2015 European Control Conference (ECC)
July 15-17, 2015. Linz, Austria

Multi-agents adaptive estimation and coverage control using Gaussian regression

Andrea Carron, Marco Todescato, Ruggero Carli, Luca Schenato, Gianluigi Pillonetto

Abstract— **We consider a scenario where the aim of a group of agents is to perform the optimal coverage of a region according to a sensory function. In particular, centroidal Voronoi partitions have to be computed. The difficulty of the task is that the sensory function is unknown and has to be reconstructed on line from noisy measurements. Hence, estimation and coverage needs to be performed at the same time. We cast the problem in a Bayesian regression framework, where the sensory function is seen as a Gaussian random field. Then, we design a set of control inputs which try to find a good balance between coverage and estimation, also discussing convergence properties of the algorithm. Numerical experiments show the effectiveness of the new approach.**

I. INTRODUCTION

The continuous progress on hardware and software is allowing the appearance of compact and relatively inexpensive autonomous vehicles embedded with multiple sensors (inertial systems, cameras, radars, environmental monitoring sensors), high-bandwidth wireless communication and powerful computational resources. While previously limited to military applications, nowadays the use of cooperating vehicles for autonomous monitoring of large environment, even for civilian applications, is becoming a reality. Although robotics research has obtained tremendous achievements with single vehicles, the trend of adopting multiple vehicles that cooperate to achieve a common goal is still very challenging and open problem.

In particular, an area that has attracted considerable attention for its practical relevance is the problem of *environmental partitioning problem* and *coverage control* whose objective is to partition an area of interest into subregions each monitored by a different robot trying to optimize some global cost function that measures the quality of service provided by the monitoring robots.

The "centering and partitioning" algorithm originally proposed by Lloyd [1] and elegantly reviewed in the survey [2] is a classic approach to environmental partitioning problems and coverage control problems. The Lloyd algorithm computes Centroidal Voronoi partitions as optimal configurations of an important class of objective functions called coverage functions. The Lloyd approach was first adapted for distributed coverage in the robotic multiagent literature control in [3]; see also the text [4] (Chapter 5 and literature notes in Section 5.4) for a comprehensive treatment. Since this beginning, similar algorithms have been applied to non-convex environments [5], [6], to dynamic routing with equitable partitioning [7], to robotic networks with limited anisotropic sensory [8] and to coverage with communication constraints [9].

Most of the works cited above assume that a global sensory cost function is known a priori by each agent. Therefore, the focus is limited to the distributed coverage control problem. However, it is often unrealistic to assume such function to be known. For instance, consider a group of underwater vehicles whose main goal is to monitor areas which present a higher concentration of pollution. The distribution of pollution is not known in advance, but vehicles are provided with sensors that can take noisy measurements of it. In this context, coverage control is much harder since the vehicles has to simultaneously explore the environment to estimate pollution distribution and to move to areas with higher pollution concentrations. This is a classical robotic task often referred to as *coverage-estimation* problem. In [10], an adaptive strategy is proposed to solve it but the agents are assumed to take an uncountable number of noiseless measurements. Moreover, the authors used a parametric approach with the assumption that the true function belongs to such class. More recently, [11] proposed a non parametric approach based on Markov Random Fields for adaptive sampling and function estimation. This approach has the advantage to provide better approximation of the underlying sensory function as well confidence bounds on the estimate.

The novelty of this work is to consider a Bayesian non parametric learning scheme where, under the framework of Gaussian regression [12], the unknown function is modeled as a zero-mean Gaussian random field. Robot coordination control is guaranteed to incrementally improve the estimate of the sensory function and simultaneously achieve asymptotic optimal coverage control. Although robot motion is generated by a centralized station, this work provides a starting point to design coordination algorithm for simultaneous estimation and coverage. Note however that the *robot to base station communication model* adopted in this paper already finds application for ocean gliders interfaces communicating with a tower [13], [14], UAV data mules that periodically visit ground robots [15], or cost-mindful use of satellite or cellular communication.

Classical learning problem consists of estimating a function from examples collected on input locations drawn from a fixed probability density function (pdf) [16], [17]. Recent extensions also replace such pdf with a convergent sequence of probability measures [18]. When performing

A. Carron, M. Todescato, R. Carli, L. Schenato, M. Todescato, G. Pillonetto are with the Department of Information Engineering, University of Padova, Via Gradenigo 6/a, 35131 Padova, Italy {carronan|todescat|carlirug|schenato|giapi} @dei.unipd.it.

978-3-9524269-3-7 ©2015 EUCA

coverage, the stochastic mechanism underlying the input locations establishes how the agents move inside the domain of interest. The peculiarity of our algorithm is that such pdf is allowed to vary over time, depending also on the current estimate of the function. Hence, agents locations consist of a non Markovian process, leading to a learning problem where stochastic adaption may happen infinitely often (with no guarantee of convergence to a limiting pdf). Under this complex scenario, we will derive conditions that ensure statistical consistency of the function estimator both assuming that the Bayesian prior is correct. The paper is so organized. After giving some mathematical preliminaries in Section II, problem statement is reported in Section III. The proposed algorithm is presented in Section IV, with its convergence propriety discussed in Section V. In Section VI some simulations results are reported. Conclusions then end the paper.

II. Mathematical preliminaries

Let $\mathscr{X}$ be a compact and convex polygon in $\mathbb{R}^2$ an let $\|\cdot\|$ denote the Euclidean distance function. Let $\mu : \mathscr{X} \to \mathbb{R}_{>0}$ be a distribution density function defined over $\mathscr{X}$. Within the context of this paper, a *partition* of $\mathscr{X}$ is a collection of N polygons $\mathscr{W} = (W_1, \ldots, W_N)$ with disjoint interiors whose union is $\mathscr{X}$. Given the list of N points in $\mathscr{X}$, $\mathbf{x} = (x_1, \ldots, x_N)$, we define the Voronoi partition $\mathscr{V}(\mathbf{x}) = \{V_1(\mathbf{x}), \ldots, V_N(\mathbf{x})\}$ generated by $\mathbf{x}$ as

$$V_i(\mathbf{x}) = \left\{ q \in \mathscr{X} \mid \|q - x_i\| \leq \|q - x_j\|, \ \forall j \neq i \right\}.$$

For each region V_i, $i \in \{1, \ldots, N\}$, we define its centroid with the respect to the density function μ as

$$c_i(V_i(\mathbf{x})) = \left(\int_{V_i(\mathbf{x})} \mu(q) dq \right)^{-1} \int_{V_i(\mathbf{x})} q \mu(q) dq.$$

We denote by

$$\mathbf{c}(\mathscr{V}(\mathbf{x})) = (c_1(V_1(\mathbf{x})), \ldots, c_N(V_N(\mathbf{x})))$$

the vector of regions centroids corresponding to the Voronoi partition generated by $\mathbf{x} = (x_1, \ldots, x_N)$. A partition is said to be a Centroidal Voronoi partition of the pair $(\mathscr{X}, \mu)$ if , for $i \in \{1, \ldots, N\}$, the point x_i is the centroid of $V_i(\mathbf{x})$.

Given $\mathbf{x} = (x_1, \ldots, x_N)$ and a density function μ we introduce the *Coverage function* $H(\mathbf{x}; \mu)$ defined as

$$H(\mathbf{x}; \mu) = \sum_{i=1}^{N} \int_{V_i(\mathbf{x})} \|q - c_i(V_i(\mathbf{x}))\|^2 \mu(q) dq$$

For a fixed density function μ, it can be shown that the set of local minima of $H(\mathbf{x}; \mu)$ is composed by the points $\mathbf{x} = (x_1, \ldots, x_N)$, which are such that $x_1, \ldots, x_N$ are the centroids of the corresponding regions $V_1(\mathbf{x}), \ldots, V_N(\mathbf{x})$, i.e, $\mathscr{V}(\mathbf{x})$ is a Centroidal Voronoi partition.

A. Coverage Control Algorithm

Let $\mathscr{X}$ be a convex and closed polygon in $\mathbb{R}^2$ and let μ be a density function defined over $\mathscr{X}$. Consider the following optimization problem

$$\min_{\mathbf{x} \in Q^N} H(\mathbf{x}; \mu).$$

The *coverage algorithm* we consider is a version of the classic Lloyd algorithm based on ”centering and partitioning” for the computation of Centroidal Voronoi partitions. Given an initial condition $\mathbf{x}(0)$ the algorithm cycles iteratively the following two steps:

1) computing the Voronoi partition corresponding to the current value of $\mathbf{x}$, namely, computing $\mathscr{V}(\mathbf{x})$;
2) updating $\mathbf{x}$ to the vector $\mathbf{c}(\mathscr{V}(\mathbf{x}))$.

In mathematical terms, for $k \in \mathbb{N}$, the algorithm is described as

$$\mathbf{x}(k+1) = \mathbf{c}(\mathscr{V}(\mathbf{x}(k))). \tag{1}$$

It can be shown [3] that the function $H(\mathbf{x}; \mu)$ is monotonically non-increasing along the solutions of (1) and that all the solutions of (1) converge asymptotically to the set of configurations that generate centroidal Voronoi partitions. It is well known [3] that the set of centroidal Voronoi partitions of the pair $(\mathscr{X}, \mu)$ are the critical points of the coverage function $H(\mathbf{x}; \mu)$.

III. Problem Formulation

Let $\mu : \mathscr{X} \to \mathbb{R}$ be an unknown function modeled as the realization of a zero-mean Gaussian random field with covariance $K : \mathscr{X} \times \mathscr{X} \to \mathbb{R}$. We restrict our attention to radial kernels, i.e. $K(a,b) = h(\| a - b \|)$, such that if $\| a - b \| \leq \| c - d \|$ then $h(\| a - b \|) \leq h(\| c - d \|)$ and $K(x,x) = \lambda, \ \forall x \in \mathscr{X}$.

Assume we are given a central base-station, and N robotic agents each moving in the space $\mathscr{X}$. The function μ is assumed to be unknown to both the agents and the central unit. Each agent $i \in \{1, \ldots, N\}$ is required to have the following basic computation, communication and sensing capabilities:

(C1) agent i can identify itself to the base station and can send information to the base station;
(C2) agent i can sense the function μ in the position it occupies; specifically, if x_i denotes its current position, it can take the noisy measurement

$$y(x_i) = \mu(x_i) + \nu_i,$$

where $\nu \backsim \mathscr{N}(0, \sigma^2)$, independent of the unknown function μ, and all mutually independent.

The base station must have the following capabilities

(C3) it can store all the measurements taken by all the agents;
(C4) it can perform computations of partitions of $\mathscr{X}$;
(C5) it can send information to each robot;
(C6) it can store an estimate $\hat{\mu}$ of the function μ and of the posterior variance.

The ultimate goal of the group of agents and central base-station is twofold:

1) to explore the environment $\mathscr{X}$ through the agents, namely, to provide an accurate estimate $\hat{\mu}$ of the function μ exploiting the measurements taken by the agents;
2) to compute a *good* partitioning of $\mathscr{X}$ using the estimate $\hat{\mu}$.

IV. The algorithm

To achieve the above goal the following *Estimation + Coverage algorithm* (denoted hereafter as EC algorithm) is employed.

Algorithm 1 EC

```
Require: The central base station (CBS) stores in memory all the measurements.
for k = 1,2,... do
    Measurements collection: For $i \in \{1,\dots,N\}$, agent $i$ takes the measurement $y_{i,k}$ and sends it to CBS.
    Estimate update: Based on $\mathbf{x_k}, \mathbf{x_{k-1}}, \dots, \mathbf{x_0}$ and $\{y_{1,s},\dots,y_{N,s}\}_{s=0}^{k}$ CBS computes $\hat{\mu}_k$ and its posterior.
    Trajectory update: Based on $\hat{\mu}_k$ CBS computes $\mathbf{u_k}$ and sends it to agents. Agents update position as $\mathbf{x_{k+1}} = \mathbf{x_k} + \mathbf{u_k}$.
end for
```

Now, introducing the dynamic, we have that for each $k \in \mathbb{N}$ the central base-station stores in memory a partition $\mathscr{W}_k = (W_{1,k},\dots,W_{N,k})$ of $\mathscr{X}$, the corresponding list of centroids $\mathbf{c}_k = (c_{1,k},\dots,c_{N,k})$, the positions of the robots $(x_{1,k},\dots,x_{N,k})$ and all the measurements received up to k by the agents. For $k \in \mathbb{N}$, agent i, $i \in \{1,\dots,N\}$, moves according to the following first-order discrete-time dynamics

$$x_{i,k+1} = x_{i,k} + u_{i,k}$$

where the input $u_{i,k}$ is assigned to agent i by the central base-station. As soon as agent i reaches the new position $x_{i,k+1}$, it senses the function μ in $x_{i,k+1}$ taking the measurement $y_{i,k+1} = \mu(x_{i,k+1}) + \nu_{i,k}$ and it sends $y_{i,k+1}$ to the central base-station. The central base-station, based on the new measurements gathered $\{y_{i,k+1}\}_{i=1}^{N}$ and on the past measurements, computes a new estimate $\hat{\mu}_{k+1}$ of μ; additionally it updates the partition $\mathscr{W}_k$, setting $\mathscr{W}_{k+1} = \mathscr{V}(x_{1,k+1},\dots,x_{N,k+1})$.

The goal is to iteratively update the position of the agents in such a way that, in a suitable metric, $\hat{\mu} \to \mu$ and the coverage function assumes values as small as possible.

In next subsections we will explain how the central base-station updates the estimate $\hat{\mu}$ based on the measurements collected from the agents, and how it designs the control inputs to drive the trajectories of the agents. It is quite intuitive that in order to have a better and better estimate of the function μ, the measurements have to be taken to reduce as much as possible a functional of the posterior variance, in particular we will adopt the maximum of the posterior variance. To do so, in the first phase of the EC algorithm the agents will be spurred to explore the environment toward the regions which have been less visited. When the error-covariance of the estimate $\hat{\mu}$ is small enough everywhere, the central base-station will update the agents' position to reduce as much as possible the value of the coverage function.

To simplify the notation let us introduce

$$z_{i,k} = \{x_{i,k}, y_{i,k}\}, \quad i = 1,\dots,N.$$

One of the key aspects of the algorithm is related to the agents movement, which establishes how positions $x_{i,k}$ are generated. In particular, as clear in the sequel, each $x_{i,k}$ is a non Markovian process, depending on the whole past history $z_{i,1},\dots,z_{i,k-1}$, $i = 1,\dots,N$. It is useful to describe first the function estimator, then detailing the agent dynamics.

A. Function estimate and posterior variance

Hereby, we use $Z_{N,t}$ to denote the set $\{z_{i,k}\}$ with $i = 1,\dots,N$ and $k = 1,\dots,t$. The agents movements are assumed to be regulated by probability densities fully defined by $Z_{N,t}$. It comes [19], [20] that the minimum variance estimate of μ given $Z_{N,t}$ is

$$\begin{aligned}\hat{\mu}_t(x) &= \mathbb{E}\left[\mu(x)|Z_{N,t}\right] \\ &= \sum_{i=1}^{N}\sum_{k=1}^{t} c_i K(x_{i,k},\cdot) \end{aligned} \tag{2}$$

where

$$\begin{bmatrix} c_1 \\ \vdots \\ c_N \end{bmatrix} = (\bar{K} + \sigma^2 \mathbb{I})^{-1} \begin{bmatrix} y_{1,1} \\ \vdots \\ y_{N,t} \end{bmatrix}$$

and

$$\bar{K} = \begin{bmatrix} K(x_{1,1},x_{1,1}) & \dots & K(x_{1,1},x_{N,t}) \\ \vdots & & \vdots \\ K(x_{N,t},x_{1,1}) & \dots & K(x_{N,t},x_{N,t}) \end{bmatrix}.$$

The a posteriori variance of the estimate, in a generic input location $x \in \mathscr{X}$, is

$$\begin{aligned} V(x) = \mathrm{Var}\left[\mu(x)|Z_{N,t}\right] = K(x,x) - \\ \begin{bmatrix} K(x_{1,1},x) & \dots & K(x_{N,t},x) \end{bmatrix} (\bar{K} + \sigma^2 \mathbb{I})^{-1} \begin{bmatrix} K(x_{1,1},x) \\ \vdots \\ K(x_{N,t},x) \end{bmatrix}. \end{aligned} \tag{3}$$

B. Description of agents dynamics

The generation of the control input can be divided in two phases: in the first, estimation and coverage are carried out together, while, when the estimate is good enough, i.e. the posterior variance is uniformly small, automatically the control switches to the second phase, where the standard coverage control algorithm reviewed in Section II-A is deployed.

1) Phase I: Let

$$u_{i,k} = \begin{bmatrix} \Re e(\rho_i e^{j\theta_i}) \\ \Im m(\rho_i e^{j\theta_i}) \end{bmatrix} \qquad i = 1, \dots, N \tag{4}$$

then the agents dynamics, for $i = 1, \dots, N$, are defined by

$$x_{i,k+1} = \begin{cases} x_{i,k} + u_{i,k} & \text{if } x_{i,k} + u_{i,k} \in \mathscr{X} \\ x_{i,k} & \text{if } x_{i,k} + u_{i,k} \notin \mathscr{X} \end{cases}$$

Hence, variation of the agent's position is given by the random vector $\rho_i e^{j\theta_i}$, where θ_i is a random variable on $[0, 2\pi]$, determining the movement's direction, while ρ_i is another random variable establishing the step length. The peculiarities of our approach are the following ones:

- the statistics of (θ_i, ρ_i) vary over time and depend on the past history through the estimate $\hat{\mu}_k$ and a function $a(\cdot)$ of the maximum of its posterior variance, i.e.

  $$a\left(\max_{x \in \mathscr{X}} V(x)\right).$$

  Note that a varies over time since it depends on the posterior variance which also varies over time as the agents move over $\mathscr{X}$. Hereby, to simplify notation, we use $a(k)$ to stress this dependence. In this way, at every k, a suitable trade-off is established between centroids targeting, which are never perfectly known, being function of μ, and the need of reducing their uncertainty. These two goals are called *exploration* and *exploitation* in [10];
- the probability densities of θ_i and ρ_i are assumed to be uniformly bounded below. This means that, irrespective of the particular agent's position and instant k, there exists $\varepsilon_\ell > 0$ such that every set of Lebesgue measure $\ell > 0$ can be reached in one step with probability greater than $\ell\varepsilon_\ell$.

Example 1: We provide a concrete example by describing the specific update rule adopted during the numerical experiments reported in section VI. The random variable ρ_i is a truncated Gaussian, constrained to assume positive values, while θ_i is a bimodal Gaussian with support limited to the interval $[0, 2\pi]$. More specifically, for $i = 1, \dots, N$, the density of θ_i is

$$p(\theta_i) = \begin{cases} \frac{1-a(k)}{b_i(k)} e^{-\frac{(\theta_i - \theta_{C_i}(k))^2}{\sigma^2_{C_i}}} + \frac{a(k)}{c_i(k)} e^{-\frac{(\theta_i - \theta_{\Delta_i}(k))^2}{\sigma^2_{\Delta_i}}}, & \theta_i \in [0, 2\pi] \\ 0, & \theta_i \notin [0, 2\pi] \end{cases}$$

where

$$b_i(k) = \int_0^{2\pi} e^{-\frac{(\theta_i - \theta_{C_i}(k))^2}{\sigma^2_{C_i}}} d\theta_i \quad , \quad c_i(k) = \int_0^{2\pi} e^{-\frac{(\theta_i - \theta_{\Delta_i}(k))^2}{\sigma^2_{\Delta_i}}} d\theta_i$$

where

- $\theta_{C_i}(k)$ determines the direction to follow at instant k to reach the current estimate of the Voronoi centroid of the agent i computed using $\hat{\mu}_k$ as defined in (2);
- θ_{Δ_i} determines the direction given by the gradient of the posterior variance (3) computed at the input location occupied by the $i-th$ agent at the instant k;
- $a(k) \in [0, 1]$ is a control parameter that establishes the trade-off between exploration and exploitation at instant k. In the next section an automatic way to tune this parameter based on the posterior variance will be presented;
- $\sigma^2_{C_i}, \sigma^2_{\Delta_i}$ determine the level of dispersion of the density around the directions given by θ_{C_i} and θ_{Δ_i}.

A simple heuristic that allows to automatically determine the value of a is based on the maximum of the posterior variance, with the constraint that a has to satisfy the following conditions:

1) $a(k)$ has to be continuous as function of the maximum of the posterior,
2) $a(k)$ has to be monotonically increasing with the maximum of the posterior,
3) if $\max_{x \in \mathscr{X}} V(x) = \lambda$ then $a(k) = 1$,
4) if $\max_{x \in \mathscr{X}} V(x) = 0$ then $a(k) = 0$.

Two examples are reported in Figure 1.

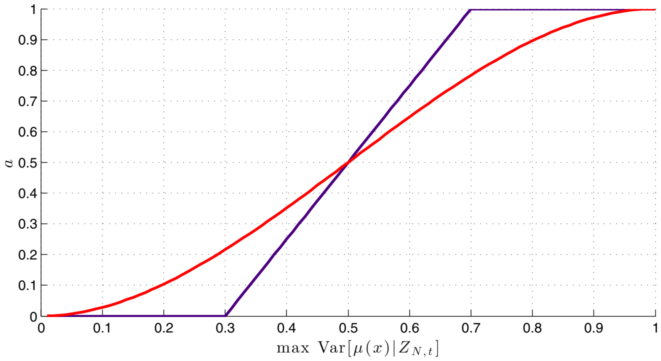

Fig. 1. Here are reported two examples of $a(k)$. In this case the $\max_{x \in \mathscr{X}} V(x) = \lambda = 1$.

At the beginning, being the posterior variance large, $a(k)$ will be close to 1 and the agents will just explore the domain. Thanks to the monotonicity of $a(k)$, while the maximum of the posterior will be reduced, also $a(k)$ will be reduced and consequently the agents will privilege the coverage. ■

2) Phase II: When $a(k)$ is under a certain threshold, i.e. the posterior variance is uniformly low, the control input switches from the update rule described in section IV-B.1 to $u_{i,k} = -(x_{i,k} - c_{i,k})$, so the agents will directly reach the estimated centroids. In other words, in this phase the Lloyd's algorithm is performed with the unknown function set to the estimate obtained at the end of the first phase.

V. Convergence properties of the algorithm

It is important to verify that (in probability) the posterior variance can be reduced as much as we want. Indeed, this fact implies that (with probability one) the agents dynamics will switch from phase I to phase II. The following result holds.

Proposition 2: Let μ be a zero-mean Gaussian random field of radial covariance K. Then, $\forall \varepsilon \geq 0, \forall \delta \in (0, 1]$ there exists t_0 such that, $\forall t \geq t_0$, one has:

$$Pr\left[\max_{x\in\mathscr{X}} \text{ Var } (\mu(x)|Z_{N,t}) \leq \varepsilon\right] \geq 1-\delta$$

Proof: Consider the following inequality

$$\lambda - \frac{(\lambda-\alpha)^2}{\lambda+\frac{\sigma^2}{m}} \leq \varepsilon.$$

Then, we can always choose a pair $\bar{\alpha}$ and $\bar{m}$ such that the previous inequality holds. By the continuity of the kernel, there exists a partition, function of $\bar{\alpha}$, given by all the subset $\mathscr{D}_j \subseteq \mathscr{X}$ such that $K(x,x^*) \geq \lambda - \bar{\alpha}$, $\forall x,x^* \in \mathscr{D}_j$. For a sufficiently large t, with a probability greater then $1-\delta$, we can collect $\bar{m}$ or more measurements in each $\mathscr{D}_j$. In fact, $\forall A \subseteq \mathscr{X}$ and $\forall x_1 \in \mathscr{X}$, $Pr[x(k+1) \in A | x(k) = x_1] \geq \varepsilon \ell_A$, where ℓ_A is the Lebesgue measure of A, since the probability densities of θ_i and ρ_i are bounded below.

Now it is not restrictive consider only $\bar{m}$ measurements falling in $\mathscr{D}_j$, which are denoted by $z_1^j,\dots,z_{\bar{m}}^j$ and collected on the input locations $x_1^j,\dots,x_{\bar{m}}^j$. Calling $\bar{K}_j$ the sampled kernel in the input location falling in $\mathscr{D}_j$ and thanks to the fact that $Tr(\bar{K}_j) = \sum \Lambda(\bar{K}_j) = m\lambda$ (where $\Lambda(\bar{K}_j)$ is the set of eigenvalues of $\bar{K}_j$) and that all the eigenvalues of $\bar{K}_j$ are real and non negative ($\bar{K}_j$ is symmetric and semi positive definite), it holds that $\bar{K}_j \preceq \bar{m}\lambda\mathbb{I}$ so that

$$(\bar{K}_j+\sigma^2) \preceq (\bar{m}\lambda+\sigma^2)\mathbb{I} \Rightarrow (\bar{K}_j+\sigma^2)^{-1} \succeq (\bar{m}\lambda+\sigma^2)^{-1}\mathbb{I}.$$

So with probability greater then $1-\delta$ it is true that

$$\text{Var}\left[\mu(x)|z_i^j,\dots,z_{\bar{m}}^j\right]$$

$$= K(x,x) - \begin{bmatrix} K(x_1^j,x) & \dots & K(x_{\bar{m}}^j,x)\end{bmatrix} (\bar{K}_j+\sigma^2\mathbb{I})^{-1} \begin{bmatrix} K(x_1^j,x) \\ \vdots \\ K(x_{\bar{m}}^j,x)\end{bmatrix}$$

$$\leq \lambda - \frac{\sum_{h=1}^{\bar{m}} K(x_h^j,x)^2}{\bar{m}\lambda+\sigma^2} \leq \lambda - \frac{\bar{m}(\lambda-\bar{\alpha})^2}{\bar{m}\lambda+\sigma^2} = \lambda - \frac{(\lambda-\bar{\alpha})^2}{\lambda+\frac{\sigma^2}{\bar{m}}} \leq \varepsilon$$

thus proving the statement. ■

The consequence of Proposition 2 is that with probability one there exists a time $\bar{k}$ such that the agents dynamics switch from phase I to phase II, namely the agents dynamics will be ruled by

$$\mathbf{x}_{k+1} = \mathbf{c}(\mathscr{V}(\mathbf{x}(k))) \qquad (5)$$

for $k > \bar{k}$, where the centroids are computed according to the estimate $\hat{\mu}_{\bar{k}}$.

Proposition 3: The trajectory generated by 5 converges to the set of configurations that generate centroidal Voronoi partitions of the pair $(\mathscr{X}, \hat{\mu}_{\bar{k}})$.

VI. Numerical Results

In this section, we provide some simulations implementing the new estimation and coverage algorithm. We consider a team of $N = 8$ agents placed, with a random initial position, in the domain $\mathscr{X} = [0,1] \times [0,1]$. Moreover, we use the Gaussian kernel

$$K(x,x') = e^{-\frac{\|x-x'\|^2}{0.02}}$$

with the estimator and the posterior variance given by (2) and (3), respectively. The unknown sensory function μ is a combination of four bi-dimensional Gaussian:

$$\mu(x) = 20\left(e^{\frac{-\|x-\mu_1\|^2}{0.04}} + e^{\frac{\|x-\mu_2\|^2}{0.04}}\right) + 5\left(e^{-\frac{\|x-\mu_3\|^2}{0.04}} + e^{-\frac{\|x-\mu_4\|^2}{0.04}}\right),$$

where

$$\mu_1 = \begin{bmatrix}0.2\\0.2\end{bmatrix} \quad \mu_2 = \begin{bmatrix}0.8\\0.8\end{bmatrix} \quad \mu_3 = \begin{bmatrix}0.8\\0.2\end{bmatrix} \quad \mu_4 = \begin{bmatrix}0.2\\0.8\end{bmatrix}.$$

For computational reasons, the function μ and the posterior variance are evaluated over a grid of step 0.05. The two parameters $\sigma_{\Delta_i}^2$ and $\sigma_{C_i}^2$ are both set to 0.1 and the threshold that allows to switch from phase I to phase II is equal to 0.3. The adopted $a(k)$ is as described in Example 1. This means that, when the maximum of the posterior is large, the value of $a(k)$ is also large to allow a good estimation. Instead, when the maximum of the posterior variance becomes small, also the value of $a(k)$ is reduced to favor agents movement towards the centroids.

An example is in Figure 2 which displays the posterior variance (contour plot), the gradient (quiver plot) and the agents (red diamonds). The figure illustrates results from the first iteration (just because the plot is more clear).

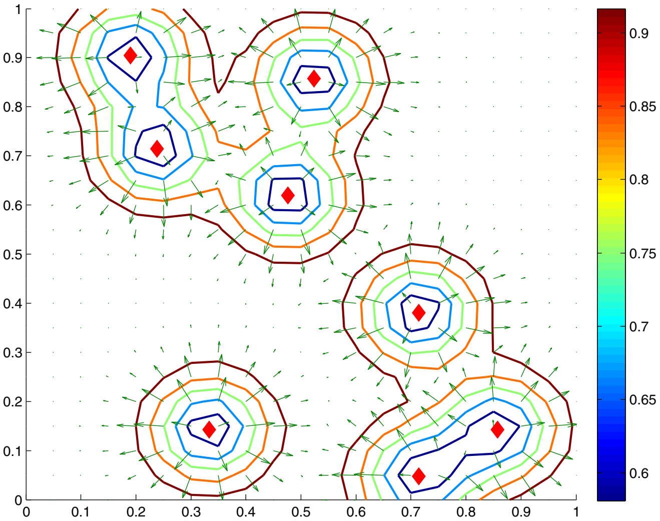

Fig. 2. Contour plot of the posterior variance with the directions of the gradient. The red diamonds are the agents.

Figure 3 plots the profile of the maximum, the average and the minimum of the posterior, as a function of the number of iterations. Finally, Figure VI reports the Voronoi regions associated with the agents, as well as the estimated function $\hat{\mu}(x)$ (contour plot). The final agents positions (red circles) are close to the ideal agents positions, computed using the true sensory function μ (black circles).

VII. Conclusions

We have proposed a new algorithm to perform simultaneously estimation and coverage. The sensory function is seen as a Gaussian random field which has to be reconstructed in an on line manner. A set of control inputs establish the agents

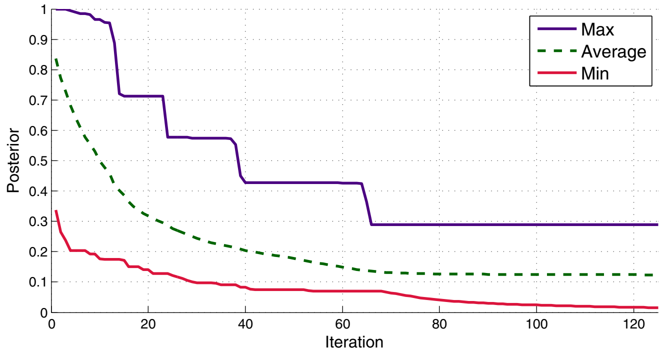

Fig. 3. Evolution of the maximum, the average $\left(\frac{1}{\int_{\mathscr{X}} dx}\int_{\mathscr{X}} V(x)dx\right)$ and the minimum of the posterior over the time.

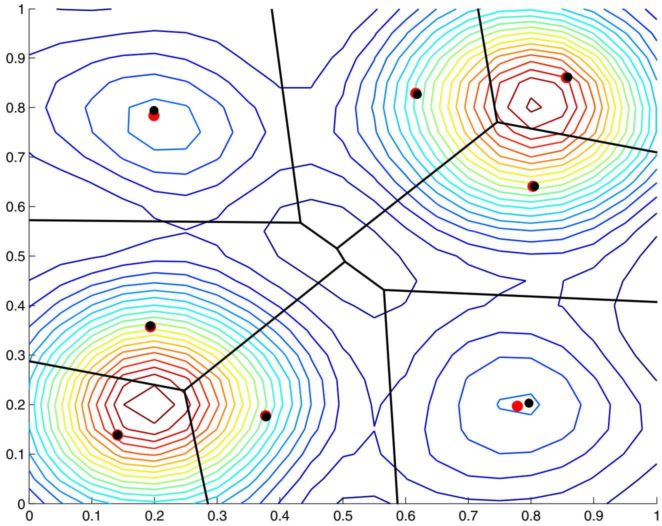

Fig. 4. Voronoi regions of the agents with contour plot of the density function. The red circles are the agents positions, the black circles are the centroids positions computed using μ instead of $\hat{\mu}_{\bar{k}}$.

movement, trying to balance coverage and estimation. We have seen that the resulting problem is also an instance of a non standard function learning problem where input locations follow a non Markovian process with stochastic adaption allowed to happen infinitely often. Numerical experiments show good performance. Even if the centralized algorithm finds many applications in different fields, such as [13] and [15], we are also working on a distributed version. The core of the algorithm is based on the on-line non-parametric regression studied in [21]. In addition, we also plan to provide a distributed algorithm possibly also accounting for time variance of the sensory function.

References

[1] S. P. Lloyd, "Least-squares quantization in pcm," *IEEE Transactions on Information Theory*, vol. 28, pp. 129–137, 1982.

[2] Q. Du, V. Faber, and M. Gunzburger, "Centroidal voronoi tessellations: Applications and algorithms," *SIAM Review*, vol. 41, pp. 637–676, 1999.

[3] J. Cortes, S. Martinez, T. Karatas, and F. Bullo, "Coverage control for mobile sensing networks," *Automatica*, vol. 20, no. 2, pp. 243–255, 2004.

[4] F. Bullo, J. Cortes, and S. Martinez, *Distributed Control of Robotic Networks*. Applied Mathematics Series, Princenton University Press, 2009.

[5] M. Zhong and C. G. Cassandras, "Distributed coverage control in sensor network environments with polygonal obstacles," in *IFAC World Congress*, 2008, pp. 4162–4167.

[6] L. C. A. Pimenta, V. Kumar, R. C. Mesquita, and G. A. S. Pereira, "Sensing and coverage for a network of heterogeneous robots," in *IEEE Conference on Decision and Control*, 2008, pp. 3947–3952.

[7] O. Baron, O. Berman, D. Krass, and Q. Wang, "The equitable location problem on the plane," *European Journal of Operational Research*, vol. 183, no. 2, pp. 578–590, 2007.

[8] K. Laventall and J. Cortes, "Coverage control by multi-robot networks with limited range anisotropic sensory," *International Journal of Control*, vol. 86, no. 6, 2009.

[9] F. Bullo, R. Carli, and P. Frasca, "Gossip coverage control for robotic networks : dynamical systems on the space of partitions," *SIAM Journal on Control and Optimization*, vol. 50, no. 1, 2012.

[10] M. Schwager, D. Rus, and J.-J. Slotine, "Decentralized, adaptive coverage control for networked robots," *Int. J. Rob. Res.*, vol. 28, no. 3, pp. 357–375, Mar. 2009.

[11] J. Choi and R. Horowitz, "Learning coverage control of mobile sensing agents in one-dimensional stochastic environments," *Automatic Control, IEEE Transactions on*, vol. 55, no. 3, pp. 804–809, March 2010.

[12] C. Rasmussen and C. Williams, *Gaussian Processes for Machine Learning*. The MIT Press, 2006.

[13] A. Pereira, H. Heidarsson, C. Oberg, D. A. Caron, B. Jones, G. S. Sukhatme, A. Pereira, C. Oberg, and G. S. Sukhatme, "A communication framework for cost-effective operation of auvs in coastal regions," in *In The 7th International Conference on Field and Service Robots*, 2009.

[14] J. Durham, R. Carli, P. Frasca, and F. Bullo, "Dynamic partitioning and coverage control with asynchronous one-to-base-station communication," in *IEEE International Conference on Decision and Control 2011*.

[15] R. Shah, S. Roy, S. Jain, and W. Brunette, "Data mules: modeling a three-tier architecture for sparse sensor networks," in *Sensor Network Protocols and Applications, 2003. Proceedings of the First IEEE. 2003 IEEE International Workshop on*, May 2003, pp. 30–41.

[16] T. Poggio and F. Girosi, "Networks for approximation and learning," in *Proceedings of the IEEE*, vol. 78, 1990, pp. 1481–1497.

[17] S. Smale and D. Zhou, "Learning theory estimates via integral operators and their approximations," *Constructive Approximation*, vol. 26, pp. 153–172, 2007.

[18] ——, "Online learning with markov sampling," *Analysis and Applications*, vol. 07, no. 01, pp. 87–113, 2009.

[19] A. Tikhonov and V. Arsenin, *Solutions of Ill-Posed Problems*. Washington, D.C.: Winston/Wiley, 1977.

[20] F. Cucker and S. Smale, "On the mathematical foundations of learning," *Bulletin of the American mathematical society*, vol. 39, pp. 1–49, 2001.

[21] D. Varagnolo, G. Pillonetto, and L. Schenato, "Distributed parametric and nonparametric regression with on-line performance bounds computation," *Automatica*, vol. 48, no. 10, pp. 2468 – 2481, October 2012.