\}$.

For convergence analysis, we need to calculate the changes in the usage probability distribution $\{p_\gamma \,|\, \gamma \in \mathcal{I}\}$ caused by changes in the influence coefficient vector $\hat{c}(\tau)$. The relationship is elaborated in the following lemma.

Lemma 6: The time derivative of the usage probability distribution $\dot{p}_{w(\tau)}$ is related to the time derivative of the influence coefficient vector $\dot{\hat{c}}(\tau)$ by

$$\dot{p}_{w(\tau)} \simeq f_U^*(q_{w(\tau)}, w(\tau)) \left[\delta\hat{c}_1^{X(w(\tau))}\right]^T \dot{\hat{c}}(\tau), \tag{36}$$

where $\delta\hat{c}_1^{X(w)}$ is defined in Eq. (26) and $f_U^* : \mathcal{U}\times\mathcal{I} \to \mathbb{R}_{\geq 0}$ is defined by

$$f_U^*(q_w, w) := \begin{cases} f_U(q_w), & \text{if } w \in \mathcal{I}\setminus\left\{\underline{\partial}x, \overline{\partial}x\right\} \\ f_U\left(\frac{q_w + q_{\overline{\varphi}(w)}}{2}\right), & \text{if } w = \underline{\partial}x, \\ f_U\left(\frac{q_w + q_{\underline{\varphi}(w)}}{2}\right), & \text{if } w = \overline{\partial}x. \end{cases} \tag{37}$$

Moreover, the approximation symbol used in Eq. (36) will be replaced with an equal sign for the case of uniform pdfs.

Proof: See Appendix I-B. This lemma is essential to prove our main result.

We introduce our main result for the uniform pdf f_U case.

Theorem 7: Consider the proposed learning coordination algorithm in Eq. (28) *under conditions A.*1 *and A.*2 *with a uniform pdf f_U. Then the locational vector $q(t)$ of the sensing agents converges to an optimal codebook vector $\bar{q}^o$ almost surely, satisfying Eq. (10).*

Remark 8: We can obtain a similar result to Theorem 7 for a class of nonuniform pdfs f_U. Define the functions $q^o : \mathcal{U} \to \mathcal{U}$ by

$$q^o(u) := \arg\left\{\min_{q_\gamma^o, \forall\gamma\in\mathcal{I}} |u - q_\gamma^o|\right\}, \tag{38}$$

where q_γ^o is defined in Eq. (10) and $w^o : \mathcal{U} \to \mathcal{I}$ by

$$w^o(u) := \arg \left\{ \min_{\gamma \in \mathcal{I}} |u - q_\gamma^o| \right\}. \tag{39}$$

Define also the normalized variation gain $\kappa(\cdot) \in \mathbb{R}$ of a pdf f_U by the formula

$$\kappa(x,\tau) := \frac{f_U^*(q_{w(x,\tau)}, w(x,\tau)) - f_U^*(q^o(u(x)), w^o(u(x)))}{2 f_U^*(q^o(u(x)), w^o(u(x)))}, \tag{40}$$

where f_U^* is defined in Eq. (37). Notice that this gain becomes zero when f_U is a uniform pdf and/or $q(\tau) = \bar{q}^o$ as defined in Eq. (10). Roughly speaking, our new learning algorithm can deal with nonuniform pdfs having relatively small $\kappa(\cdot)$ in Eq. (40) (see [28]).

Now we present the proof of our main result.

Proof of Theorem 7: Let us define the lower bounded functionals

$$\begin{aligned} V(\hat{c}(\tau)) &:= V_1(\hat{c}(\tau)) + V_2(\hat{c}(\tau)), \\ V_1(\hat{c}(\tau)) &:= \beta_1 \int_{\mathcal{X}} \frac{p_{w(x)}}{p^o_{w(x)} f_U^*(q^o(u(x)), w^o(u(x)))} f_X(x) dx, \\ V_2(\hat{c}(\tau)) &:= \frac{\beta_2}{2} \int_{\mathcal{X}} \frac{\left(\left| q'_{w(x)} \right| - q'_{w(x)} \right)^2}{p^o_{w(x)}} f_X(x) dx, \end{aligned} \tag{41}$$

where f_U^* is defined in Eq. (37) and $w(x) = w(u(x))$ is based on $q_\gamma(\tau)$ in Eqs. (13) and (14) given by Eq. (3). p_γ^o is the predefined optimal usage target probability distribution, as defined in Eq. (7). $\beta_1 > 0$ and $\beta_2 > 0$ are the weighting gains.

Applying the ODE approach to Eqs. (3), (25), (27), and (30), we obtain

$$\dot{\hat{c}}(\tau) = -\Delta \hat{c}_1(\tau) - \Delta \hat{c}_2(\tau), \tag{42}$$

where $\Delta \hat{c}_1(\tau)$ and $\Delta \hat{c}_2(\tau)$ are defined by Eq. (33).

Differentiating $V_1(\hat{c}(\tau))$ in Eq. (41) with respect to time τ, and utilizing Eq. (36) in Lemma 6, we obtain

$$\begin{aligned} \dot{V}_1(\hat{c}(\tau)) &= \int_{\mathcal{X}} \frac{\beta_1 \left(\frac{\partial p_{w(x)}}{\partial \hat{c}} \right)^T f_X(x)}{p^o_{w(x)} f_U^*(q^o(u(x)), w^o(u(x)))} dx \dot{\hat{c}}(\tau) \\ &= \int_{\mathcal{X}} \left(\frac{\beta_1 \delta \hat{c}_1^{X(w(x))}}{p^o_{w(x)}} \frac{f_U^*(q_w, w)}{f_U^*(q^o(u(x)), w^o(u(x)))} \right)^T f_X(x) dx \dot{\hat{c}}(\tau) \\ &= \int_{\mathcal{X}} \delta \hat{c}_1^T(x) f_X(x) dx \dot{\hat{c}}(\tau) \\ &+ \int_{\mathcal{X}} \delta \hat{c}_1^T(x) \left[\frac{f_U^*(q_w, w) - f_U^*(q^o(u(x)), w^o(u(x)))}{f_U^*(q^o(u(x)), w^o(u(x)))} \right] f_X(x) dx \dot{\hat{c}}(\tau) \\ &= \int_{\mathcal{X}} \delta \hat{c}_1^T(x) f_X(x) dx \dot{\hat{c}}(\tau) \\ &= [\Delta \hat{c}_1(\tau)]^T \dot{\hat{c}}(\tau). \end{aligned} \tag{43}$$

Taking the second partial derivative of $V_1(\hat{c})$ with respect to $\hat{c}$, we obtain

$$\frac{\partial}{\partial \hat{c}}\left(\frac{\partial V_1(\hat{c})}{\partial \hat{c}_m}\right) = \frac{\partial}{\partial \hat{c}}\left(\beta_1 \int_{\mathcal{X}} \frac{\left(\frac{\partial p_{w(x)}}{\partial \hat{c}}\right)_{(m)}}{f_U^*(q^o(u(x)), w^o(u(x)))} \frac{f_X(x)}{p^o_{w(x)}} dx\right) = 0 \in \mathbb{R}^{N+2m}, \tag{44}$$

where $\hat{c}_m$ is the m-th element of $\hat{c}$. As can be seen in Eq. (44), the second derivative of $V_1(\hat{c})$ with respect to $\hat{c}$ is a zero matrix, $\frac{\partial^2 V_1(\hat{c})}{\partial \hat{c}^2} = 0$. Taking the time derivative of $V_2(\hat{c}(\tau))$ with respect to time τ, $\dot{V}_2(\hat{c}(\tau))$ is obtained by

$$\begin{aligned}
\dot{V}_2(\hat{c}(\tau)) =& \beta_2 \int_{\mathcal{X}} \frac{f_X(x)}{p^o_{w(x)}} \left(\left|q'_{w(x)}\right| - q'_{w(x)}\right) \left[\text{sign}(q'_{w(x)}) - 1\right] k'^T_{w(x)} dx \dot{\hat{c}}(\tau) \\
=& \left(\frac{\partial V_2(\hat{c}(\tau))}{\partial \hat{c}(\tau)}\right)^T \dot{\hat{c}}(\tau) \\
=& \left(\int_{\mathcal{X}} \delta \hat{c}_2(x) f_X(x) dx\right)^T \dot{\hat{c}}(\tau).
\end{aligned} \tag{45}$$

The matrix of the second derivative of $V_2(\hat{c})$ with respect to $\hat{c}$ is positive semi-definite

$$\frac{\partial^2 V_2(\hat{c})}{\partial \hat{c}^2} = \beta_2 \int_{\mathcal{X}} k'_{w(x)} k'^T_{w(x)} \left[\text{sign}(q'_{w(x)}) - 1\right]^2 \frac{f_X(x)}{p^o_{w(x)}} dx \succeq 0. \tag{46}$$

From Eqs. (43) and (45), we have

$$\begin{aligned}
\dot{V}_1(\hat{c}(\tau)) &= \left(\frac{\partial V_1(\hat{c})}{\partial \hat{c}}\right)^T \dot{\hat{c}}(\tau) = \Delta \hat{c}_1(\tau)^T \dot{\hat{c}}(\tau) \\
\dot{V}_2(\hat{c}(\tau)) &= \left(\frac{\partial V_2(\hat{c})}{\partial \hat{c}}\right)^T \dot{\hat{c}}(\tau) = \Delta \hat{c}_2^T(\tau) \dot{\hat{c}}(\tau).
\end{aligned} \tag{47}$$

From Eqs. (41), (42) and (47), $\dot{V}(\hat{c}(\tau))$ can be represented as

$$\begin{aligned}
\dot{V}(\hat{c}(\tau)) &= \dot{V}_1(\hat{c}(\tau)) + \dot{V}_2(\hat{c}(\tau)) \\
&= -\|\Delta \hat{c}_1(\tau) + \Delta \hat{c}_2(\tau)\|^2.
\end{aligned} \tag{48}$$

From Eq. (48), $\dot{V}(\hat{c}(\tau))$ is negative semi-definite. Integrating Eq. (48) with respect to time, for all $T \geq 0$, $V(\hat{c}(T)) - V(\hat{c}(0)) = \int_0^T \dot{V}(\hat{c}(\tau)) d\tau$. This implies that $V(\hat{c}(T)) \leq V(\hat{c}(0))$, $V(\hat{c}) \in \mathcal{L}_\infty$, and $V_1(\hat{c})$ and $V_2(\hat{c})$ are bounded. Notice that $\Delta \hat{c}_1 \in \mathcal{L}_\infty$. From Eqs. (41) and (45), utilizing Cauchy-Schwartz inequality we obtain

$$\|\Delta \hat{c}_2(\tau)\| \leq 2\sqrt{2\beta_2} \left\{\int_{\mathcal{X}} k'^T_{w(x)} k'_{w(x)} dx\right\}^{1/2} V_2^{1/2}(\hat{c}(\tau)). \tag{49}$$

Thus, $\Delta \hat{c}_2(\tau) \in \mathcal{L}_\infty$ since $V_2(\hat{c}) \in \mathcal{L}_\infty$. Now we obtain

$$\dot{\hat{c}}(\tau) = -(\Delta \hat{c}_1(\tau) + \Delta \hat{c}_2(\tau)) \in \mathcal{L}_\infty. \tag{50}$$

From Eq. (47) we obtain

$$\frac{d\Delta \hat{c}_1(\tau)}{d\tau} = \left(\frac{\partial^2 V_1(\hat{c})}{\partial \hat{c}^2}\right)^T \dot{\hat{c}}(\tau), \; \frac{d\Delta \hat{c}_2(\tau)}{d\tau} = \left(\frac{\partial^2 V_2(\hat{c})}{\partial \hat{c}^2}\right)^T \dot{\hat{c}}(\tau). \tag{51}$$

$\dot{\hat{c}}(\tau)$, $\left(\frac{\partial^2 V_1(\hat{c})}{\partial \hat{c}^2}\right)$ and $\left(\frac{\partial^2 V_2(\hat{c})}{\partial \hat{c}^2}\right)$ are bounded from Eqs. (50), (44) and (46). By differentiating $\dot{\hat{c}}(\tau)$ and $\dot{V}(\hat{c})$ in Eqs. (47) and (50), respectively, with respect to time τ, we can conclude that $\ddot{\hat{c}}(\tau) \in \mathcal{L}_\infty$ and $\ddot{V}(\tau) \in \mathcal{L}_\infty$. Thus, $\dot{V}(\tau)$ is uniformly continuous in time τ. By Barbalat's lemma [29] or Lyapunov like lemma [30], we conclude that

$$\lim_{\tau\to\infty} \dot{V}(\tau) = -\lim_{\tau\to\infty} \|\Delta\hat{c}_1(\tau) + \Delta\hat{c}_2(\tau)\|^2 = 0 \quad a.s. \tag{52}$$

Due to the condition A.2, Eq. (52) implies that

$$\lim_{\tau\to\infty} \Delta\hat{c}_1(\tau) = 0,\ \lim_{\tau\to\infty} \Delta\hat{c}_2(\tau) = 0 \quad a.s. \tag{53}$$

From Eq. (53), $\Delta\hat{c}_1(\tau)$ can then be rewritten as

$$\begin{aligned}
\int_{\mathcal{X}} \delta\hat{c}_1(x) f_X(x) dx &= \int_{\mathcal{X}} \frac{\beta_1 \delta \hat{c}_1^{X(w(x))} f_X(x)}{p^o_{w(x)}} dx \\
&= \int_{\mathcal{X}} \frac{\beta_1 \delta \hat{c}_1^{X(1)} \delta_{1,w(x)} f_X(x)}{p^o_1} dx + \cdots \\
&\quad + \int_{\mathcal{X}} \frac{\beta_1 \delta \hat{c}_1^{X(N)} \delta_{N,w(x)} f_X(x)}{p^o_N} dx \\
&= \frac{\beta_1 p_1}{p^o_1}\left[\frac{k_1+k_2}{2}\right] + \frac{\beta_1 p_2}{p^o_2}\left[\frac{k_3-k_1}{2}\right] + \cdots \\
&\quad + \frac{\beta_1 p_{N-1}}{p^o_{N-1}}\left[\frac{k_N-k_{N-2}}{2}\right] + \frac{\beta_1 p_N}{p^o_N}\left[\frac{-k_N-k_{N-1}}{2}\right].
\end{aligned} \tag{54}$$

Since k_γ's in Eq. (54) are discretized radial basis kernels centered at $\gamma \in \mathcal{I}$, from Eq. (54) (see Fig. 4), we conclude that $\Delta\hat{c}_1(\tau) = \Delta\hat{c}_2(\tau) = 0$ and $\sum_{i\in\mathcal{I}} p_i = \sum_{i\in\mathcal{I}} p^o_i = 1$ imply that the usage probability p_γ in Eq. (6) is equal to the target probability p^o_γ in Eq. (7) for all $\gamma \in \mathcal{I}$, i.e., $p = p^o$. $\Delta\hat{c}_2(\tau) = 0$ along with $\Delta\hat{c}_1(\tau) = 0$ implies that q_γ is monotonically non-decreasing and $q(x,t)$ is orientation preserving (A.1). By A.1, the locational vector $q(t)$ converges to an optimal codebook vector $\bar{q}^o$ almost surely. This completes the proof of Theorem 7. QED.

Remark 9: The convergence result (Theorem 7) for the centralized form of the proposed learning coverage control algorithm in Eq. (31) (in terms of $\hat{c}(t)$) implies the same convergence result for the decentralized form of the algorithm (in terms of $q_\gamma(t)$ and $q'_\gamma(t)$) in Eq. (28).

VII. Simulation Results and Discussion

Numerical simulations were carried out for different pdfs (f_U) and target probability distributions (p^o). A total of 34 agents and 4 fictitious agents ($m = 2$), as in Eq. (12) were used in all simulations. The local kernel used in this simulation study is given by

$$\mathcal{K}(x,\nu) = \begin{cases} \frac{1}{\sigma\sqrt{2\pi}}\, e^{-\frac{1}{2\sigma^2}(x-\nu)^2}, & \text{if } |x-\nu| < \underline{\sigma}, \\ b(x,\nu), & \text{if } \underline{\sigma} \le |x-\nu| \le \bar{\sigma}, \\ 0, & \text{if } |x-\nu| > \bar{\sigma}, \end{cases} \tag{55}$$

where $b(x,\nu)$ is a ($\mathcal{C}^1$) bump function that makes the kernel smoothly diminish to 0, and $\bar{\sigma} > \underline{\sigma} > 3\sigma > 0$ and $\bar{\sigma} \simeq \underline{\sigma} \simeq 3\sigma$. A value of $\sigma = 1.5$ was used for the kernel given by Eq. (55). The initial positions of agents $q_\gamma(0)$'s were randomly distributed. The performance of the proposed scheme for different situations will be analyzed in the following systematic manner:

- We will compare the partitioning border vector $b(t) := [b_1(t) \cdots b_{N-1}(t)]^T$ calculated from the locations of agents at iteration time t

$$b_i(t) := \frac{q_i(t) + q_{i+1}(t)}{2}, \forall i \in \mathcal{I} \setminus \{N\}, \tag{56}$$

  with respect to the analytically calculated optimal border vector[3] b^o for the corresponding pdf f_U and the target probability distribution p^o. The root mean square (RMS) error value between $b(t)$ and b^o will be computed.
- We will compare the usage probability distribution p with respect to the target usage probability distribution p^o in terms of the Kullback-Leibler (KL) measure of cross-entropy between p and p^o [31]. Since p_γ is not available, it will be estimated by the following algorithm:

$$\begin{aligned} \hat{p}_\gamma(t+1) &= \hat{p}_\gamma(t) + \alpha(t)(-\hat{p}_\gamma(t) + \delta_{\gamma,w(t)}), \quad \forall \gamma \in \mathcal{I}, \\ \hat{p}_\gamma(0) \geq 0 \quad \forall \gamma \in \mathcal{I}, &\quad \sum_{\gamma=1}^{N} \hat{p}_\gamma(0) = 1, \end{aligned} \tag{57}$$

  where $w(t)$ is the winning index given by Eq. (3), $\delta_{i,j}$ is the Kronecker delta function, and $\alpha(t)$ is the stochastic approximation gain introduced in Eq. (21). The KL measure between $\hat{p} := [\hat{p}_1 \cdots \hat{p}_N]^T$ from Eq. (57) and p^o is given by

$$D(\hat{p}, p^o) = \sum_{\gamma=1}^{N} \hat{p}_\gamma \ln \frac{\hat{p}_\gamma}{p^o_\gamma},$$

  where $D(\hat{p}, p^o) \geq 0$, and $D(\hat{p}, p^o)$ vanishes if and only if $\hat{p} = p^o$.

A. A Uniform Probability Density Function

Consider a uniform pdf f_U with $u \sim \mathrm{U}[0, 30]$ and the target probability distribution $\{p^o_\gamma \,|\, \gamma \in \mathcal{I}\}$ (see green circles in Fig. 8-(a))

$$p^o_\gamma := \frac{\cos\left(\frac{8\pi\gamma}{N-1}\right) + 2\left(1 + \frac{\gamma}{N-1}\right)}{\sum_{j=0}^{N-1}\left[\cos\left(\frac{8\pi j}{N-1}\right) + 2\left(1 + \frac{j}{N-1}\right)\right]}, \forall \gamma \in \mathcal{I}. \tag{58}$$

In this case, the border vector $b(t)$ of agents from the proposed algorithm successfully converges to the analytically calculated optimal border vector b^o for f_U and p^o after 20000 iterations as depicted in Fig. 7-(a). The convergence rate in terms of the RMS error value between $b(t)$ and b^o v.s. iteration time is plotted in a logarithmic scale to magnify the transient behavior of the algorithm as shown in Fig. 7-(b). The final RMS error value between $b(t)$ and b^o at $t = 20000$ is 0.8185.

Fig. 8-(a) depicts the estimated usage probability $\hat{p}_\gamma$ from the new learning law, as computed in Eq. (57). This plot shows that $p \approx \hat{p}$ converges to p^o defined in Eq. (58) after 20000 iterations. Fig. 8-(b) illustrates the KL measure between $\hat{p}$ and p^o v.s. iteration time, showing that $\hat{p} \to p^o$ $a.s.$ as $t \to \infty$.

[3]Notice that for given f_U and p^o, only b^o can be determined uniquely.

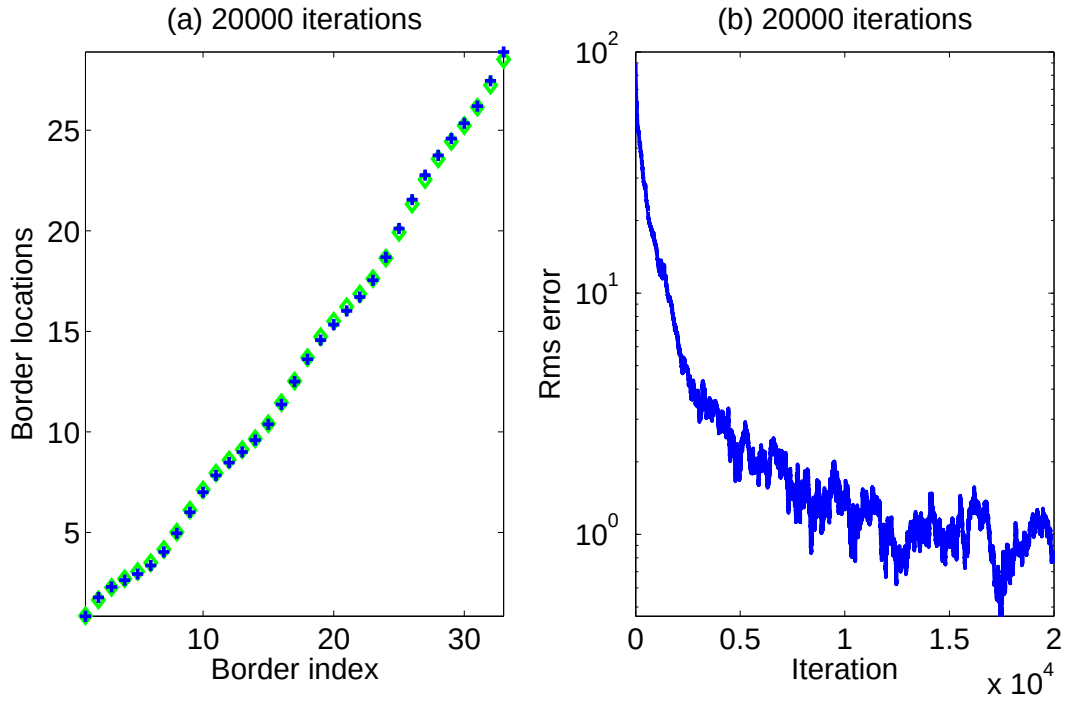

Fig. 7. Results for a uniform pdf f_U and a target usage probability p^o in Eq. (58). (a) The analytically calculated border locations b^o (diamonds $\diamond$) and border locations of agents $b(t)$ calculated from the new learning algorithm (pluses +) after 20000 iterations. (b) The RMS error value between $b(t)$ and b^o v.s. iteration time. The final RMS error value is 0.8185.

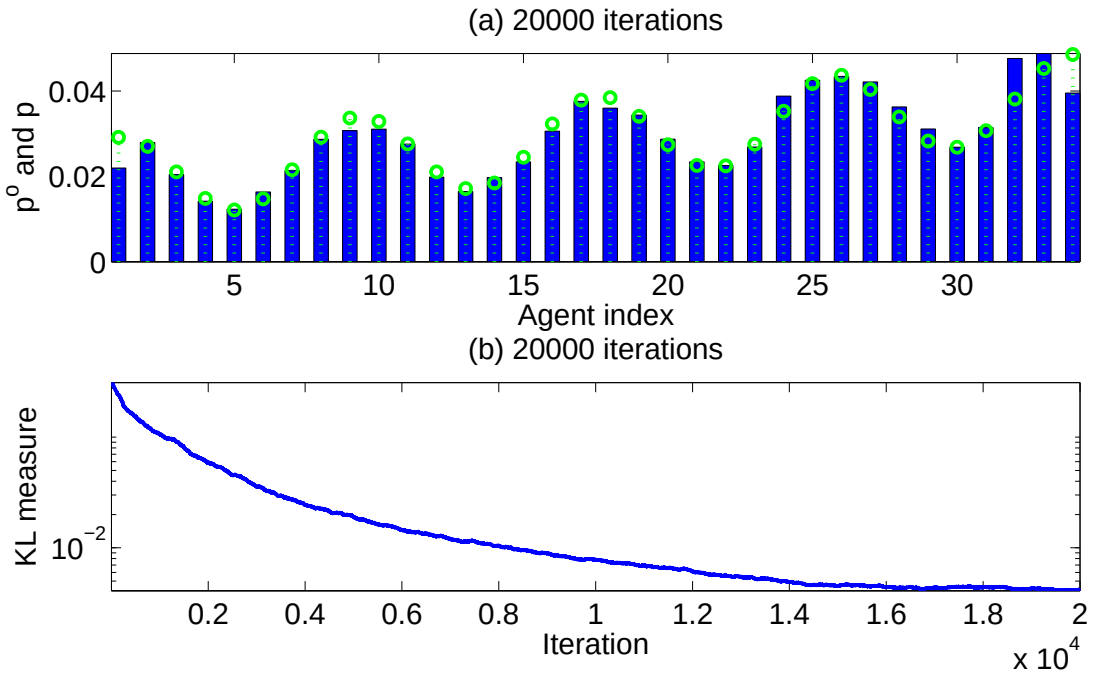

Fig. 8. A uniform pdf f_U: (a) The usage probability distribution estimate $\hat{p}$ (solid bars), and the target usage probability distribution p^o (green circles) in Eq. (58). (b) The Kullback-Leibler (KL) measure of cross-entropy between $\hat{p}$ and p^o v.s. iteration time.

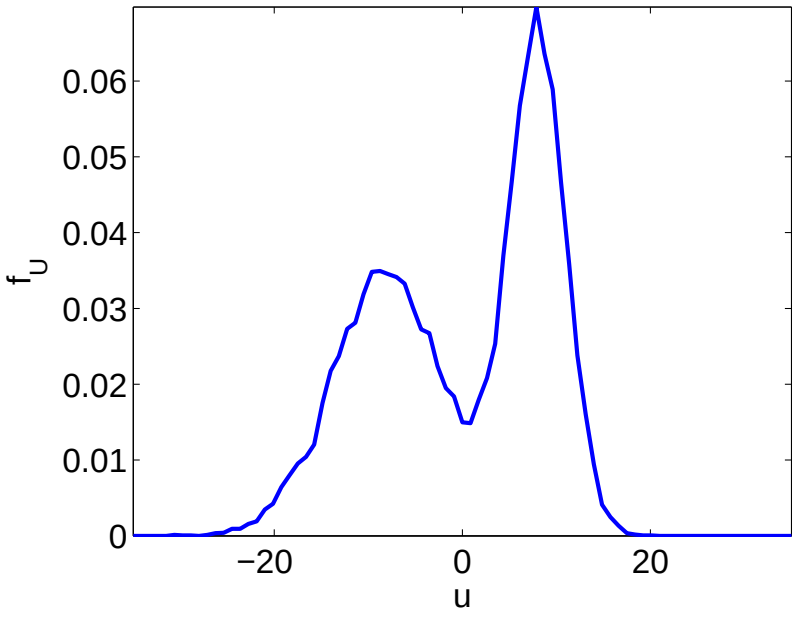

Fig. 9. The histogram for a bimodal mixture model with Gaussian components defined in Eq. (59).

B. A Bimodal Mixture Model

To test our algorithm to a nonuniform pdf f_U, we consider a bimodal mixture model with two normal components. The locational random sequence u is assigned to one of the two normal densities, each of which has a prior discrete probability P_1 or P_2. The joint probability density function of the random sequence u is given by

$$f_U(u|\{\theta_1,\theta_2\}) = f_{U_1}(u|\theta_1)P_1 + f_{U_2}(u|\theta_2)P_2, \tag{59}$$

where $\theta_i = [m_i, \sigma_i]^T$ is the sufficient statistics, $f_{U_i}(u|\theta_i) = \frac{1}{\sigma_i\sqrt{2\pi}}\, e^{-\frac{1}{2\sigma_i^2}(u-m_i)^2}$ is the i-th conditional probability density function, and $P_1 = 1/2$ and $P_2 = 1/2$ are the mixing probabilistic weights. We used that $m_1 = 8$, $\sigma_1 = 3$, $m_2 = -8$, and $\sigma_2 = 6$ for this case. The histogram of this bimodal mixture model is shown in Fig. 9.

Consider the target probability distribution $\{p^o_\gamma \,|\, \gamma \in \mathcal{I}\}$ (see green circles in Fig. 11-(a))

$$p^o_\gamma := \frac{\sin\left(\frac{8\pi\gamma}{N-1}\right) + 2\left(1+\frac{\gamma}{N-1}\right)}{\sum_{j=0}^{N-1}\left[\sin\left(\frac{8\pi j}{N-1}\right) + 2\left(1+\frac{j}{N-1}\right)\right]}, \forall\gamma \in \mathcal{I}. \tag{60}$$

Fig. 10-(a) shows that border vector $b(t)$ of agents using the proposed algorithm successfully converges to the analytically calculated b^o for the bimodal mixture pdf and p^o in Eqs. (59) and (60) respectively, after 20000 iterations. Fig. 10-(b) presents the convergence rate of the RMS error value between $b(t)$ and b^o v.s. iteration time. The final RMS error value between $b(t)$ and b^o is 1.4105.

Fig. 11-(a) depicts the estimated usage probability $\hat{p}_\gamma$ of the new learning law, which shows that $p \approx \hat{p}$ converges to p^o in Eq. (60) after 20000 iterations. Fig. 11-(b) also presents the KL measure between $\hat{p}$ and p^o v.s. iteration time, validating that $\hat{p} \to p^o$ $a.s.$ as $t \to \infty$.

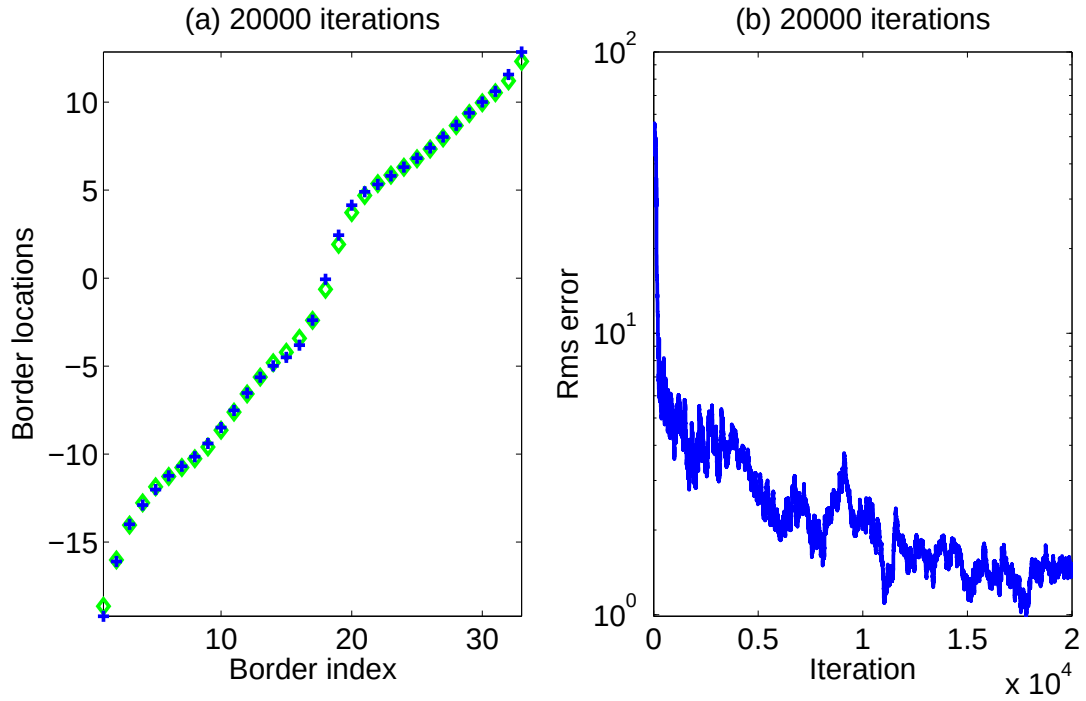

Fig. 10. Results for a bimodal mixture model pdf f_U and p^o in Eqs. (59) and (60) respectively. (a) The analytically calculated border locations of agents (diamonds $\diamond$) and border locations obtained from the new learning algorithm (pluses +) after 20000 iterations. (b) The RMS error value between $b(t)$ and b^o v.s. iteration time. The final RMS error value is 1.4105.

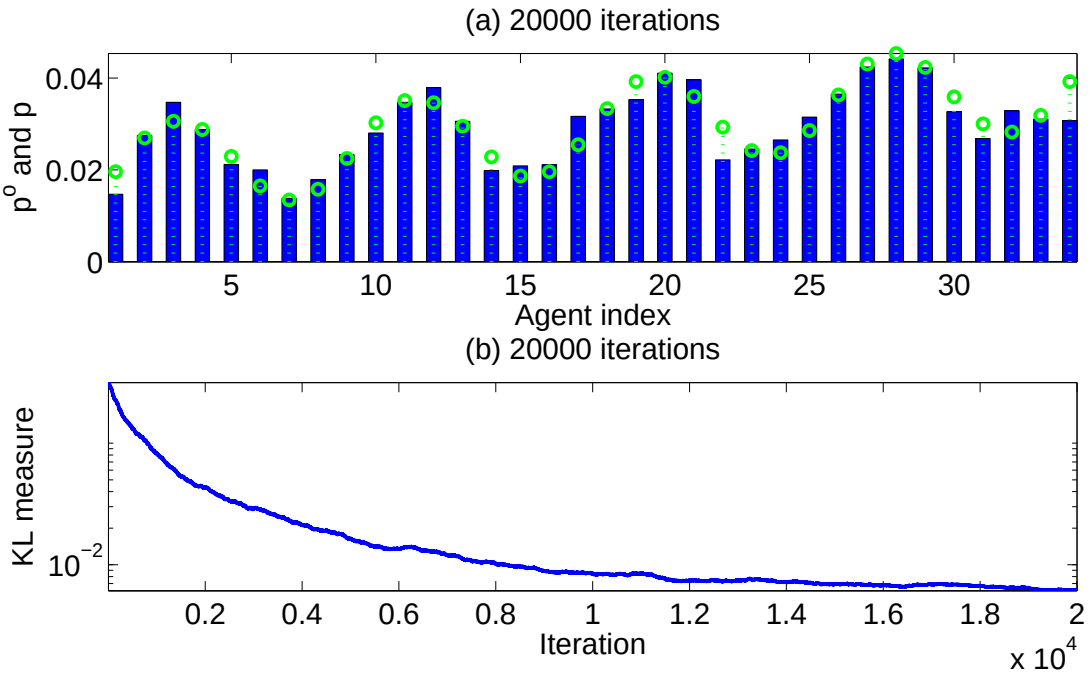

Fig. 11. A bimodal mixture model distribution: (a) The usage probability distribution estimate $\hat{p}$ (solid bars), and the uniform target usage probability distribution p^o (green circles). (b) The KL measure of cross-entropy between $\hat{p}$ and p^o v.s. iteration time.

C. A Piecewise Stationary Process

Consider p^o in Eq. (60) (see green circles in Fig. 11-(a)). We evaluated our algorithm under a non-stationary process that switches from the bimodal mixture model in Eq. (59) to a normal distribution. Suppose that the unknown and non-stationary environment has the same bimodal mixture distribution presented in Eq. (59) at the beginning. At iteration time $t = 3,000$, those a prior probabilities $P_1 = 1/2$ and $P_2 = 1/2$ in Eq. (59) switch instantaneously to $P_1 = 0$ and $P_2 = 1$ leaving only the second normal density component $\mathcal{N}(-8, 6^2)$. This simulates a non-stationary environment. Fig. 12 shows the new algorithm's adaptability to the time-varying non-stationary distribution. Mobile agents covered the region of the first normal component quickly moved to that of the second normal component in order to achieve optimality under new environment after $t = 3,000$.

Figs. 13 and 14 show that the proposed scheme is successfully applied to the given scenario in terms of our performance measures. In particular, Figs. 13-(b) and 14-(b) explicitly illustrate the transient effects and adaptability of our scheme on the abrupt change from the bimodal mixture model to the unimodal Gaussian distribution at $t = 3000$. As can be shown in these figures, agents quickly converge to the new distribution in terms of the performance measures. The time varying optimal border vector

$$b^o(t) = \begin{cases} b_1^o, & \text{if } t \leq 3000; \\ b_2^o, & \text{if } t \geq 3001, \end{cases}$$

was used for those two distributions in Fig. 13-(b). The final RMS error value is 1.5394.

In general, the proposed algorithm can deal with non-stationary processes with a slow switching behavior among different stationary processes[4] if the periods of those stationary processes are long enough for agents to converge. If the stochastic approximation gain $\alpha(t)$ in Eq. (20) is too small at the switching time for agents to adapt to the new process, the initialization of $\alpha(t)$ is necessary for the proper convergence. The detection of the switching time is feasible using the features of agents' states, and so they can reset $\alpha(t)$ for the correct convergence.

VIII. Conclusions

A new formulation of learning coverage control for distributed mobile sensing agents was presented. This new one-dimensional coordination algorithm enables us to control the agent's usage probability and a formation order for given (non-)stationary and unknown statistics of random signal locations. The almost sure convergence properties of the proposed coordination algorithm were analyzed using the ODE approach for random signal locations with unknown uniform pdfs. Successful simulation results for a uniform pdf and a bimodal mixture model with normal components of signal locations verified the convergence of the new coordination algorithm. Adaptability of the proposed algorithm to a piecewise stationary process was numerically evaluated. Extensions to the multidimensional sensor space are currently under investigation.

IX. Acknowledgements

The authors thank anonymous reviewers for their valuable suggestions on how to improve the readability of the paper.

[4]It is called a piecewise stationary process.

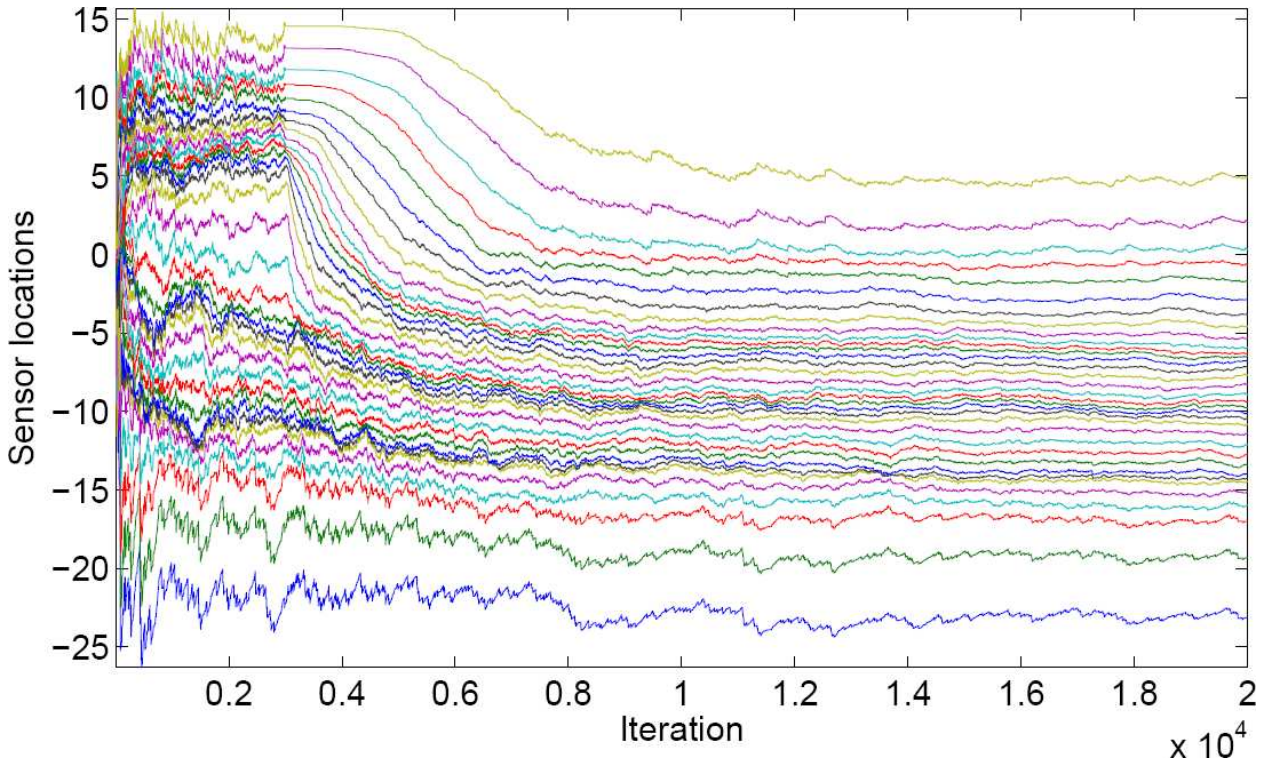

Fig. 12. Results for a non-stationary distribution that switches from a bimodal mixture model to a normal distribution and p^o in Eq. (60): Locations of agents v.s. iteration time.

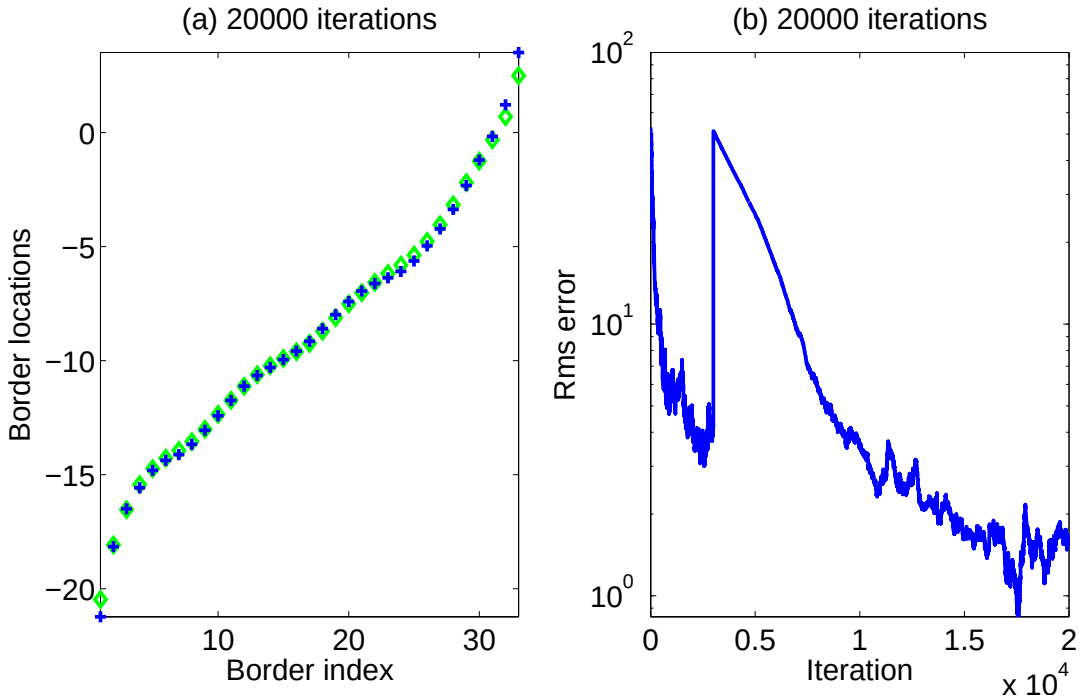

Fig. 13. A non-stationary distribution: (a) The analytically calculated border locations of agents (diamonds $\diamond$) and border locations obtained from the new learning algorithm (pluses +) after 20000 iterations. (b) The RMS error value between $b(t)$ and $b^o(t)$ v.s. iteration time. The final RMS error value is 1.5394.

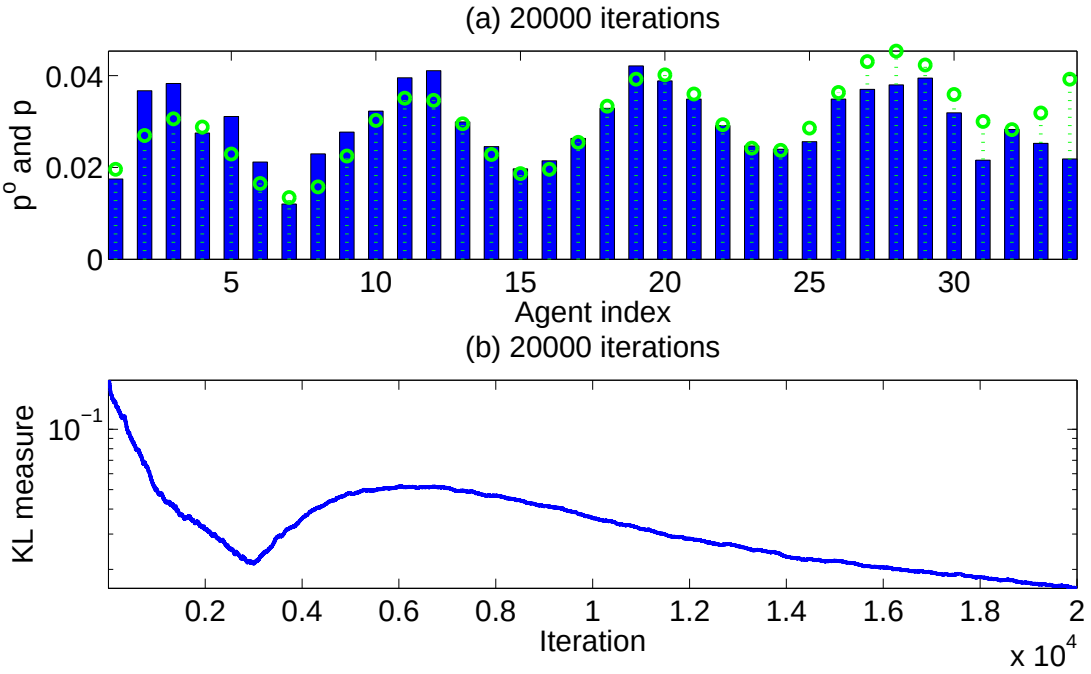

Fig. 14. A non-stationary distribution: (a) The usage probability distribution estimate $\hat{p}$ (solid bars), and the uniform target usage probability distribution p^o (green circles). (b) The KL measure of cross-entropy between $\hat{p}$ and p^o v.s. iteration time.

APPENDIX I
PROOFS

A. Proof of Lemma 3

$w(\hat{c}_1, u_1)$ is the winning index for $\hat{c}_1$ and u_1. For any index $\gamma \in \mathcal{I}$, $\gamma \neq w$, there exists an $\epsilon > 0$ such that $|u_1 - k_\gamma^T \hat{c}_1| - |u_1 - k_w^T \hat{c}_1| > \epsilon$, since agent w is a winner. Select $\delta(\epsilon)$ such that $\delta < \frac{\epsilon}{2(1+\|K\|_{i,\infty})}$, where $\|K\|_{i,\infty}$ is the induced infinity norm defined by $\|K\|_{i,\infty} := \sup_{z\in\mathbb{R}^m, z\neq 0} \frac{\|Kz\|_\infty}{\|z\|_\infty}$. Then, using the triangular inequality, we obtain

$$\epsilon \leq |u_2 - k_\gamma^T \hat{c}_2| - |u_2 - k_w^T \hat{c}_2| + 2|u_1 - u_2| + 2\|K\|_{i,\infty}\|\hat{c}_1 - \hat{c}_2\|_\infty. \tag{61}$$

Since $\|K\|_{i,\infty} \geq \frac{\|K\Delta\hat{c}\|_\infty}{\|\Delta\hat{c}\|_\infty}$ and $2\delta(1 + \|K\|_{i,\infty}) < \epsilon$, therefore, for any u_2, and $\hat{c}_2$ such that $\|\hat{c}_1 - \hat{c}_2\|_\infty < \delta$ and $|u_1 - u_2| < \delta$, $|u_2 - k_\gamma^T \hat{c}_2| - |u_2 - k_w^T \hat{c}_2| > 0$, for all $\gamma \in \mathcal{I}, \gamma \neq w$. Thus, $|w(\hat{c}_1, u_1) - w(\hat{c}_2, u_2)| = 0$ follows. QED.

B. Proof of Lemma 6

Taking partial derivative of p_i with respect to $q_{\underline{\varphi}(i)}$ gives us

$$\begin{aligned}\frac{\partial p_i}{\partial q_{\underline{\varphi}(i)}} &= \frac{\partial}{\partial q_{\underline{\varphi}(i)}} \int_{\frac{q_{\underline{\varphi}(i)}+q_i}{2}}^{\frac{q_i+q_{\overline{\varphi}(i)}}{2}} f_U(\xi)d\xi = \frac{\partial F_U(\frac{q_i+q_{\overline{\varphi}(i)}}{2})}{\partial q_{\underline{\varphi}(i)}} - \frac{\partial F_U(\frac{q_{\underline{\varphi}(i)}+q_i}{2})}{\partial q_{\underline{\varphi}(i)}} \\ &= -\frac{\partial F_U(\xi)}{\partial \xi}\bigg|_{\xi=\frac{q_{\underline{\varphi}(i)}+q_i}{2}} \frac{\partial \xi}{\partial q_{\underline{\varphi}(i)}}\bigg|_{\xi=\frac{q_{\underline{\varphi}(i)}+q_i}{2}} = -f_U((q_{\underline{\varphi}(i)} + q_i)/2)\frac{1}{2},\end{aligned} \tag{62}$$

where F_U is the cumulative distribution function of the random variable u. $F_U(\xi)$ is a monotonic function by definition, and we have $F_U(\xi) = \int_{-\infty}^{\xi} f_U(\eta)d\eta$, and $\frac{\partial F_U(\xi)}{\partial \xi}\big|_{\xi=u} = f_U(u)$. Similarly, we obtain the partial derivative of p_i with respect to $q_{\overline{\varphi}(i)}$ as:

$$\frac{\partial p_i}{\partial q_{\overline{\varphi}(i)}} = f_U\left((q_i + q_{\overline{\varphi}(i)})/2\right)\frac{1}{2}. \tag{63}$$

It is important to notice the following property:

$$\frac{\partial p_i}{\partial q_k} = 0, \text{ where } k \notin \{i, \underline{\varphi}(i), \overline{\varphi}(i)\}. \tag{64}$$

We will derive $\dot{p}_w$ in terms of $\dot{\hat{c}}$ for the following three cases.

1) A Case with $w \in \mathcal{I}\backslash\{\underline{\partial}x, \overline{\partial}x\}$: Consider a case with $w \in \mathcal{I}\backslash\{\underline{\partial}x, \overline{\partial}x\}$. If $i \notin \{\underline{\partial}x, \overline{\partial}x\}$ then $\frac{\partial p_i}{\partial q_i}$ is obtained by

$$\frac{\partial p_i}{\partial q_i} = \frac{\partial}{\partial q_i}\int_{\frac{q_{\underline{\varphi}(i)}+q_i}{2}}^{\frac{q_i+q_{\overline{\varphi}(i)}}{2}} f_U(\xi)d\xi = f_U((q_i + q_{\overline{\varphi}(i)})/2)\frac{1}{2} - f_U((q_{\underline{\varphi}(i)} + q_i)/2)\frac{1}{2}. \tag{65}$$

From Eqs. (62), (63), (64), and (65), we can take the time derivative of p_w as

$$\begin{aligned}\dot{p}_w &= \frac{\partial}{\partial t}\left(F_U((q_w + q_{\overline{\varphi}(w)})/2) - F_U((q_w + q_{\underline{\varphi}(w)})/2)\right) \\ &= \left(f_U(\xi_{2(w)})\frac{k_w^T + k_{\overline{\varphi}(w)}^T}{2} - f_U(\xi_{1(w)})\frac{k_w^T + k_{\underline{\varphi}(w)}^T}{2}\right)\dot{\hat{c}},\end{aligned} \tag{66}$$

where $\xi_{1(w)} = (q_w + q_{\underline{\varphi}(w)})/2$ and $\xi_{2(w)} = (q_w + q_{\overline{\varphi}(w)})/2$. Since $f_U(u)$ is continuous, for $w \in \mathcal{I}\backslash\{\underline{\partial}x, \overline{\partial}x\}$, we can make the following approximations

$$f_U(\xi_{1(w)}) \simeq f_U(q_w),\ f_U(\xi_{2(w)}) \simeq f_U(q_w). \tag{67}$$

Using Eq. (67) along with Eq. (66), $\frac{\partial p_w}{\partial \hat{c}}$ can be computed by

$$\frac{\partial p_w}{\partial \hat{c}} \simeq f_U(q_w)\left(\frac{k_{\overline{\varphi}(w)} - k_{\underline{\varphi}(w)}}{2}\right). \tag{68}$$

From Eq. (68), we start constructing a function $\delta\hat{c}_1^{X(\cdot)} : \mathcal{I} \to \mathbb{R}^{N+2m}$ by

$$\delta\hat{c}_1^{X(w)} := \left(\frac{k_{\overline{\varphi}(w)} - k_{\underline{\varphi}(w)}}{2}\right) \quad \text{if } w \in \mathcal{I}\backslash\left\{\underline{\partial}x, \overline{\partial}x\right\}. \tag{69}$$

We also start building the function $f_U^* : \mathcal{U} \times \mathcal{I} \to \mathbb{R}_{\geq 0}$ by

$$f_U^*(q_w, w) := f_U(q_w) \quad \text{if } w \in \mathcal{I}\backslash\left\{\underline{\partial}x, \overline{\partial}x\right\}. \tag{70}$$

2) A Case with $w = \underline{\partial} x$: Now we derive the time derivative of p_w, with $w = \underline{\partial} x$ in terms of the time derivative of the influence coefficient vector $\hat{c}$ by

$$\begin{aligned} \dot{p}_w &= \frac{\partial}{\partial t} \int_{-\infty}^{(q_w + q_{\overline{\varphi}(w)})/2} f_U(\xi) d\xi \\ &= \frac{\partial}{\partial t} \left(F_U((q_w + q_{\overline{\varphi}(w)})/2) - F_U(-\infty) \right) \\ &= \left(f_U(\xi_{2(w)}) \frac{k_w^T + k_{\overline{\varphi}(w)}^T}{2} \right) \dot{\hat{c}}. \end{aligned} \tag{71}$$

$\frac{\partial p_w}{\partial \hat{c}}$, with $w = \underline{\partial} x$ is then given by

$$\frac{\partial p_w}{\partial \hat{c}} = \left(f_U(\xi_{2(w)}) \frac{k_w + k_{\overline{\varphi}(w)}}{2} \right). \tag{72}$$

Define $\delta \hat{c}_1^{X(\cdot)}$ and f_U^*, for $w = \underline{\partial} x$ respectively by

$$\delta \hat{c}_1^{X(w)} := \left(\frac{k_w + k_{\overline{\varphi}(w)}}{2} \right) \quad \text{if } w = \underline{\partial} x, \tag{73}$$

and

$$f_U^*(q_w, w) := f_U(\xi_{2(w)} = (q_w + q_{\overline{\varphi}(w)})/2) \quad \text{if } w = \underline{\partial} x. \tag{74}$$

3) A Case with $w = \overline{\partial} x$: We derive the time derivative of p_w, with $w = \overline{\partial} x$ in terms of the time derivative of the influence coefficient vector $\hat{c}$ by

$$\begin{aligned} \dot{p}_w &= \frac{\partial}{\partial t} \int_{(q_w + q_{\underline{\varphi}(w)})/2}^{\infty} f_U(\xi) d\xi \\ &= \frac{\partial}{\partial t} \left(F_U(\infty) - F_U((q_w + q_{\underline{\varphi}(w)})/2) \right) \\ &= \left(-f_U(\xi_{1(w)}) \frac{k_w^T + k_{\underline{\varphi}(w)}^T}{2} \right) \dot{\hat{c}}. \end{aligned}$$

$\frac{\partial p_w}{\partial \hat{c}}$, with $w = \overline{\partial} x$ is then obtained by

$$\frac{\partial p_w}{\partial \hat{c}} = \left(-f_U(\xi_{1(w)}) \frac{k_w + k_{\underline{\varphi}(w)}}{2} \right). \tag{75}$$

We complete the construction of the functions $\delta \hat{c}_1^{X(\cdot)}$ and f_U^*, for the case with $w = \underline{\partial} x$, respectively by

$$\delta \hat{c}_1^{X(w)} := \left(-\frac{k_w + k_{\underline{\varphi}(w)}}{2} \right) \quad \text{if } w = \overline{\partial} x, \tag{76}$$

and

$$f_U^*(q_w, w) := f_U(\xi_{1(w)} = (q_w + q_{\underline{\varphi}(w)})/2) \quad \text{if } w = \overline{\partial} x. \tag{77}$$

Combining Eqs. (69), (73) (76), (67), (70), (74), and (77) the functions $\delta \hat{c}_1^{X(w)}$ and $f_U^*(q_w, w)$ are respectively redefined by Eqs. (26) and (37) in a unified fashion. From Eqs. (68), (72), (75), (26), and (37), we complete the proof of Lemma 6. QED.

REFERENCES

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

[2] J. Cortes and F. Bullo, "Coordination and geometric optimization via distributed dynamical systems," *SIAM Journal on Control and Optimization*, vol. 44, no. 5, pp. 1543–1574, 2005.

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

[4] J. Cortes, S. Martinez, and F. Bullo, "Spatially-distributed coverage optimization and control with limited-range interactions," *ESAIM. Control, Optimisation and Calculus of Variations*, vol. 11, pp. 691–719, 2005.

[5] Olfati-Saber, "Flocking for multi-agent dynamic systems: Algorithm and theory," *IEEE Transactions on Automatic Control*, vol. 51, no. 3, pp. 401–420, March 2006.

[6] E. Frazzoli and F. Bullo, "Decentralized algorithms for vehicle routing in a stochastic time-varying environment," in *Proceedings of Conference on Decision and Control*, Paradise Island, Bahamas, Dec. 2004, pp. 3357–3363.

[7] A. Arsie and E. Frazzoli, "Efficient routing of multiple vehicles with no communications," *International Journal of Robust and Nonlinear Control*, vol. 18, no. 2, pp. 154–164, 2008.

[8] A. Gersho and R. M. Gray, *Vector Quantization and Signal Compression*. Kluwer Academic Publishers, 2001.

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

[10] J. B. MacQueen, "Some methods of classification and analysis of multivariate observations," in *Proceedings of the Fifth Berkeley Symposium on Mathemtical Statistics and Probability*, 1967, pp. 281–297.

[11] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: a survey," *Computer Networks*, vol. 38, no. 4, pp. 393–422, March 2002.

[12] M. A. Abidi and R. C. Gonzales, Eds., *Data Fusion in Robotics and Machine Intelligence*. Academic Press, Inc., 1992.

[13] M. Chu, H. Haussecker, and F. Zhao, "Scalable information-driven sensor querying and routing for ad hoc heterogeneous sensor networks," *International Journal of High Performance Computing Applications*, vol. 16, no. 3, pp. 293–313, 2002.

[14] S. Oh, L. Schenato, P. Chen, and S. Sastry, "Tracking and coordination of multiple agents using sensor networks: system design, algorithms and experiments," *Proceedings of IEEE*, vol. 95, no. 1, pp. 234–254, Jan 2007.

[15] R. Grabowski, L. Navarro-Serment, C. Paredis, and P. Khosla, "Heterogeneous Teams of Modular Robots for Mapping and Exploration," *Autonomous Robots*, vol. 8, no. 3, pp. 293–308, 2000.

[16] J. Choi and R. Horowitz, "Topology preserving neural networks that achieve a prescribed feature map probability density distribution," in *Proceedings of the 24th American Control Conference (ACC)*, June 2005.

[17] M. de Berg, O. Schwarzkopf, M. van Kreveld, and M. Overmars, *Computational Geometry: Algorithms and Applications*, second, Ed. Springer-Verlag, 2000.

[18] E. B. Kosmatopoulos and M. A. Christodoulou, "Convergence properties of a class of learning vector quantization algorithms," *IEEE Transactions on Image Processing*, vol. 5, no. 2, pp. 361–368, February 1996.

[19] T. Hofmann and J. M. Buhmann, "Competitive learning algorithms for robust vector quantization," *IEEE Transactions on Signal Processing*, vol. 46, no. 6, pp. 1665–, June 1998.

[20] T. Kohonen, *Self Organizing Maps*. Spinger-Verlag, 1995.

[21] T. Martinetz and K. Schulten, "Topology representing networks," *Neural Networks*, vol. 7, no. 3, pp. 507–522, 1994.

[22] A. A. Sadeghi, "Asymptotic behavior of self-organizing maps with nonuniform stimuli distribution," *The Annals of Applied Probability*, vol. 8, no. 1, pp. 281–299, 1998.

[23] M. Hrsch, "Systems of differential equations wihch are competitive or cooperative II: convergence almost everywhere," *SIAM J. of Math. Anal.*, vol. 16, pp. 423–439, 1985.

[24] A. J. Storkey, "Truncated covariance matrices and toeplitz methods in gaussian processes," *Artificial Neural Networks-ICANN*, 1999.

[25] L. Ljung, "Analysis of recursive stochastic algorithms," *IEEE Transactions on Automatic Control*, vol. 22, no. 4, pp. 551–575, 1977.

[26] H. J. Kushner and G. G. Yin, *Stochastic Approximation Algorithms and Applications*. Springer, 1997.

[27] L. Ljung and T. Soderstrom, *Theory and Practice of Recursive Identification*. Cambridge, Massachusetts, London, England: The MIT Press, 1983.

[28] J. Choi, "Self-organizing algorithms for controlling network feature map probability distributions, and synthesizing multiple robust controllers," Ph.D. dissertation, University of California at Berkeley, May 2006.

[29] H. K. Khalil, *Nonlinear systems*, 2nd ed. Prentice Hall, 1996.

[30] J.-J. E. Slotine and W. Li, *Applied nonlinear control*. Prentice Hall, 1991.

[31] J. Kapur and H. K. Kesavan, *Entropy Optimization Principle with Applications*. Academic Press Inc., 1992.