2009 American Control Conference
Hyatt Regency Riverfront, St. Louis, MO, USA
June 10-12, 2009



# Distributed sampling of random fields with unknown covariance

Rishi Graham Jorge Cortés

*Abstract*— **This paper considers robotic sensor networks performing spatial estimation tasks. We model a physical process of interest as a spatiotemporal random field with mean unknown and covariance known up to a scaling parameter. We design a distributed coordination algorithm for an heterogeneous network composed of mobile agents that take point measurements of the field and static nodes that fuse the information received from the agents and compute directions of maximum descent of the estimation uncertainty. The technical approach builds on a novel reformulation of Bayesian sequential field estimation, and combines tools from distributed linear iterations, nonlinear programming, and spatial statistics.**

## I. Introduction

Networks of environmental sensors are playing an increasingly important role in scientific studies of the ocean, rivers, and the atmosphere. Envisioned tasks include pollutant detection, fire monitoring, and mapping of ocean currents. Mobile sensing robots can improve the efficiency of data collection, adapt to changes in the environment, and provide a robust response to sensor failures. Complex statistical techniques come into play in the analysis of environmental processes. Consequently, the operation of robotic sensors must be driven by statistically-aware algorithms that make the most of the network capabilities for data collection and fusion. At the same time, such algorithms need to be distributed and scalable to make robotic networks capable of operating in an autonomous and robust fashion. The combination of these two objectives, complex statistical modeling and distributed coordination, presents grand technical challenges: traditional statistical modeling and inference assume full availability of all measurements and central computation. While the availability of data at a central location is certainly a desirable property, the paradigm for motion coordination builds on partial, fragmented information. This work is a step forward in bridging the gap between sophisticated statistical modeling and distributed motion coordination.

*Literature review:* Complex statistical techniques allow a detailed account of uncertainty in modeling physical phenomena. Of particular relevance to this work are [1], [2], regarding statistical models, and [3], [4], regarding the application of optimal design techniques to Bayesian models. Under certain conditions on the covariance structure, data taken far from the prediction site have very little impact on the predictor [5]. When the random field does not have a covariance structure with finite spatial correlation, an approximation which does may be generated via covariance tapering [6]. Optimal design [7], [8] addresses the problem of choosing sample locations which optimize estimation.

Rishi Graham is with the Department of Applied Mathematics and Statistics, University of California, Santa Cruz, rishig@ams.ucsc.edu

Jorge Cortés is with the Department of Mechanical and Aerospace Engineering, University of California, San Diego, cortes@ucsd.edu

In cooperative control, various works consider mobile sensor networks performing spatial estimation tasks. [9] introduces performance metrics for oceanographic surveys by autonomous underwater vehicles. [10] chooses optimal sampling trajectories from a parameterized set of paths. In [11], [12], [13] the focus is on estimating deterministic fields with random measurement noise. When the physical process is not as well understood, or accurate deterministic models require high dimensional parameter spaces, random field models can be a useful alternative. In previous work [14], we have considered the estimation of random fields with known covariance. In this paper, we focus on the additional complexity in the algorithm design caused by unknown parameters in the field covariance.

*Statement of contributions:* We begin with a widely accepted Bayesian model for the prediction of a spatiotemporal random field with mean unknown and covariance known up to a scaling parameter. The predictive variance of this model can be written as a scaled product of two components, one corresponding to uncertainty about the covariance of the field, the other corresponding to uncertainty of the prediction conditional on the covariance. Our first contribution is the development of a novel procedure for distributed calculation of the first component sequentially as new measurements arrive. We also introduce an upper bound for the second component which can be calculated in a distributed way. These two results allow us to identify an objective function for gathering data which minimizes uncertainty in the resulting estimation. Our second contribution is the characterization of the smoothness properties of the objective function and the computation of its gradient. Using consensus and distributed Jacobi overrelaxation algorithms, we show how the objective function and its gradient can be computed in a distributed way across a network composed of robotic agents and static nodes. Our third contribution is the design of a coordination algorithm based on projected gradient descent which guarantees one-step-ahead locally optimal data collection.

## II. Preliminary notions

Let $\mathbb{R}$, $\mathbb{R}_{>0}$, and $\mathbb{R}_{\geq 0}$ denote the set of reals, positive reals and nonnegative reals, respectively. For $p \in \mathbb{R}^d$ and $r \in \mathbb{R}_{>0}$, let $\overline{B}(p, r)$ be the *closed ball* of radius $r$ centered at $p$. Given $\underline{u} = (u_1, \ldots, u_a)^T$, $a \in \mathbb{Z}_{>0}$, and $\underline{v} = (v_1, \ldots, v_b)^T$, $b \in \mathbb{Z}_{>0}$, we denote by $(\underline{u}, \underline{v})$ the concatenation $(\underline{u}, \underline{v}) = (u_1, \ldots, u_a, v_1, \ldots, v_b)^T$. We denote by $\partial S$ the boundary of a set $S$. The *$\epsilon$-contraction* of a set $S$, with $\epsilon > 0$, is the set $S_\epsilon = \{q \in S \mid \mathrm{d}(q, \partial S) \geq \epsilon\}$. A *convex polytope* is the convex hull of a finite point set. For a bounded set $S \subset \mathbb{R}^d$, we let $\mathrm{CR}(S)$ denote the *circumradius* of $S$, that is, the radius of the smallest-radius $d$-sphere enclosing $S$. We denote by $\mathbb{F}(S)$ the collection of finite subsets of $S$.

978-1-4244-4524-0/09/$25.00 ©2009 AACC

We consider a convex polytope $\mathcal{D} \subset \mathbb{R}^d$, $d \in \mathbb{N}$. Let $\mathcal{D}_e = \mathcal{D} \times \mathbb{R}$ denote the space of points over $\mathcal{D}$ and time. The *Voronoi partition* $\mathcal{V}(\underline{s}) = (V_1(\underline{s}), \ldots, V_n(\underline{s}))$ of $\mathcal{D}$ generated by the points $\underline{s} = (s_1, \ldots, s_n)$ is defined by $V_i(\underline{s}) = \{q \in \mathcal{D} \mid \|q - s_i\| \leq \|q - s_j\|, \ \forall j \neq i\}$. Each $V_i(\underline{s})$ is called a *Voronoi cell*. Two points $s_i$ and $s_j$ are *Voronoi neighbors* if their Voronoi cells share a boundary.

### A. Bayesian modeling of space-time processes

Let $Z$ denote a random space-time process taking values on $\mathcal{D}_e$. Let $\underline{y} = (y_1, \ldots, y_m)^T \in \mathbb{R}^m$ be $m \in \mathbb{N}$ measurements taken from $Z$ at corresponding locations $\underline{x} = (x_1, \ldots, x_m)^T \in \mathcal{D}_e^m$, with $x_i = (s_i, t_i), \ i \in \{1, \ldots, m\}$. Given these data, various models allow for prediction of $Z$ at any point in $\mathcal{D}_e$, with associated uncertainty.

In a Bayesian setting, the prediction takes the form of a distribution, called the posterior predictive [15]. If the field is modeled as a Gaussian process with known covariance, the posterior predictive mean corresponds to the *Best Linear Unbiased Predictor*, and its variance corresponds to the mean-squared prediction error. If the covariance of the field is not known, however, few analytical results exist which take the full uncertainty into account. The model we present here [2] allows for uncertainty in the covariance process and still produces an analytical posterior predictive distribution.

We assume that the measurements are distributed as

$$\underline{y} \sim \mathrm{N}_m \left( \mathbf{F}^T \beta, \sigma^2 \mathbf{K} \right). \tag{1}$$

Here $\beta \in \mathbb{R}^p$ is a vector of unknown regression parameters, $\sigma^2 \in \mathbb{R}_{>0}$ is the unknown variance parameter, and $\mathbf{K}$ is a correlation matrix whose $i,j$th element is $\mathbf{K}_{ij} = \mathrm{Cor}[y_i, y_j]$. We assume a finite correlation range in space, $r \in \mathbb{R}$, such that if $\|s_i - s_j\| \geq r$, then $\mathbf{K}_{ij} = \mathbf{K}_{ji} = 0$. The matrix $\mathbf{F} \in \mathbb{R}^{p \times m}$ is determined by a set of $p \in \mathbb{N}$ known basis functions $f_i : \mathcal{D}_e \to \mathbb{R}$ evaluated at the locations $\underline{x}$. We assume conjugate prior distributions for the parameters,

$$\beta | \sigma^2 \sim \mathrm{N}_p \left( \beta_0, \sigma^2 \mathbf{K}_0 \right) \tag{2a}$$

$$\sigma^2 \sim \Gamma^{-1} \left( \frac{\nu}{2}, \frac{q\nu}{2} \right). \tag{2b}$$

Here $\beta_0 \in \mathbb{R}^p$, $\mathbf{K}_0 \in \mathbb{R}^{p \times p}$, and $q, \nu \in \mathbb{R}_{>0}$ are constants, known as *tuning parameters* for the model, and $\Gamma^{-1}(a, b)$ denotes the inverse gamma distribution with shape parameter $a$ and scale parameter $b$ (see, e.g. [16]).

**Proposition II.1 (Posterior predictive distribution [2])** *Under the Bayesian model* (1)*, the posterior predictive at location $x_0 \in \mathcal{D}_e$ is a shifted Students t distribution (see, e.g. [16]) with $\gamma = \nu + m + 1$ degrees of freedom and variance,* $\mathrm{Var}[Z | \underline{y}, \underline{x}] = \frac{\varphi(\underline{y}, \underline{x})}{\gamma} \phi(x_0; \underline{x})$*, where,*

$$\begin{aligned}
\phi(x_0; \underline{x}) &= \mathrm{Cor}[Z, Z] - \mathbf{k}^T \mathbf{K}^{-1} \mathbf{k} + \xi_0^T \left( \mathbf{K}_0^{-1} + E \right)^{-1} \xi_0 \\
\xi_0 &= f(x_0) - \mathbf{F} \mathbf{K}^{-1} \mathbf{k} \\
\varphi(\underline{y}, \underline{x}) &= q\nu + \frac{1}{2} \left( \underline{y} - \mathbf{F}^T \hat{\beta} \right)^T \mathbf{K}^{-1} \left( \underline{y} - \mathbf{F}^T \hat{\beta} \right) + \\
&\quad + \frac{1}{2} \left( \hat{\beta} - \beta_0 \right)^T \left( \mathbf{K}_0 + E^{-1} \right)^{-1} \left( \hat{\beta} - \beta_0 \right),
\end{aligned}$$

*with $\hat{\beta} = E^{-1} \mathbf{F} \mathbf{K}^{-1} \underline{y}$, $E = \mathbf{F} \mathbf{K}^{-1} \mathbf{F}^T$, and $\mathbf{k} = \mathrm{Cor}[\underline{y}, Z]$.*

## III. Problem statement

Here we introduce the model for the group of robotic agents and static nodes, and detail the overall objective.

### A. Robotic sensor network model

Consider a group $\{S_1, \ldots, S_m\}$ of $m \in \mathbb{N}$ static nodes at locations $Q = (q_1, \ldots, q_m) \in \mathcal{D}^m$. Assume that each node has a limited communication radius, $R \in \mathbb{R}_{>0}$, and that they are positioned so that each one can communicate with its Voronoi neighbors. In addition to the static nodes, consider a group $\{R_1, \ldots, R_n\}$ of $n$ robotic sensor agents. The position of robot $i \in \{1, \ldots, n\}$ at time $t \in \mathbb{R}$ is denoted by $p_i(t) \in \mathcal{D}$. The robots take point samples of the spatial field at discrete instants of time in $\mathbb{Z}_{\geq 0}$. Between sample instants, each robot moves according to the discrete dynamics

$$p_i(k+1) = p_i(k) + u_i(k),$$

where $\|u_i\| \leq u_{\max}$ for some $u_{\max} \in \mathbb{R}_{>0}$. The communication radius of the robotic agents is also $R$. Each node will need to be able to communicate with any robot which may be within covariance range of the points in its Voronoi region at the following timestep. To that end, we assume that

$$R \geq \max_{i \in \{1, \ldots, m\}} \{\mathrm{CR}(V_i(Q))\} + r + u_{\max}. \tag{3}$$

The robots can sense the positions of other robots within a distance of $2u_{\max}$. At discrete timesteps, each robot communicates the sample and location to static nodes within communication range, along with the locations of any other sensed robots. The nodes then compute control vectors, and relay them back to robots within communication range. The implementation does not require direct communication between robots. We refer to this network model as $\mathcal{N}$.

To avoid agent collision, we further restrict the motion of the robotic agents as follows. Consider the locations $P^{(k)} = (p_1(k), \ldots, p_n(k))^T$. Between timestep $k$ and timestep $k+1$, robot $i$ moves within the region, $\Omega_i^{(k)} \subset \mathcal{D}$ defined by,

$$\Omega_i^{(k)} = (V_i(P^{(k)}))_{\omega/2} \cap \overline{B}(p_i(k), u_{\max}),$$

where $(V_i(P^{(k)}))_{\omega/2}$ denotes the $\omega/2$-contraction of $V_i(P^{(k)})$. This requirement combines the restriction imposed by $u_{\max}$ with a minimum distance requirement such that any two robots are always at least $\omega$ away from each other [14]. Let $\Omega^{(k)} = \prod_{i=1}^n \Omega_i^{(k)} \subset \mathcal{D}^n$ denote the region of allowed movement of all the robotic agents at timestep $k \in \mathbb{N}$.

### B. The average variance as objective function

For predictions over a region in space and time, the average variance is a natural measure of uncertainty, corresponding to A-optimality. We consider the average over the spatiotemporal region of the posterior predictive variance,

$$\mathcal{A} = \frac{1}{\gamma} \varphi(\underline{y}, \underline{x}) \int_{\mathcal{D}} \int_T \phi((y_0, t_0); \underline{x}) \, dt_0 \, dy_0. \tag{4}$$

Here, $\underline{y} \in \mathbb{R}^{k_{\max}}$ is a sequence of samples taken at discrete times $\{1, \ldots, k_{\max}\}$, $k_{\max} \in \mathbb{Z}_{>0}$, at space-time locations $\underline{x} \in (\mathcal{D}_e^n)^{k_{\max}}$. $T = [1, k_{\max}]$ is the time interval of interest.

One would like to choose the sample locations that minimize $\mathcal{A}$. Since samples are taken sequentially, with each

new set restricted to a region nearby the previous, and since $\varphi(\underline{y}, \underline{x})$ depends on the actual values of the samples, one cannot simply optimize over $(\mathcal{D}_e^n)^{k_{\max}}$ a priori.

Consider, instead, a greedy approach in which we use past samples to choose the positions for the next ones. At each timestep we choose the next locations to minimize the average posterior variance of the predictor given the data known so far. In Section IV, we develop a sequential formulation of the average posterior predictive variance and discuss its amenability to distributed implementation over $\mathcal{N}$.

## IV. Distributed criterion for adaptive design

In this section we develop an optimality criterion to maximally reduce the average predictive variance at each timestep. First we reformulate the posterior predictive variance to allow for estimation based on previous sample values. Given centralized computing capabilities, this equation can be used to perform sequential optimal design, but is not amenable to distributed computation. We therefore provide an upper bound whose computation is distributed over $\mathcal{N}$.

### A. Sequential formulation of $\varphi$

At timestep $k$, assume that samples, $\underline{y}_s \in \mathbb{R}^{nk}$ have already been taken at locations $\underline{x}_s \in \mathcal{D}_e^{nk}$. We are interested in choosing *unsampled* locations, $\underline{x}_u \in \mathcal{D}_e^n$ at which to take the next samples, $\underline{y}_u \in \mathbb{R}^n$. Let $\underline{y} = (\underline{y}_u^T, \underline{y}_s^T)^T \in \mathbb{R}^{n(k+1)}$ denote the full set of samples at timestep $k+1$, at locations $\underline{x} = (\underline{x}_u^T, \underline{x}_s^T)^T \in \mathcal{D}_e^{n(k+1)}$. Let $\mathbf{K}_s$ denote the correlation matrix of the vector $\underline{y}_s$, and let $\mathbf{K}_{us} = \mathbf{K}_{su}^T$ denote the matrix whose $(i,j)$th element is the correlation between $y_{u:i}$ and $y_{s:j}$. Once all samples have been taken, the average posterior predictive variance is given by Equation (4). However, $\varphi(\underline{y}, \underline{x})$ cannot be calculated until the new samples are taken. Our approach is to use the generalized least squares estimate, and compute the induced errors in the approximation.

**Proposition IV.1** *Let $\hat{\underline{y}}_{LS} = \mathbf{K}_{us}\mathbf{K}_s^{-1}\underline{y}_s$ be the generalized least squares estimate of $\underline{y}_u$ based on samples $\underline{y}_s$, and let $\overline{\underline{y}}_{LS} = \underline{y}_u - \hat{\underline{y}}_{LS}$. Then we can write,*

$$\varphi(\underline{y}, \underline{x}) = \hat{\varphi}(\underline{y}_s, \underline{x}_s, \underline{x}_u) + \tilde{\varphi}(\underline{y}, \underline{x})\overline{\underline{y}}_{LS},$$

*where $\tilde{\beta} = E^{-1}\mathbf{F}_s\mathbf{K}_s^{-1}\underline{y}_s$ and $\hat{\varphi}(\underline{y}_s, \underline{x}_s, \underline{x}_u) = \hat{\varphi}$ is*

$$\hat{\varphi} = q\nu + \underline{y}_s^T\mathbf{K}_s^{-1}\underline{y}_s - \frac{1}{2}\tilde{\beta}^T E\tilde{\beta} + \frac{1}{2}\left(\tilde{\beta} - \beta_0\right)^T\left(\mathbf{K}_0 + E^{-1}\right)^{-1}\left(\tilde{\beta} - \beta_0\right).$$

In Proposition IV.1, the function $\hat{\varphi}$, which does not depend on the new data, signifies the change in uncertainty about $\sigma^2$ which may be predicted by assuming the generalized least squares estimate $\hat{\underline{y}}_{\text{LS}}$. On the other hand, the quantity $\tilde{\varphi}\overline{\underline{y}}_{\text{LS}}$ denotes the extra uncertainty induced by having made that prediction, once the data $\underline{y}_u$ have been measured.

Using Proposition IV.1, we can rewrite the one-step ahead average prediction variance as follows. Let $\hat{\varphi}^{(k)} : \Omega^{(k)} \to \mathbb{R}$ map the location of the next set of measurements to the value of $\hat{\varphi}$ at timestep $k$. Let $\phi^{(k)} : \mathcal{D}_e \times \Omega^{(k)} \to \mathbb{R}$ map predictive location and unsampled locations to the conditional variance at timestep $k$. Let $\gamma^{(k)} = \nu + n * (k+1) + 1$, and let $(P, k+1)$ denote the space-time locations at spatial positions $P = (p_1, \ldots, p_n) \in \mathcal{D}^n$ and time $k+1$. To optimize the average posterior predictive variance at the $k+1$st timestep, we choose $P$ to minimize $\mathcal{A}^{(k)} : \mathcal{D}^n \to \mathbb{R}$ defined by

$$\mathcal{A}^{(k)}(P) = \frac{\hat{\varphi}^{(k)}(P)}{\gamma^{(k)}}\int_{\mathcal{D}}\int_T \phi^{(k)}\left((s,t);P)\right)\, dt\, ds. \quad (5)$$

In Section V we will show how $\hat{\varphi}^{(k)}$ can be calculated in a distributed way by $\mathcal{N}$. However, due to dependence on the quantity $\mathbf{k}^T\mathbf{K}^{-1}\mathbf{k}$, the conditional variance, $\phi^{(k)}$, can not. In the next section, we detail an upper bound for $\phi^{(k)}$, which may be computed locally by each node.

### B. Upper bound of the average posterior predictive variance

In [14], we established that the conditional variance can be upper bounded using only a subset of the measurements. Using this result, $\mathcal{A}^{(k)}$ can be upper bounded as follows.

**Proposition IV.2 (Spatial approximation for distributed implementation)** *Let $\phi_j^{(k)} : \mathcal{D}_e \times \Omega^{(k)} \to \mathbb{R}$ denote the value of $\phi^{(k)}$ as calculated with only those measurements correlated to $V_j(Q)$. Let $\tilde{\mathcal{A}}_j^{(k)} : \mathcal{D}^n \to \mathbb{R}$ be defined by*

$$\tilde{\mathcal{A}}_j^{(k)}(P) = \frac{\hat{\varphi}^{(k)}(P)}{\gamma^{(k)}}\int_{V_j(Q)}\int_T \phi_j^{(k)}\left((s,t),P\right)\, dt\, ds.$$

*Then $\mathcal{A}^{(k)} \le \tilde{\mathcal{A}}^{(k)} = \sum_{j=1}^m \tilde{\mathcal{A}}_j^{(k)}$. In addition, equality holds if, for all $j \in \{1, \ldots, m\}$, the samples not used in calculation of $\phi_j^{(k)}$ are uncorrelated to those which are.*

We refer to $\tilde{\mathcal{A}}^{(k)}$ as the *aggregate average prediction variance*. Unlike $\mathcal{A}^{(k)}$, the function $\tilde{\mathcal{A}}^{(k)}$ may be computed in a distributed manner over $\mathcal{N}$.

### C. Smoothness of the aggregate average prediction variance

Next, we characterize the smoothness properties of $\tilde{\mathcal{A}}^{(k)}$. For simplicity, let $\nabla_i$ denote $\frac{\partial}{\partial p_i}$. Given matrix, $A$, we denote by $\nabla_i A$ the component-wise partial derivative of $A$. Assume the ordering $\underline{x} = ((P, k+1), \underline{x}_s) \in (\mathcal{D}_e)^{n*(k+1)}$, so that the $i$th row and column of $\mathbf{K}$, e.g., with $i \le n$, are the correlations between $(p_i, k+1)$ and $\underline{x}$.

**Lemma IV.3** *Assume that $f_1, \ldots, f_p$ and the covariance of $Z$ are $C^1$ with respect to the spatial position of their arguments. Then the map $P \mapsto \phi_j^{(k)}(x_0, P)$ is $C^1$ on $\Omega^{(k)}$ and the $i$th component of its gradient is*

$$\begin{aligned}\nabla_i\phi_j^{(k)} &= -2\mathbf{k}^T\mathbf{K}^{-1}\nabla_i\mathbf{k} + \mathbf{k}^T\mathbf{K}^{-1}\nabla_i\mathbf{K}\mathbf{K}^{-1}\mathbf{k} - \\ &\quad - \xi_0^T\left(\mathbf{K}_0^{-1} + E\right)^{-1}\nabla_i E\left(\mathbf{K}_0^{-1} + E\right)^{-1}\xi_0 + \\ &\quad + 2\xi_0^T\left(\mathbf{K}_0^{-1} + E\right)^{-1}\nabla_i\xi_0, \text{ where} \\ \nabla_i\xi_0 &= -\nabla_i\mathbf{F}\mathbf{K}^{-1}\mathbf{k} - \mathbf{F}\mathbf{K}^{-1}\nabla_i\mathbf{k} + \mathbf{F}\mathbf{K}^{-1}\nabla_i\mathbf{K}\mathbf{K}^{-1}\mathbf{k} \\ \nabla_i E &= \nabla_i\mathbf{F}\mathbf{K}^{-1}\mathbf{F}^T + \mathbf{F}\mathbf{K}^{-1}\nabla_i\mathbf{F}^T - \mathbf{F}\mathbf{K}^{-1}\nabla_i\mathbf{K}\mathbf{K}^{-1}\mathbf{F},\end{aligned}$$

*where the matrices are built with a location vector comprised of an ordering of the samples correlated to $V_j(Q)$.*

It is worth noting that the matrix $\nabla_i\mathbf{F} \in \mathbb{R}^{p\times n(k+1)}$ is nonzero only in column $i$. The matrix $\nabla_i\mathbf{K} \in \mathbb{R}^{n(k+1)\times n(k+1)}$ is nonzero only in row and column $i$.

Additionally, due to the finite correlation range, only those elements corresponding to correlation with other measurement locations $x = (s,t)$ which satisfy $\|p_i - s\| \leq r$ are nonzero.

**Lemma IV.4** *Under the assumptions of Lemma IV.3, assume, in addition, that the partial derivatives of $f_1, \ldots, f_p$ and the covariance of $Z$ are $C^1$ with respect to the spatial position of their arguments. Then the map $P \mapsto \nabla_i \phi_j^{(k)}(x_0, P)$ is globally Lipschitz on $\Omega^{(k)}$.*

Note that the value of $\hat{\varphi}^{(k)}(P)$ depends on $P$ only through the matrix $E$, whose partial derivative is given in Lemma IV.3. This leads us to the following continuity results.

**Lemma IV.5** *Under the assumptions of Lemma IV.3, $\hat{\varphi}^{(k)}$ is $C^1$ on $\Omega^{(k)}$ and the $i$th component of its gradient is $\nabla_i \hat{\varphi}^{(k)}(P) = \sum_{j=1}^m \nabla_i \hat{\varphi}_j^{(k)}(P)$, where,*

$$\nabla_i \hat{\varphi}_j^{(k)}(P) = \frac{1}{2} \Psi^T \nabla_i E \, \Psi, \text{ and}$$
$$\Psi = E^{-1} \left( \mathbf{K}_0 + E^{-1} \right)^{-1} \left( \mathbf{K}_0 E \tilde{\beta} + \beta_0 \right).$$

*Additionally, under the assumptions of Lemma IV.4, $\nabla_i \hat{\varphi}^{(k)}$ is globally Lipschitz on $\Omega^{(k)}$.*

We are finally ready to state the smoothness properties of $\tilde{\mathcal{A}}^{(k)}$ and provide an explicit expression for its gradient.

**Proposition IV.6** *Under the assumptions of Lemma IV.3, $\tilde{\mathcal{A}}^{(k)}$ is $C^1$ on $\Omega^{(k)}$ and the $i$th component of its gradient is*

$$\nabla_i \tilde{\mathcal{A}}^{(k)}(P) = \frac{\hat{\varphi}^{(k)}(P)}{\gamma^{(k)}} \int_{V_j(Q)} \int_T \nabla_i \phi_j^{(k)}\left((s,t), P\right) \, dt \, ds +$$
$$+ \frac{\nabla_i \hat{\varphi}^{(k)}(P)}{\gamma^{(k)}} \int_{V_j(Q)} \int_T \phi_j^{(k)}\left((s,t), P\right)) \, dt \, ds.$$

*Additionally, under the assumptions of Lemma IV.4, $\tilde{\mathcal{A}}^{(k)}$ is globally Lipschitz on $\Omega^{(k)}$.*

## V. Distributed computation of aggregate average prediction variance and its gradient

In this section, we substantiate our assertion that the aggregate average prediction variance and its gradient are distributed over the network $\mathcal{N}$. Since $\mathcal{V}(Q)$ is a partition of the physical space, we may partition all sample locations by region. Thus for each $(s,t) \in i_{\mathbb{F}}(\underline{x})$, there is exactly one $j \in \{1, \ldots, m\}$ such that $s \in V_j(Q)$. In order for the network to calculate $\tilde{\mathcal{A}}^{(k)}$ and its gradient at $P$, it is sufficient for $S_j$ to compute $\tilde{\mathcal{A}}_j^{(k)}$ and $\nabla_i \tilde{\mathcal{A}}_j^{(k)}$ for each robot in $V_j(Q)$. Then $\tilde{\mathcal{A}}^{(k)}$ may be calculated via discrete time average consensus [17], while $\nabla_i \tilde{\mathcal{A}}^{(k)}$ may be calculated from information local to $R_i$. From Propositions IV.2 and IV.6, it can be seen that calculation of $\tilde{\mathcal{A}}_j^{(k)}$ and $\nabla_i \tilde{\mathcal{A}}_j^{(k)}$ requires only local information and the values of $\hat{\varphi}^{(k)}$ and $\nabla_i \hat{\varphi}^{(k)}$.

Next we use consensus and the distributed JOR algorithm [18] to calculate $\hat{\varphi}^{(k)}$ and its gradient. Let $R_{\text{in}}^{(1:k)} : \mathbb{N} \to \mathbb{F}(\mathbb{N})$ map the index of the node to the set of indices of samples whose spatial position lies inside its Voronoi cell, $R_{\text{in}}^{(1:k)}(j) = \{i \in \{1, \ldots, nk\} \mid x_{s:i} = (s,t) \text{ and } s \in V_j(Q)\}$.

With a slight abuse of notation, define $R_{\text{in}}^{(1:k+1)}(j, P)$ to be the equivalent set of indices into the full vector of measurement locations, $\underline{x}$, given future locations $P$.

Our first result illustrates the parts of $\hat{\varphi}^{(k)}$ which do not include the locations $P$. We use the notation $\text{col}_i(M)$ to denote the $i$th column of the matrix $M$.

**Proposition V.1** *Assume that $S_j$ for each $j \in \{1, \ldots, m\}$ knows $x_i, y_i$ for each $i \in R_{in}^{(1:k)}(j)$. After $p+1$ executions of the JOR algorithm and 2 subsequent consensus algorithms, $S_j$ has access to,*

*#1: element $i$ of $\mathbf{K}_s^{-1} \underline{y}_s \in \mathbb{R}$, $i \in R_{in}^{(1:k)}(j)$ via JOR;*
*#2: $col_i\left(\mathbf{F}_s \mathbf{K}_s^{-1}\right) \in \mathbb{R}^p$, $i \in R_{in}^{(1:k)}(j)$ via JOR;*
*#3: $\mathbf{F}_s \mathbf{K}_s^{-1} \underline{y}_s \in \mathbb{R}^p$ via consensus;*
*#4: $\underline{y}_s^T \mathbf{K}_s^{-1} \underline{y}_s \in \mathbb{R}^p$ via consensus;*

Next, we describe calculations which may be done at each step of a gradient descent algorithm at locations $P$.

**Proposition V.2** *Given $P \in \Omega^{(k)}$, assume that $S_j$ for each $j \in \{1, \ldots, m\}$ knows $x_i$ for each $i \in R_{in}^{(1:k+1)}(j, P)$ and the results of Proposition V.1. After p executions of JOR, and $p^2$ of consensus, $S_j$ has access to,*

*#5: $col_i\left(\mathbf{F}\mathbf{K}^{-1}\right) \in \mathbb{R}^p$, $i \in R_{in}^{(1:k+1)}(j, P)$ via JOR;*
*#6: $E \in \mathbb{R}^{p \times p}$ via consensus;*

*After these computations, $S_j$ can calculate $\tilde{\beta}$ and $\nabla_i E$, and subsequently $\hat{\varphi}^{(k)}$ and $\nabla_i \hat{\varphi}^{(k)}$ at $P$ for each robot in $\{i \in \{1, \ldots, n\} \mid p_i \in V_j(Q)\}$.*

## VI. Distributed optimization of the aggregate average predictive variance

Here we outline a distributed version of the projected gradient descent algorithm (see, e.g. [19]), which is guaranteed to converge to a stationary point of $\tilde{\mathcal{A}}^{(k)}$ on $\Omega^{(k)}$. For convenience, let $P_j' : \mathbb{R} \times \mathcal{D}^n \to \mathbb{F}(\mathcal{D})$ map a step size and configuration to the set of next locations calculated by $S_j$,

$$P_j'(\alpha, P) = \Big\{ \text{proj}_{\Omega_i} \left( p_i + \alpha \nabla_i \tilde{\mathcal{A}}(P) \right), \text{ foreach } i \text{ s.t. } \mathrm{d}\left(p_i, V_j(Q)\right) \leq r + u_{\max} + \omega \Big\}.$$

Let $\mathrm{d}_j : \mathbb{R} \times \mathcal{D}^n \to \mathbb{R}_{\geq 0}$ denote the total distance traveled by robots entering $V_j(Q)$, i.e.,

$$\mathrm{d}_j(\alpha, P) = \sum_{\substack{i \in \{1,\ldots,n\} \text{ such that} \\ \text{proj}_{\Omega_i}\left(p_i + \alpha \nabla_i \tilde{\mathcal{A}}(P)\right) \in V_j(Q)}} \| \text{proj}_{\Omega_i} \left( p_i + \alpha \nabla_i \tilde{\mathcal{A}}(P) \right) - p_i \|^2.$$

Globally, let $P' : \mathbb{R} \times \mathcal{D}^n \to \mathcal{D}^n$, $P'(\alpha, P) = \text{proj}_{\Omega}(P + \alpha \nabla \tilde{\mathcal{A}}(P))$. Table I describes a distributed line search with a starting position of $P \in \Omega$. The line search starts with a factor $\alpha_{\max}$ which scales the smallest nonzero partial to $u_{\max}$, ensuring all robots with nonzero partial derivatives can move the maximum distance,

$$\alpha_{\max} = \frac{u_{\max}}{\min\{\|\nabla_i \tilde{\mathcal{A}}(P)\| \mid \nabla_i \tilde{\mathcal{A}}(P) \neq 0\}}. \tag{6}$$

We are ready to present our technique for a greedy optimization algorithm. At timestep $k$, the nodes follow a

| | |
|---|---|
| **Name:** | DISTRIBUTED LINE SEARCH ALGORITHM |
| **Goal:** | Compute step size for gradient descent of $\tilde{\mathcal{A}}^{(k)}$ |
| **Input:** | Configuration, $P = (p_1, \ldots, p_n) \in \mathcal{D}^n$ |
| **Assumes:** | (i) Connected network of static nodes<br>(ii) $S_j$ knows $p_i$, $\tilde{\mathcal{A}}_j^{(k)}(P)$, $\nabla_i \tilde{\mathcal{A}}^{(k)}(P)$ and $\Omega_i$ for each robot within communication range<br>(iii) $S_j$ knows items #3 and #4 from Proposition V.1, and $\gamma^{(k)}$<br>(iv) Shrinkage factor $\tau$ and tolerance $\theta \in (0,1)$ known a priori by all static nodes |
| **Output:** | Step size $\tau \in \mathbb{R}$. |

Initialization

1: $S_1, \ldots, S_m$ calculate $\alpha_{\max}$, cf. (6) via a consensus algorithm

For $j \in \{1, \ldots, m\}$, node $S_j$ executes concurrently

1: $\alpha = \alpha_{\max}$
2: **repeat**
3: calculates $\mathrm{d}_j(\alpha, P)^2$
4: calculates $\hat{\varphi}^{(k)}\left(P_j'(\alpha, P)\right)$ according to Proposition V.2
5: calculates $\tilde{\mathcal{A}}_j^{(k)}\left(P_j'(\alpha, P)\right)$
6: execute consensus algorithm to calculate the following:

$$\tilde{\mathcal{A}}^{(k)}\left(P'(\alpha, P)\right) = \sum_{j=1}^{m} \tilde{\mathcal{A}}_j^{(k)}\left(P_j'(\alpha, P)\right)$$

$$\left\|P - P'(\alpha, P)\right\|^2 = \sum_{j=1}^{m} \mathrm{d}_j(\alpha, P)^2$$

7: $\nu = \frac{\theta}{\alpha}\left\|P - P'(\alpha, P)\right\|^2 + \tilde{\mathcal{A}}^{(k)}(P'(\alpha, P)) - \tilde{\mathcal{A}}^{(k)}(P)$
8: **if** $\nu > 0$ **then**
9: $\alpha = \alpha\tau$
10: **until** $\nu \leq 0$

TABLE I
DISTRIBUTED LINE SEARCH ALGORITHM.

gradient descent algorithm to define a sequence of configurations, $\{P_l^\dagger\}$, $l \in \mathbb{N}$, such that $P_1^\dagger$ is $P^{(k)} \in \mathcal{D}^n$, the vector of current spatial locations of the robotic agents and

$$P_{l+1}^\dagger = \mathrm{proj}_\Omega\left(P_l^\dagger - \alpha \nabla \tilde{\mathcal{A}}(P_l^\dagger)\right), \alpha \in \mathbb{R}_{\geq 0},$$

where $\alpha$ is chosen via DISTRIBUTED LINE SEARCH ALGORITHM. When $|\tilde{\mathcal{A}}^{(k)}(P_{l+1}^\dagger) - \tilde{\mathcal{A}}^{(k)}(P_l^\dagger)| = 0$, the algorithm terminates, and the nodes set $P^{(k+1)} = P_{l+1}^\dagger$. By the end of this calculation, each node knows the identity of robotic agents in its Voronoi cell at timestep $k+1$. Node $S_j$ transmits $p_i(k+1)$ to robot $R_i$, which then moves to the location between timesteps. The overall algorithm is in Table II.

**Proposition VI.1** *The* DISTRIBUTED PROJECTED GRADIENT DESCENT ALGORITHM *is distributed over the network $\mathcal{N}$. Moreover, under the assumptions of Lemma IV.4, any execution is such that the robots do not collide and, at each timestep after the first, measurements are taken at stationary configurations of $P \mapsto \tilde{\mathcal{A}}^{(k)}(P)$ over $\Omega^{(k)}$.*

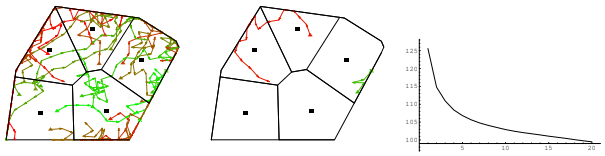
The proposed algorithm is robust to agent failures. If an agent stops sending position updates, it ceases to receive new control vectors. The rest of the network continues operating with the available resources and will eventually sample the areas previously covered by the failing agents.

### A. Simulations

We show here an implementation of the DISTRIBUTED PROJECTED GRADIENT DESCENT ALGORITHM with the following parameters: $m = 5$ static nodes, $n = 20$ robotic agents, and the convex polygon $\mathcal{D}$ with vertices $\{(0,.1), (2.5,.1), (3.45,1.6), (3.5,1.7), (3.45,1.8), (2.7,2.2), (1,2.4), (0.2,1.3)\}$. We used the separable covariance function defined by $\mathrm{Cov}[Z(s_1,t_1), Z(s_2,t_2)] = C_{\text{trunc}}(\|s_1 - s_2\|, 0.3) C_{\text{trunc}}(|t_1 - t_2|, 3.5)$, where

$$C_{\text{trunc}}(\delta, r) = \begin{cases} e^{-15\left(\frac{\delta}{r}\right)^2} & \text{if } \delta \leq r, \\ 0 & \text{otherwise.} \end{cases}$$

While the covariance function is not $C^1$ everywhere, the difference lies within the error margin of the simulation. We use $\omega = 0.02$ and $u_{\max} = 0.3$. The values of our hyperparameters were $\nu = 0.1$, $q = 2$, $\beta_0 = \mathbf{0}$, and $\mathbf{K}_0 = \boldsymbol{I}$. We simulated the sampled data by drawing random variables from the distribution $N(\beta_0, \sigma_0^2 \mathbf{K}_u)$, where $\sigma_0^2 = \frac{q\nu}{\nu - 2}$, the prior mean of $\sigma^2$, and $\mathbf{K}_u$ is the correlation matrix of $\underline{y}_u$. For the mean regression functions $f_i$, we used $f(x,y,t) = (1,x,y)^T$. To illustrate the robustness to failure, $R_2$ ceased communications after timestep 2, and $R_5$ after timestep 4. Figure 1 shows the trajectories taken by the robots. This



(a) (b) (c)

Fig. 1. (a) Trajectories of all robots, (b) two representative robot trajectories and (c) evolution of the objective function. The filled squares represent the (static) positions of the nodes, and the filled triangles show the starting positions of the robots. The X's represent the positions of the two robots who dropped communication.

example is representative of cases for which the data samples lie within a reasonable range of the predictive model. In the cases where the samples do not match the model, the surface of $\tilde{\mathcal{A}}^{(k)}$ is relatively flat, signifying that the amount of information to be gained is not significantly different whether the agents move or not. As information is a model-dependent quantity, this is not surprising. Furthermore, if the model is too far off, the approximation $\varphi \approx \hat{\varphi}$ is very bad.

## VII. CONCLUSIONS AND FUTURE WORK

We have considered a network of static computing nodes and mobile robotic sensing platforms taking measurements of a time-varying random process with covariance known up to a scaling parameter. We have used a Bayesian approach, treating the field as a spatiotemporal Gaussian random process, and developed a novel iterative approach to calculating the variance of the posterior predictive distribution. Using this sequential formulation, we have developed a projected gradient descent algorithm which is distributed over the network of nodes and robots. Future work will focus on theoretical guarantees on the accuracy of the approximation $\tilde{\mathcal{A}}^{(k)}$ and on the robustness to failure of the proposed

**Name:** DISTRIBUTED PROJECTED GRADIENT DESCENT ALGORITHM

**Goal:** Find a local minimum of $\tilde{\mathcal{A}}^{(k)}$ within $\Omega^{(k)}$.

**Assumes:** (i) Connected network of static computing nodes and mobile robotic sensing agents
(ii) Static nodes deployed over $\mathcal{D}$ such that $R \geq \max_{i\in\{1,\dots,m\}} \{\mathrm{CR}(V_i(Q))\} + r + u_{\max}$, robotic agents in initial configuration $P^{(1)} \in \mathcal{D}^n$
(iii) Line search shrinkage factor $\tau$ and tolerance value $\theta \in (0,1)$ known a priori by all nodes
(iv) A termination marker known to all nodes and robots which may be sent to mark the end of a gradient descent loop.

**Uses:** (i) Each node uses the temporary vectors $P_{\text{cur}}$, respectively $P_{\text{next}}$ to hold the configuration at the current, respectively next step of the gradient projection algorithm. For ease of exposition, we use global notation although $S_j$ only calculates and uses the parts of these vectors which correspond to agents currently within communication range.

At time $k \in \mathbb{Z}_{\geq 0}$, node $S_j$ executes:

sets $R_{\text{cov}}(j) = \{R_i \mid \mathrm{d}(p_i(k), V_j(Q)) \leq r\}$
collects initial samples and locations from $R_i$ for each $i \in R_{\text{cov}}(j)$.
computes first $\tilde{\mathcal{A}}_j^{(k)}\big(P^{(k)}\big)$ and then $\tilde{\mathcal{A}}^{(k)}\big(P^{(k)}\big)$ via consensus
sets $P_{\text{next}} = P^{(k)}$
**repeat**
  sets $P_{\text{cur}} = P_{\text{next}}(j)$ and calculates $-\nabla\tilde{\mathcal{A}}_j^{(k)}(P_{\text{cur}})$
  transmits vector $\nabla_i\tilde{\mathcal{A}}_j^{(k)}(P_{\text{cur}})$ to all robots in $R_{\text{cov}}(j)$
  collects sum $\nabla_i\tilde{\mathcal{A}}^{(k)}(P_{\text{cur}})$ from all robots in $R_{\text{cov}}(j)$
  runs DISTRIBUTED LINE SEARCH ALGORITHM at $P_{\text{cur}}$ to get $\alpha$
  sets $P_{\text{next}} = P_{\text{cur}} + \alpha\nabla\tilde{\mathcal{A}}^{(k)}(P_{\text{cur}})$
  calculates $|\tilde{\mathcal{A}}^{(k)}(P_{\text{next}}) - \tilde{\mathcal{A}}^{(k)}(P_{\text{cur}})|$ from known quantities
**until** $|\tilde{\mathcal{A}}^{(k)}(P_{\text{next}}) - \tilde{\mathcal{A}}^{(k)}(P_{\text{cur}})| = 0$
sets $P^{(k+1)} = P_{\text{next}}$ and sends next position to robots in $V_j(Q)$

At time $k \in \mathbb{Z}_{\geq 0}$, robot $R_i$ executes:

takes measurement at $p_i(k)$
sets $S_{\text{cov}}(i) = \{S_j \mid \mathrm{d}(p_i(k), V_j(Q)) \leq r\}$
sends measurement and position to all nodes in $S_{\text{cov}}(i)$
**repeat**
  receives $\nabla_i\tilde{\mathcal{A}}_j^{(k)}(P^{(k)})$ from nodes in $S_{\text{cov}}(i)$
  calculates sum $\nabla_i\tilde{\mathcal{A}}^{(k)}(P^{(k)})$
  sends $\nabla_i\tilde{\mathcal{A}}^{(k)}(P^{(k)})$ to all nodes in $S_{\text{cov}}(i)$
**until** receives termination marker from any node
receives next location $p_i(k+1)$
moves to $p_i(k+1)$.

TABLE II
DISTRIBUTED PROJECTED GRADIENT DESCENT ALGORITHM.

coordination algorithm, the quantification of the communication requirements of the proposed strategy, and possible methods of reducing those requirements.

## VIII. ACKNOWLEDGMENTS

The authors thank B. Sansó and H. Lee for several conversations on Bayesian spatial statistics. This research was partially supported by NSF CAREER Award ECS-0546871.

## REFERENCES

[1] P. K. Kitanidis, "Parameter uncertainty in estimation of spatial functions: Bayesian analysis," *Water resources research*, vol. 22, pp. 449–507, 1986.
[2] M. Gaudard, M. Karvson, E. Linder, and D. Sinha, "Bayesian spatial prediction," *Environmental and Ecological Statistics*, vol. 6, pp. 147–171, 1999.
[3] N. D. Lee and J. V. Zidek, *Statistical Analysis of Environmental Space-Time Processes*. Springer Series in Statistics, New York: Springer, 2006.
[4] K. Chaloner and I. Verdinelli, "Bayesian experimental design, a review," *Statistical Science*, vol. 10, no. 3, pp. 273–304, 1995.
[5] M. L. Stein, "The screening effect in kriging," *The Annals of Statistics*, vol. 30, pp. 298–323, Feb. 2002.
[6] R. Furrer, M. G. Genton, and D. Nychka, "Covariance tapering for interpolation of large spatial datasets," *Journal of Computational and Graphical Statistics*, vol. 15, no. 3, pp. 502–523, 2006.
[7] F. Pukelsheim, *Optimal Design of Experiments*, vol. 50 of *Classics in Applied Mathematics*. Philadelphia, PA: SIAM, 2006.
[8] E. P. Liski, N. K. Mandal, K. R. Shah, and B. K. Sinha, *Topics in Optimal Design*, vol. 163 of *Lecture Notes in Statistics*. New York: Springer, 2002.
[9] J. S. Willcox, J. G. Bellingham, Y. Zhang, and A. B. Baggeroer, "Performance metrics for oceanographic surveys with autonomous underwater vehicles," *IEEE Journal of Oceanic Engineering*, vol. 26, no. 4, pp. 711–725, 2001.
[10] N. E. Leonard, D. Paley, F. Lekien, R. Sepulchre, D. M. Fratantoni, and R. Davis, "Collective motion, sensor networks, and ocean sampling," *Proceedings of the IEEE*, vol. 45, pp. 48–74, Jan. 2007. Special Issue on Networked Control Systems.
[11] K. M. Lynch, I. B. Schwartz, P. Yang, and R. A. Freeman, "Decentralized environmental modeling by mobile sensor networks," *IEEE Transactions on Robotics*, vol. 24, no. 3, pp. 710–724, 2008.
[12] S. Martínez, "Distributed representation of spatial fields through an adaptive interpolation scheme," in *American Control Conference*, (New York City, NY), pp. 2750–2755, 2007.
[13] M. A. Demetriou and I. I. Hussein, "Estimation of spatially distributed processes using mobile spatially distributed sensor network," *SIAM Journal on Control and Optimization*, vol. 48, no. 1, pp. 266–291, 2009.
[14] R. Graham and J. Cortés, "A cooperative deployment strategy for optimal sampling in spatiotemporal estimation," in *IEEE Conf. on Decision and Control*, (Cancun, Mexico), pp. 2432–2437, Dec. 2008.
[15] T. Leonard and J. S. J. Hsu, *Bayesian Methods*, vol. 1 of *Cambridge series in statistical and probabilistic mathematics*. Cambridge, UK: Cambridge University Press, 1999.
[16] C. P. Robert and G. Casella, *Monte Carlo statistical methods*. Springer texts in statistics, New York: Springer, 2004.
[17] R. Olfati-Saber and R. M. Murray, "Consensus problems in networks of agents with switching topology and time-delays," *IEEE Transactions on Automatic Control*, vol. 49, no. 9, pp. 1520–1533, 2004.
[18] D. P. Bertsekas and J. N. Tsitsiklis, *Parallel and Distributed Computation: Numerical Methods*. Athena Scientific, 1997.
[19] D. P. Bertsekas, "On the Goldstein-Levitin-Polyak gradient projection method," *IEEE Transactions on Automatic Control*, vol. 21, no. 2, pp. 174–184, 1976.