Distributed gradient ascent of random fields by robotic sensor networks

Jorge Cortés

***Abstract*— This paper considers a robotic sensor network, deployed in an environment of interest, that takes successive measurements of a spatial random field. Taking a Bayesian perspective on the kriging interpolation technique from geostatistics, we design the DISTRIBUTED KRIGING ALGORITHM to estimate the distribution of the random field and of its gradient. The proposed algorithm makes use of a novel distributed strategy to compute weighted least squares estimates when measurements are spatially correlated. This strategy results from the combination of the Jacobi overrelaxation method with dynamic consensus algorithms. The network agents use the information gained on the spatial field to implement a gradient ascent coordination algorithm, whose convergence is analyzed via stochastic Lyapunov functions in the absence of measurement errors. We illustrate our results in simulation.**

I. INTRODUCTION

Problem statement: Consider a robotic sensor network taking successive measurements of a physical process modeled as a spatial random field. Our objective is to design a distributed estimation and motion coordination algorithm that enables the network to find maxima of the spatial field. This type of tasks are relevant in multiple scenarios, including environmental monitoring, oceanographic exploration, and atmospheric research, when one might be interested in finding high pollutant concentrations, areas of maximum salinity, or locations where algae are abundant. Similar ideas can be used to localize areas of rapid variability of physical processes.

Literature review: In cooperative control, [1] designs network coordination strategies to seek out local optima of a deterministic, static field using uncorrelated noisy measurements and all-to-all communication. The works [2], [3] develop distributed optimal estimation strategies for networks with connected communication topology. Objective analysis is employed in [4] to find optimal network trajectories in restricted parameterized families of curves. Parallel algorithms for static networks are thoroughly studied in [5]. The works [6], [7] introduce distributed fusion algorithms based on averaging. Dynamic consensus algorithms that track the average of a given time-varying signal are studied in [8], [9]. In geostatistics, kriging [10], [11] is a standard technique to produce estimates of spatial processes based on data collected at finitely many locations. An advantage of kriging over other interpolation methods is that it provides a measure of the uncertainty of the predictor. A source of inspiration for our technical approach is [12], which develops an inferential framework for directional gradients of spatial fields based on point-referenced data. The convergence properties of our gradient ascent strategy are analyzed via stochastic stability analysis, in particular, the supermartingale convergence theorem [13] and stochastic Lyapunov functions [14].

Statement of contributions: The contributions of this paper are: (i) the formulation of the spatial field estimation problem via Bayesian universal kriging, and the incorporation of statistically sound gradient derivatives of the spatial field; (ii) the synthesis of a distributed algorithm to compute weighted least squares estimates when sensor measurements are correlated. This algorithm combines the Jacobi overrelaxation method with dynamic average consensus algorithms; (iii) the design of a distributed algorithm to estimate the distribution of the spatial field and of its gradient; and (iv) building on the previous contributions, the synthesis of a distributed motion coordination strategy that makes individual robotic agents converge with probability one to the set of critical points of the random spatial field. For reasons of space, proofs are omitted. The interested reader may find them in [15].

Organization: Section II presents reviews kriging interpolation. Section III introduces the models for the physical process and the robotic sensor network. Section IV describes the sequential estimation of the spatial field and of its gradient. Section V presents a distributed algorithm to compute weighted least squares estimates when sensor measurements are correlated. This algorithm is used in Section VI to design a distributed implementation of the estimation discussed in Section IV. Section VII analyzes the coordination algorithm executed by the network to localize critical points of the spatial field. Section VIII presents our conclusions.

Notation: Vectors in Euclidean space are understood as column vectors. Let $e_1, \dots, e_d$ denote the canonical basis of $\mathbb{R}^d$. Given a matrix $A \in \mathbb{R}^{n \times m}$, $\text{row}_i(A) \in \mathbb{R}^m$ and $\text{col}_j(A) \in \mathbb{R}^n$ denote the ith row and the jth column of A, respectively. For an undirected graph $G = (V, E)$ consisting of a set of vertices V and a set of edges $E \subset V \times V$, the neighbors of $v \in V$ in G are denoted by $\mathcal{N}_G(v) = \{w \in V \mid (v, w) \in E\}$. Usually, $V = \{1, \dots, n\}$. The adjacency matrix of G is $\mathcal{A}(G) = (a_{ij}) \in \mathbb{R}^{n \times n}$ defined by $a_{ij} = 1$ if $(i, j) \in E$, and $a_{ij} = 0$ otherwise. We often denote it by $\mathcal{A}$.

II. RANDOM SPATIAL FIELDS

We review important notions on random spatial fields and kriging. The reader is referred to [10], [11] for more details. Let Z be a scalar random spatial field on $\mathbb{R}^d$, $d \in \mathbb{Z}_{>0}$, with positive definite covariance $C : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}_{\geq 0}$, $\text{Cov}(Z(s), Z(s')) = C(s, s')$, $s, s' \in \mathbb{R}^d$. The field Z is *stationary* if $C(s, s') = K(s - s')$, for $K : \mathbb{R}^d \to \mathbb{R}_{\geq 0}$, and *isotropic* if $C(s, s') = \widetilde{K}(\|s - s'\|)$, for $\widetilde{K} : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$.

A. Universal kriging

Let us briefly review the spatial estimation procedure called *universal kriging*. Consider a stationary spatial field Z with mean function of the form $\mu(s) = x(s)^T \beta =$

Research partially supported by NSF CAREER Award ECS-0546871.

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

$\sum_{i=1}^p \beta_i x_i(s)$. Here, the components of $x : \mathbb{R}^d \to \mathbb{R}^p$ are known, and $\beta = (\beta_1, \ldots, \beta_p) \in \mathbb{R}^p$ is an unknown parameter. Given $n \in \mathbb{Z}_{>0}$, let $X : (\mathbb{R}^d)^n \to \mathbb{R}^{n\times p}$ map $(p_1, \ldots, p_n)$ to the matrix $X(p_1, \ldots, p_n)$ whose ith row is $x(p_i)^T$. Given measurements $\boldsymbol{Z} = (Z(p_1), \ldots, Z(p_n))$ of Z at locations $p_1, \ldots, p_n$, the *universal kriging predictor at* $s \in \mathbb{R}^d$ minimizes the mean-squared prediction error $\sigma(s; \boldsymbol{Z}) = E(Z(s) - p(s; \boldsymbol{Z}))^2$ among all linear unbiased predictors $p(s; \boldsymbol{Z}) = \sum_{i=1}^n l_i Z(p_i)$, $l = (l_1, \ldots, l_n) \in \mathbb{R}^n$. The explicit expression of the universal kriging predictor is

$$\widehat{p}_{\text{UK}}(s; \boldsymbol{Z}) = \tag{1a}$$
$$(\boldsymbol{\gamma}(s) + \boldsymbol{X}(\boldsymbol{X}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{X})^{-1}(x(s) - \boldsymbol{X}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\gamma}(s)))^T\boldsymbol{\Sigma}^{-1}\boldsymbol{Z},$$

where we have used the shorthand notation

$$\begin{aligned}\boldsymbol{X} &= X(p_1, \ldots, p_n) \in \mathbb{R}^{n\times p},\\ \boldsymbol{\Sigma} &= \big(K(p_i - p_j)\big)_{i,j=1}^n \in \mathbb{R}^{n\times n},\\ \boldsymbol{\gamma}(s)^T &= (K(s - p_1), \ldots, K(s - p_n)) \in \mathbb{R}^n.\end{aligned}$$

The mean-squared prediction error of $\widehat{p}_{\text{UK}}(s; \boldsymbol{Z})$ at $s \in \mathbb{R}^d$ is

$$\sigma_{\text{UK}}(s; \boldsymbol{Z}) = K(0) - \boldsymbol{\gamma}(s)^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\gamma}(s) + (x(s) \tag{1b}$$
$$- \boldsymbol{X}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\gamma}(s))^T(\boldsymbol{X}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{X})^{-1}(x(s) - \boldsymbol{X}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\gamma}(s)).$$

1) Universal kriging as a two-step procedure: Universal kriging can alternatively be described as a two-step procedure, whereby one (i) estimates the unknown β from the data, and then (ii) performs *simple kriging*. In (i), the best linear unbiased estimator of β is the weighted least-squares

$$\widehat{\beta}_{\text{LS}} = (\boldsymbol{X}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{Z}, \tag{2}$$

with covariance matrix $\operatorname{cov}(\widehat{\beta}_{\text{LS}}) = (\boldsymbol{X}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{X})^{-1}$. In (ii), simple kriging assumes μ (i.e., β) known, and performs spatial interpolation to yield, see e.g., [10], [11],

$$\widehat{p}_{\text{SK}}(s; \boldsymbol{Z}) = x(s)^T\beta + \boldsymbol{\gamma}(s)^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{Z} - \boldsymbol{X}\beta), \tag{3a}$$
$$\sigma_{\text{SK}}(s; \boldsymbol{Z}) = K(0) - \boldsymbol{\gamma}(s)^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\gamma}(s). \tag{3b}$$

The predictor (3a) with $\beta = \widehat{\beta}_{\text{LS}}$ results in the universal kriging predictor (1a). The universal kriging variance (1b) is then the sum of the simple kriging variance (3b) and the spatial error induced by parameter estimation uncertainty.

B. Gradient random spatial fields

The discussion here follows [12]. Given a stationary random field Z on $\mathbb{R}^d$ and a vector $u \in \mathbb{R}^d$, a *directional gradient* field on $\mathbb{R}^d$ is defined as

$$D_u Z(s) = \lim_{h\to 0} \frac{Z(s + hu) - Z(s)}{h}, \quad s \in \mathbb{R}^d,$$

where the limit is understood in the L_2 sense (i.e., $\lim_{h\to 0} E(\frac{Z(s+hu)-Z(s)}{h} - D_u Z(s))^2 = 0$). The random field Z is *mean square differentiable at* $s_0 \in \mathbb{R}^d$ if there exists a vector $\nabla Z(s_0) \in \mathbb{R}^d$ such that, for all $u \in \mathbb{R}^d$,

$$\lim_{h\to 0} E\Big(\frac{Z(s_0 + hu) - Z(s_0)}{h} - \nabla Z(s_0)^T u\Big)^2 = 0.$$

It follows that $D_u Z(s_0) = \nabla Z(s_0)^T u$, for all $u \in \mathbb{R}^d$ (where the equality is understood in the L_2-sense). In particular,

$$\nabla Z(s_0) = (D_{e_1} Z(s_0), \ldots, D_{e_n} Z(s_0)).$$

Throughout the paper, we deal with random fields that are mean square differentiable everywhere.

If Z is a stationary Gaussian random field, the resulting joint (d+1-dimensional multivariate) Gaussian field $(Z, \nabla Z)$ on $\mathbb{R}^d$ has a valid cross-covariance function

$$\operatorname{Cov}((Z(s), \nabla Z(s)), (Z(s'), \nabla Z(s'))) = \begin{pmatrix} K(s - s') & -(\nabla K(s - s'))^T \\ \nabla K(s - s') & -\operatorname{H}(K)(s - s') \end{pmatrix}, \tag{4}$$

where $\operatorname{H}(K)(s)$ denotes the Hessian matrix of K at s. This joint distribution allows predictive inference for the gradient at arbitrary points given measurements of Z at arbitrary locations. Given measurements $Z(p_1), \ldots, Z(p_n)$ and $s \in \mathbb{R}^d$, define the following shorthand notation,

$$\begin{aligned}\boldsymbol{\mu} &= (\mu(p_1), \ldots, \mu(p_n)) \in \mathbb{R}^n,\\ \boldsymbol{\nabla\gamma}(s)^T &= (\nabla K(s - p_1), \ldots, \nabla K(s - p_n)) \in \mathbb{R}^{d\times n}.\end{aligned}$$

According to (4), $(\mathbf{Z}, \nabla Z(s))$ is distributed as the $(d + n)$-dimensional normal distribution

$$N_{d+n}\left(\begin{pmatrix}\boldsymbol{\mu}\\ \nabla\mu(s)\end{pmatrix}, \begin{pmatrix}\boldsymbol{\Sigma} & \boldsymbol{\nabla\gamma}(s)\\ \boldsymbol{\nabla\gamma}(s)^T & -\operatorname{H}(K)(0)\end{pmatrix}\right).$$

From here, the predictive distribution of ∇Z conditional on the data $\boldsymbol{Z}$ is the d-dimensional normal distribution

$$\nabla Z(s) \mid \boldsymbol{Z} \sim N_d\left(\nabla\mu(s) + \boldsymbol{\nabla\gamma}(s)^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{Z} - \boldsymbol{\mu}),\right.$$
$$\left. -\operatorname{H}(K)(0) - \boldsymbol{\nabla\gamma}(s)^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\nabla\gamma}(s)\right), \tag{5}$$

where $\sim$ means "distributed according to." A *critical point* of Z is a point $s_* \in \mathbb{R}^d$ such that $\nabla Z(s_*) = 0$. A critical point satisfies $D_u Z(s_*) = 0$ for all $u \in \mathbb{R}^d$, and hence corresponds to a maximum, a minimum, or a saddle point of Z.

III. Problem statement

Our objective is to design a distributed estimation and motion coordination algorithm that enables a robotic sensor network to find the maxima of a physical process of interest.

A. Physical process model

We restrict our attention to processes that do not evolve in time. However, agents can take measurements at different times, and therefore we need to provide a model for the temporal correlation between them. We do this as follows: Z is a spatio-temporal Gaussian random field on $\mathbb{R}^d \times \mathbb{R}_{\geq 0}$

$$Z(s, t) = \mu(s) + \delta(s, t), \tag{6}$$

where $\mu(s) = x(s)^T\beta$ is continuously differentiable. Here, δ is a zero-mean Gaussian random field with the same (separable) covariance as Z, which is given by

$$\operatorname{Cov}(Z(s, t), Z(s', t')) = \begin{cases} K(s - s') & \text{if } |t - t'| < 1,\\ 0 & \text{otherwise,}\end{cases} \tag{7}$$

where $K : \mathbb{R}^d \to \mathbb{R}_{\geq 0}$. We assume that the field has a finite spatial correlation, i.e., there exists $r \in \mathbb{R}_{>0}$ such that

$$K(s - s') = 0 \quad \text{for} \quad \|s - s'\| > r. \tag{8}$$

B. Network model

Consider a network of n agents evolving in $\mathbb{R}^d$ according to $\dot{p}_i = u_i$, $i \in \{1, \dots, n\}$. The control is bounded $\|u_i\| \leq u_{\max} \in \mathbb{R}_{>0}$. Agents are equipped with identical sensors, and take point measurements at times $k \in \mathbb{Z}_{\geq 0}$. The measurement taken by agent i located at p_i at time k is corrupted by white noise

$$Y_i(k) = Z(p_i, k) + \epsilon_i, \tag{9}$$

where $\epsilon_i \sim N(0, \sigma)$. Measurement errors are assumed independent. Each agent can communicate with other agents located within a distance $R \in \mathbb{R}_{>0}$. As we show later, each agent can construct a distributed representation of the spatial field and of its gradient in a ball centered at its location with radius $R - r$ (note that, if $\|p_i - s\| > R - r$, then Z at s is correlated with points that fall outside the communication ball of agent i). Therefore, we make the assumption

$$u_{\max} \leq R - r.$$

The communication capabilities of the agents induce the network topology corresponding to the R-disk graph $\mathcal{G}_{R\text{-disk}}$. At each configuration $(p_1, \dots, p_n) \in (\mathbb{R}^d)^n$, $\mathcal{G}_{R\text{-disk}}(p_1, \dots, p_n)$ is an undirected graph with vertex set $\{p_1, \dots, p_n\}$ and edge set $\{(p_i, p_j) \mid \|p_i - p_j\| \leq R\}$. This graph is a particular example of a proximity graph, see e.g., [16]. We assume that either the number of agents n is a priori known to everybody, or that agents run a consensus algorithm to determine it.

Remark 3.1 (Distributed computation): One can find formal definitions of the notion of distributed computation of functions, see e.g., [16]. Here, we refer to a computation as distributed over a graph if each node can perform the computation using information provided by its neighbors. •

IV. Sequential spatial estimation

In this section, we take a Bayesian perspective to incorporate previous knowledge into the estimation of the spatial field and its gradient. Our setup is different from Section II-A in that errors are present in the measurements according to (9), and in that prior information on the unknown parameter is incorporated. We consider the spatial field estimation when measurements are taken at multiple time instants, or *sequentially*. We follow the next scheme: (i) Section IV-A computes the posterior distribution of the parameter given the data, (ii) Section IV-B computes the conditional distribution of the spatial field and its gradient given the data and the parameter, and (iii), Section IV-C merges (i) and (ii).

Before proceeding, let us introduce some useful notation. Let $\boldsymbol{Y}(t) = (Y_1(t), \dots, Y_n(t)) \in \mathbb{R}^n$ denote the measurements taken by the network agents at time $t \in \mathbb{Z}_{\geq 0}$. Given $k \in \mathbb{Z}_{\geq 0}$, let $\boldsymbol{Y}_{\leq k} = (\boldsymbol{Y}(0), \dots, \boldsymbol{Y}(k))$ denote the data available up to time k. Since measurements are taken by the network only at time instants in $\mathbb{Z}_{\geq 0}$, for $t \in \mathbb{R}_{\geq 0}$, one has $\boldsymbol{Y}_{\leq t} = \boldsymbol{Y}_{\leq \lfloor t \rfloor}$. For $k \in \{0, \dots, \lfloor t \rfloor\}$, let

$$\begin{aligned}
\boldsymbol{\Sigma}(k) &= \big(K(p_i(k) - p_j(k))\big) \in \mathbb{R}^{n \times n},\\
\boldsymbol{X}(k) &= X(p_1(k), \dots, p_n(k)) \in \mathbb{R}^{n \times p},\\
\boldsymbol{\gamma}(s,k)^T &= (K(s - p_1(k)), \dots, K(s - p_n(k))) \in \mathbb{R}^n,\\
\boldsymbol{\nabla\gamma}(s,k)^T &= (\nabla K(s - p_1(k)), \dots, \nabla K(s - p_n(k))) \in \mathbb{R}^{d \times n}.
\end{aligned}$$

The stucture of the spatial field covariance (7) has some important consequences. First, the ith components of $\boldsymbol{\gamma}(s,k)$ and $\boldsymbol{\nabla\gamma}(s,k)$ can only be nonvanishing if $\|s - p_i(k)\| \leq r$. Second, the covariance matrix of the data $\boldsymbol{Y}_{\leq t}$ is the block-diagonal matrix $\boldsymbol{\Sigma}_{\leq t} + \sigma I_{n(\lfloor t \rfloor + 1)}$, where

$$\boldsymbol{\Sigma}_{\leq t} = \begin{pmatrix} \boldsymbol{\Sigma}(0) & 0_{n\times n} & \dots & 0_{n\times n} \\ 0_{n\times n} & \boldsymbol{\Sigma}(1) & & \vdots \\ \vdots & & \ddots & 0_{n\times n} \\ 0_{n\times n} & \dots & 0_{n\times n} & \boldsymbol{\Sigma}(\lfloor t \rfloor) \end{pmatrix}. \tag{10a}$$

Let us also define the shorthand notation

$$\boldsymbol{X}_{\leq t}^T = \big(\boldsymbol{X}(0), \dots, \boldsymbol{X}(\lfloor t \rfloor)\big)^T, \tag{10b}$$
$$\boldsymbol{\gamma}_{\leq t}(s)^T = \big(0, \dots, 0, \boldsymbol{\gamma}(s, \lfloor t \rfloor)^T\big), \tag{10c}$$
$$\boldsymbol{\nabla\gamma}_{\leq t}(s)^T = \big(0, \dots, 0, \boldsymbol{\nabla\gamma}(s, \lfloor t \rfloor)^T\big). \tag{10d}$$

A. Sequential parameter estimation

Assume that β is distributed according to a multivariate normal distribution $\beta \sim N_p(\nu, V)$. From equations (6) and (9), the posterior distribution of β at time $t \in \mathbb{R}_{\geq 0}$ is

$$\begin{aligned}
&\beta \,|\, \boldsymbol{Y}_{\leq t} \sim N_p(\omega_{\leq t}, W_{\leq t}),\\
&\quad \omega_{\leq t} = W_{\leq t}\big(\boldsymbol{X}_{\leq t}^T(\boldsymbol{\Sigma}_{\leq t} + \sigma I_{n(\lfloor t \rfloor + 1)})^{-1}\boldsymbol{Y}_{\leq t} + V^{-1}\nu\big),\\
&\quad W_{\leq t} = (\boldsymbol{X}_{\leq t}^T(\boldsymbol{\Sigma}_{\leq t} + \sigma I_{n(\lfloor t \rfloor + 1)})^{-1}\boldsymbol{X}_{\leq t} + V^{-1})^{-1}.
\end{aligned}$$

Alternatively, the mean $\omega_{\leq t}$ and covariance $W_{\leq t}$ can be expressed as follows.

Lemma 4.1 (Sequential parameter estimation): Assume that initially $\beta \sim N_p(\nu, V)$. For all $t \in \mathbb{R}_{\geq 0}$, the mean and the covariance matrix of the posterior distribution of β with data collected up to time t can be written as

$$\begin{aligned}
\omega_{\leq t} &= W_{\leq t}\big(\boldsymbol{X}(\lfloor t \rfloor)^T\boldsymbol{\Sigma}(\lfloor t \rfloor)_\sigma^{-1}\boldsymbol{Y}(\lfloor t \rfloor) + W_{\leq t-1}^{-1}\omega_{\leq t-1}\big),\\
W_{\leq t} &= \big(\boldsymbol{X}(\lfloor t \rfloor)^T\boldsymbol{\Sigma}(\lfloor t \rfloor)_\sigma^{-1}\boldsymbol{X}(\lfloor t \rfloor) + W_{\leq t-1}^{-1}\big)^{-1},
\end{aligned}$$

where $\omega_{\leq -1} = \nu$, $W_{\leq -1} = V$, and $\boldsymbol{\Sigma}(k)_\sigma = \boldsymbol{\Sigma}(k) + \sigma I_n$.

Lemma 4.1 provides an iterative fashion of computing $\omega_{\leq t}$ and $W_{\leq t}$ that is appropriate for a distributed implementation by the robotic network. We describe this in Section VI-A.

B. Sequential simple kriging

The conditional distribution $Z(s,t) \,|\, (\boldsymbol{Y}_{\leq t}, \beta)$ is

$$\begin{aligned}
N\big(&x(s)^T\beta + \boldsymbol{\gamma}_{\leq t}(s)^T(\boldsymbol{\Sigma}_{\leq t} + \sigma I_{n(\lfloor t \rfloor + 1)})^{-1}(\boldsymbol{Y}_{\leq t} - \boldsymbol{X}_{\leq t}\beta),\\
&K(0) - \boldsymbol{\gamma}_{\leq t}(s)^T(\boldsymbol{\Sigma}_{\leq t} + \sigma I_{n(\lfloor t \rfloor + 1)})^{-1}\boldsymbol{\gamma}_{\leq t}(s)\big).
\end{aligned}$$

In the absence of measurement errors, this corresponds to the simple kriging predictor and variance. According to (5), the conditional distribution $\nabla Z(s,t) \,|\, (\boldsymbol{Y}_{\leq t}, \beta)$ is

$$\begin{aligned}
N_d\big(&\nabla x(s)^T\beta + \boldsymbol{\nabla\gamma}_{\leq t}(s)^T(\boldsymbol{\Sigma}_{\leq t} + \sigma I_{n(\lfloor t \rfloor + 1)})^{-1}(\boldsymbol{Y}_{\leq t} - \boldsymbol{X}_{\leq t}\beta),\\
&-\mathrm{H}(K)(0) - \boldsymbol{\nabla\gamma}_{\leq t}(s)^T(\boldsymbol{\Sigma}_{\leq t} + \sigma I_{n(\lfloor t \rfloor + 1)})^{-1}\boldsymbol{\nabla\gamma}_{\leq t}(s)\big).
\end{aligned}$$

Using (10) above, one can readily see that only the information collected at time $\lfloor t \rfloor$ is used to construct the simple kriging estimator up to time t, as the following result states.

Lemma 4.2 (Sequential simple kriging): For all $t \in \mathbb{R}_{\geq 0}$ and all $s \in \mathbb{R}^d$, one has, the normal distribution $Z(s,t) \,|\, (\boldsymbol{Y}_{\leq t}, \beta)$ with mean

$$x(s)^T\beta + \boldsymbol{\gamma}(s, \lfloor t \rfloor)^T \boldsymbol{\Sigma}(\lfloor t \rfloor)_\sigma^{-1}(\boldsymbol{Y}(\lfloor t \rfloor) - \boldsymbol{X}(\lfloor t \rfloor)\beta),$$

and variance $K(0) - \boldsymbol{\gamma}(s, \lfloor t \rfloor)^T \boldsymbol{\Sigma}(\lfloor t \rfloor)_\sigma^{-1} \boldsymbol{\gamma}(s, \lfloor t \rfloor)$; and the normal distribution $\nabla Z(s,t) \,|\, (\boldsymbol{Y}_{\leq t}, \beta)$ with mean

$$\nabla x(s)^T\beta + \boldsymbol{\nabla\gamma}(s, \lfloor t \rfloor)^T \boldsymbol{\Sigma}(\lfloor t \rfloor)_\sigma^{-1}(\boldsymbol{Y}(\lfloor t \rfloor) - \boldsymbol{X}(\lfloor t \rfloor)\beta),$$

and variance $-\mathrm{H}(K)(0) - \boldsymbol{\nabla\gamma}(s, \lfloor t \rfloor)^T \boldsymbol{\Sigma}(\lfloor t \rfloor)_\sigma^{-1} \boldsymbol{\nabla\gamma}(s, \lfloor t \rfloor)$, where, for brevity, we let $\boldsymbol{\Sigma}(k)_\sigma = \boldsymbol{\Sigma}(k) + \sigma I_n$.

C. Sequential Bayesian universal kriging

We construct the Bayesian universal kriging predictor of the spatial field and its gradient by merging Sections IV-A and IV-B. Specifically, for $s \in \mathbb{R}^d$, the posterior predictive distribution of $Z(s,t)$ and $\nabla Z(s,t)$ given $\boldsymbol{Y}_{\leq t}$ is obtained by marginalizing the estimates in Lemma 4.2 over the posterior distribution $\beta \,|\, \boldsymbol{Y}_{\leq t}$ in Lemma 4.1. Accordingly, we obtain the normal distribution $Z(s,t) \,|\, \boldsymbol{Y}_{\leq t}$ with mean

$$x(s)^T\omega_{\leq t} + \boldsymbol{\gamma}(s, \lfloor t \rfloor)^T \boldsymbol{\Sigma}(\lfloor t \rfloor)_\sigma^{-1}(\boldsymbol{Y}(\lfloor t \rfloor) - \boldsymbol{X}(\lfloor t \rfloor)\omega_{\leq t})$$

and variance

$$\begin{aligned} K(0) &- \boldsymbol{\gamma}(s, \lfloor t \rfloor)^T \boldsymbol{\Sigma}(\lfloor t \rfloor)_\sigma^{-1} \boldsymbol{\gamma}(s, \lfloor t \rfloor) \\ &+ \big(x(s) - \boldsymbol{X}(\lfloor t \rfloor)^T \boldsymbol{\Sigma}(\lfloor t \rfloor)_\sigma^{-1} \boldsymbol{\gamma}(s, \lfloor t \rfloor)\big)^T W_{\leq t} \\ &\quad \big(x(s) - \boldsymbol{X}(\lfloor t \rfloor)^T \boldsymbol{\Sigma}(\lfloor t \rfloor)_\sigma^{-1} \boldsymbol{\gamma}(s, \lfloor t \rfloor)\big); \end{aligned}$$

and the normal distribution $\nabla Z(s,t) \,|\, \boldsymbol{Y}_{\leq t}$ with mean

$$\nabla x(s)^T\omega_{\leq t} + \boldsymbol{\nabla\gamma}(s, \lfloor t \rfloor)^T \boldsymbol{\Sigma}(\lfloor t \rfloor)_\sigma^{-1}(\boldsymbol{Y}(\lfloor t \rfloor) - \boldsymbol{X}(\lfloor t \rfloor)\omega_{\leq t}),$$

and variance

$$\begin{aligned} -\mathrm{H}(K)(0) &- \boldsymbol{\nabla\gamma}(s, \lfloor t \rfloor)^T \boldsymbol{\Sigma}(\lfloor t \rfloor)_\sigma^{-1} \boldsymbol{\nabla\gamma}(s, \lfloor t \rfloor) \\ &\big(\nabla x(s) - \boldsymbol{X}(\lfloor t \rfloor)^T \boldsymbol{\Sigma}(\lfloor t \rfloor)_\sigma^{-1} \boldsymbol{\nabla\gamma}(s, \lfloor t \rfloor)\big)^T W_{\leq \lfloor t \rfloor} \\ &\quad \big(\nabla x(s) - \boldsymbol{X}(\lfloor t \rfloor)^T \boldsymbol{\Sigma}(\lfloor t \rfloor)_\sigma^{-1} \boldsymbol{\nabla\gamma}(s, \lfloor t \rfloor)\big). \end{aligned}$$

Our next objective is to design a distributed coordination algorithm so that network agents can compute these quantities.

V. Distributed average weighted least squares

Given a network of n agents with interaction topology described by an undirected graph G, $B \in \mathbb{R}^{n\times n}$ invertible, $c \in \mathbb{R}^n$, and $M \in \mathbb{R}^{n\times p}$, we introduce here an algorithm distributed over G to compute

$$\frac{1}{n} M^T B^{-1} c. \tag{15}$$

As will be clear in Section VI, this quantity can be given the interpretation of an average weighted least squares estimate. This is why we refer to our procedure as the Weighted Least Squares Algorithm. The capability to compute such estimates is instrumental later to synthesize a distributed implementation of the estimation procedure of Section IV. To compute (15) in a distributed way, the idea is to combine a Jacobi iteration and a dynamic consensus algorithm into a single procedure. Let us first explain these ingredients.

A. Jacobi overrelaxation algorithm

Given an invertible matrix $B \in \mathbb{R}^{n\times n}$ and a vector $c \in \mathbb{R}^n$, consider the linear system $By = c$. The Jacobi overrelaxation (JOR) algorithm [5] is an iterative procedure to compute the unique solution $y = B^{-1}c \in \mathbb{R}^n$. It is formulated as the discrete-time dynamical system

$$y_i(\ell+1) = (1-h)y_i(\ell) - h\frac{1}{b_{ii}}\Big(\sum_{j\neq i} b_{ij}y_j(\ell) - c_i\Big),$$

for $\ell \in \mathbb{Z}_{\geq 0}$ and $i \in \{1, \dots, n\}$, with $y(0) \in \mathbb{R}^n$ and $h \in (0,1)$. The convergence properties of the JOR algorithm can be fully characterized [5] in terms of the eigenvalues of the matrix describing the iteration. Here we use the following sufficient convergence criteria from [17, Theorem 2].

Lemma 5.1: For $B \in \mathbb{R}^{n\times n}$ symmetric, positive definite and any $c \in \mathbb{R}^n$, if $h < 2/n$, the JOR algorithm converges to the solution of $By = c$ starting from any initial condition.

As long as (i) agent i has access to c_i, and (ii) if $b_{ij} \neq 0$, then i, j are neighbors in G, the JOR algorithm is amenable to distributed implementation in the following sense: agent i can compute the ith component y_i of the solution $y = B^{-1}c$ with the information provided by its neighbors in G.

B. Dynamic average consensus algorithms

Dynamic average consensus filters [8], [9] are distributed algorithms that allow the network to track the average of a given time-varying signal. Here, we use a particular instance of the proportional-integral dynamic consensus estimators studied in [9] but formulated for higher-dimensional signals. The algorithm works for time-dependent graphs, but here we restrict our attention to a fixed graph.

Let $\tau \mapsto u(\tau) \in (\mathbb{R}^m)^n$ be a time-varying function, that we refer to as *signal*. Note that $u(\tau)$ is a n-dimensional vector with each component $u_i(\tau)$, $i \in \{1, \dots, n\}$, being itself a m-dimensional vector. Consider the dynamical system

$$\begin{aligned} \frac{dv_i}{d\tau} &= \gamma(u_i(\tau) - v_i(\tau)) - \sum_{j\neq i} a_{ij}(v_i(\tau) - v_j(\tau)) \\ &\quad + \sum_{j\neq i} a_{ij}(w_i(\tau) - w_j(\tau)), \end{aligned} \tag{16a}$$

$$\frac{dw_i}{d\tau} = -\sum_{j\neq i} a_{ij}(v_i(\tau) - v_j(\tau)), \tag{16b}$$

for $i \in \{1, \dots, n\}$, where $\gamma > 0$ and $v, w \in (\mathbb{R}^m)^n$. Here, $\mathcal{A} = (a_{ij}) \in \mathbb{R}^{n\times n}$ is the adjacency matrix of G. If agent i has access to the ith-component u_i of the signal u, then this algorithm is distributed over G, i.e., agent i can compute the evolution of v_i and w_i with information provided by its G-neighbors. It can be proved [9] that for G connected, for any $\gamma > 0$, any constant input $\tau \mapsto u(\tau) = u \in (\mathbb{R}^m)^n$, and any initial $v(0), w(0) \in (\mathbb{R}^m)^n$, the algorithm (16) satisfies

$$v_i(\tau) - \frac{1}{n}\sum_{i=1}^n u_i(\tau) \to 0 \quad \text{as } \tau \to +\infty \tag{17}$$

exponentially fast for all $i \in \{1, \dots, n\}$. For slowly-varying signals, the estimator guarantees small steady-state errors.

C. The WEIGHTED LEAST SQUARES ALGORITHM

Here, we combine the JOR algorithm and the dynamic consensus algorithm to synthesize the algorithm in Table I.

Name:	WEIGHTED LEAST SQUARES ALGORITHM
Goal:	Compute average weighted least squares
Requires:	$B \in \mathbb{R}^{n\times n}$, $c \in \mathbb{R}^n$, and $M \in \mathbb{R}^{n\times p}$
Assumes:	(i) Network topology modeled by G (ii) B invertible, with non-vanishing diagonal entries, and such that $b_{ij} \neq 0$ implies agent i and j are neighbors in G (iii) Agent $i \in \{1,\dots,n\}$ knows $\text{row}_i(B) \in \mathbb{R}^n$, $c_i \in \mathbb{R}$, $\text{row}_i(M) \in \mathbb{R}^p$

Initialization:

1: $y(0) = c \in \mathbb{R}^n$, $\gamma \in \mathbb{R}_{>0}$, and $h \in (0, 2/n)$

2: $v(0) = w(0) = (\text{row}_1(M)z_1, \dots, \text{row}_n(M)z_n) \in (\mathbb{R}^p)^n$

Agent $i \in \{1,\dots,n\}$ executes concurrently

1: Jacobi overrelaxation algorithm, for $\ell \in \mathbb{Z}_{\geq 0}$

$$y_i(\ell+1) = (1-h)y_i(\ell) - h\frac{1}{b_{ii}}\Big(\sum_{j\neq i} b_{ij}y_j(\ell) - c_i\Big)$$

2: Dynamic average consensus algorithm, for $\tau \in \mathbb{R}_{\geq 0}$

$$\begin{aligned}\frac{dv_i}{d\tau} &= \gamma(u_i(\tau) - v_i(\tau)) - \sum_{j\neq i} a_{ij}(v_i(\tau) - v_j(\tau)) \\ &\quad + \sum_{j\neq i} a_{ij}(w_i(\tau) - w_j(\tau)), \\ \frac{dw_i}{d\tau} &= -\sum_{j\neq i} a_{ij}(v_i(\tau) - v_j(\tau)),\end{aligned}$$

where $\mathcal{A}(G) = (a_{ij})$ and $\tau \mapsto u(\tau) \in (\mathbb{R}^p)^n$ is given by $u(\tau) = (\text{row}_1(M)y_1(\lfloor\tau\rfloor), \dots, \text{row}_n(M)y_n(\lfloor\tau\rfloor))$.

TABLE I
WEIGHTED LEAST SQUARES ALGORITHM.

Proposition 5.2: Consider the WEIGHTED LEAST SQUARES ALGORITHM described in Table I. For $B \in \mathbb{R}^{n\times n}$ invertible, $c \in \mathbb{R}^n$, and $M \in \mathbb{R}^{n\times p}$, define $\texttt{WLS}(B,c,M) : \mathbb{R}_{\geq 0} \to (\mathbb{R}^p)^n$ and $\texttt{JOR}(B,c) : \mathbb{R}_{\geq 0} \to \mathbb{R}^n$ by, respectively,

$$\texttt{WLS}(B,c,M)(\tau) = v(\tau) \quad \text{and} \quad \texttt{JOR}(B,c)(\tau) = y(\lfloor\tau\rfloor),$$

where v and y are defined in Table I. Then,

(i) the WEIGHTED LEAST SQUARES ALGORITHM is distributed over G, in the sense that agent $i \in \{1,\dots,n\}$ can compute $\texttt{WLS}_i(B,c,M)$ and $\texttt{JOR}_i(B,c)$ with information provided by its neighboring agents in G;

(ii) the function $\texttt{JOR}(B,c)$ verifies

$$\texttt{JOR}(B,c)(\tau) \to B^{-1}c \quad \text{as } \tau \to +\infty.$$

(iii) if G is connected, the function $\texttt{WLS}(B,c,M)$ verifies

$$\texttt{WLS}_i(B,c,M)(\tau) \to \frac{1}{n}M^TB^{-1}c \quad \text{as } \tau \to +\infty,$$

for all $i \in \{1,\dots,n\}$.

VI. DISTRIBUTED SEQUENTIAL ESTIMATION

In this section we introduce the DISTRIBUTED KRIGING ALGORITHM. The underlying idea is that, instead of working directly with the posterior predictive distributions obtained in Section IV-C, each agent performs in a distributed way (i) the sequential parameter estimation described in Section IV-A and (ii) the sequential simple kriging described in Section IV-B. From these two constructions, each agent can then compute the desired posterior predictive distributions. The implementation of both (i) and (ii) relies on the WEIGHTED LEAST SQUARES ALGORITHM introduced in Section V.

Name:	DISTRIBUTED KRIGING ALGORITHM
Goal:	Compute Bayesian universal kriging predictors for the spatial field and its gradient
Assumes:	(i) $\mathcal{G}_{R\text{-disk}}$ is connected along evolution (ii) Initially all agents know $\beta \sim N(\nu, V)$

Initialization: $\omega_{\leq -1} = \nu$ and $W_{\leq -1} = V$

At time $k \in \mathbb{Z}_{\geq 0}$, agent $i \in \{1,\dots,n\}$

1: measures $\boldsymbol{Y}_i(k) = Y_i(k)$, sets $\text{row}_i(\boldsymbol{X}(k)) = x(p_i(k))^T$

2: acquires location of neighbors in $\mathcal{G}_{R\text{-disk}}(p_1(k),\dots,p_n(k))$ and computes $\text{row}_i(\boldsymbol{\Sigma}(k)_\sigma)$

3: **for** $j = 1$ to p **do**

4: executes the WEIGHTED LEAST SQUARES ALGORITHM over $\mathcal{G}_{R\text{-disk}}(p_1(k),\dots,p_n(k))$ for $(\boldsymbol{\Sigma}(k)_\sigma, \text{col}_j(\boldsymbol{X}(k)), \boldsymbol{X}(k))$

5: executes the WEIGHTED LEAST SQUARES ALGORITHM over $\mathcal{G}_{R\text{-disk}}(p_1(k),\dots,p_n(k))$ for $(\boldsymbol{\Sigma}(k)_\sigma, \boldsymbol{Y}(k), \boldsymbol{X}(k))$

6: computes weighted least squares estimate

$$\begin{aligned}\text{cov}(\widehat{\beta}_{\text{LS}}(k)) &:= \frac{1}{n}\big(\texttt{WLS}_i(\boldsymbol{\Sigma}(k)_\sigma, \boldsymbol{X}(k), \boldsymbol{X}(k))(\infty)\big)^{-1} \\ \widehat{\beta}_{\text{LS}}(k) &:= \big(\texttt{WLS}_i(\boldsymbol{\Sigma}(k)_\sigma, \boldsymbol{X}(k), \boldsymbol{X}(k))(\infty)\big)^{-1} \\ &\quad \texttt{WLS}_i(\boldsymbol{\Sigma}(k)_\sigma, \boldsymbol{Y}(k), \boldsymbol{X}(k))(\infty)\end{aligned}$$

7: fuses with previous information

$$\begin{aligned}W_{\leq k} &= \big(\text{cov}(\widehat{\beta}_{\text{LS}}(k))^{-1} + W_{\leq k-1}^{-1}\big)^{-1} \\ \omega_{\leq k} &= W_{\leq k}\big(\text{cov}(\widehat{\beta}_{\text{LS}}(k))^{-1}\widehat{\beta}_{\text{LS}}(k) + W_{\leq k-1}^{-1}\omega_{\leq k-1}\big),\end{aligned}$$

8: sets variables

$$\begin{aligned}(\boldsymbol{\Sigma}(k)_\sigma^{-1}\boldsymbol{Y}(k))_i &:= \texttt{JOR}_i(\boldsymbol{\Sigma}(k)_\sigma, \boldsymbol{Y}(k))(\infty), \\ \text{row}_i(\boldsymbol{\Sigma}(k)_\sigma^{-1}\boldsymbol{X}(k)) &:= \text{row}_i(\texttt{JOR}(\boldsymbol{\Sigma}(k)_\sigma, \boldsymbol{X}(k))(\infty))\end{aligned}$$

9: computes predictors at $s \in B(p_i(k), R-r)$

$$\begin{aligned}E(Z(s,k) \mid (\boldsymbol{Y}_{\leq k}, \beta)) &= x(s)^T\beta + \sum_{\|s-p_j(k)\|\leq r} \boldsymbol{\gamma}_j(s,k) \\ &\quad \Big(\big(\boldsymbol{\Sigma}(k)_\sigma^{-1}\boldsymbol{Y}(k)\big)_j - \text{row}_j(\boldsymbol{\Sigma}(k)_\sigma^{-1}\boldsymbol{X}(k))\beta\Big), \\ E(\nabla Z(s,k) \mid (\boldsymbol{Y}_{\leq k}, \beta)) &= \nabla x(s)^T\beta + \sum_{\|s-p_j(k)\|\leq r} \boldsymbol{\nabla\gamma}_j(s,k) \\ &\quad \big(\big(\boldsymbol{\Sigma}(k)_\sigma^{-1}\boldsymbol{Y}(k)\big)_j - \text{row}_j(\boldsymbol{\Sigma}(k)_\sigma^{-1}\boldsymbol{X}(k))\beta\big),\end{aligned}$$

10: computes Bayesian universal kriging predictors at $s \in B(p_i(k), R-r)$ and $t \in [k, k+1)$

$$E(Z(s,t) \mid \boldsymbol{Y}_{\leq k}) \quad \text{and} \quad E(\nabla Z(s,t) \mid \boldsymbol{Y}_{\leq k}).$$

TABLE II
THE DISTRIBUTED KRIGING ALGORITHM.

Lemma 6.1: The DISTRIBUTED KRIGING ALGORITHM outlined in Table II allows each network agent to compute, at any time, the Bayesian universal kriging predictors of the spatial field and its gradient on a ball centered at its current location of radius $R-r$, and only requires communication with neighboring agents in $\mathcal{G}_{R\text{-disk}}$.

Next, we detail the steps of the algorithm.

A. Distributed sequential parameter estimation

Here, we describe the strategy that network agents implement to compute the sequential parameter estimation of Section IV-A. Recall that $\boldsymbol{Y}(k)$ denotes the measurements taken by the network at time k, with associated covariance matrix $\boldsymbol{\Sigma}(k)_\sigma = \boldsymbol{\Sigma}(k) + \sigma I_n$. Assume the mean $\omega_{\leq k-1}$ and the covariance matrix $W_{\leq k-1}$ of the posterior distribution of β with data collected up to time $k-1$ are known to each agent. This is certainly the case for $k = 0$, where,

$$\omega_{\leq -1} = \nu, \quad W_{\leq -1} = V.$$

According to Lemma 4.1, agent $i \in \{1, \dots, n\}$ can compute $\omega_{\leq k}$ and $W_{\leq k}$ if it has access to

$$\boldsymbol{X}(k)^T \boldsymbol{\Sigma}(k)_\sigma^{-1} \boldsymbol{Y}(k) \in \mathbb{R}^p, \tag{18a}$$

$$\boldsymbol{X}(k)^T \boldsymbol{\Sigma}(k)_\sigma^{-1} \boldsymbol{X}(k) \in \mathbb{R}^{p \times p}. \tag{18b}$$

This is equivalent to the computation of the weighted least squares estimate of β and its variance with data $\boldsymbol{Y}(k)$,

$$\operatorname{cov}(\widehat{\beta}_{\text{LS}}(k)) = (\boldsymbol{X}(k)^T \boldsymbol{\Sigma}(k)_\sigma^{-1} \boldsymbol{X}(k))^{-1},$$
$$\widehat{\beta}_{\text{LS}}(k) = (\boldsymbol{X}(k)^T \boldsymbol{\Sigma}(k)_\sigma^{-1} \boldsymbol{X}(k))^{-1} \boldsymbol{X}(k)^T \boldsymbol{\Sigma}(k)_\sigma^{-1} \boldsymbol{Y}(k).$$

Next, we describe how the network computes (18).

1) Distributed computation of the covariance matrix of the weighted least squares estimate: The network computes (18b) as follows. For $j \in \{1, \dots, p\}$, agent i has access to the ith component of $\operatorname{col}_j(\boldsymbol{X}(k)) \in \mathbb{R}^n$, and to $\operatorname{row}_i(\boldsymbol{X}(k)) = x(p_i(k))^T$. Now, (8) guarantees that agent i can compute $\operatorname{row}_i(\boldsymbol{\Sigma}(k)_\sigma)$ by knowing the location of its $\mathcal{G}_{R\text{-disk}}$-neighbors. Hence, for $\mathcal{G}_{R\text{-disk}}(p_1(k), \dots, p_n(k))$ connected, the execution of WEIGHTED LEAST SQUARES ALGORITHM with $B = \boldsymbol{\Sigma}(k)_\sigma$, $c = \operatorname{col}_j(\boldsymbol{X}(k))$, and $M = \boldsymbol{X}(k)$ guarantees, cf. Proposition 5.2,

$$\mathtt{WLS}_i(\boldsymbol{\Sigma}(k)_\sigma, \operatorname{col}_j(\boldsymbol{X}(k)), \boldsymbol{X}(k))(\tau) \longrightarrow \frac{1}{n} \boldsymbol{X}(k)^T \boldsymbol{\Sigma}(k)_\sigma^{-1} \operatorname{col}_j(\boldsymbol{X}(k)),$$

as $\tau \to +\infty$, for all $i \in \{1, \dots, n\}$. Hence, the execution of p instances of the WEIGHTED LEAST SQUARES ALGORITHM allows agent i to compute the time-dependent matrix $\mathtt{WLS}_i(\boldsymbol{\Sigma}(k)_\sigma, \boldsymbol{X}(k), \boldsymbol{X}(k))(\tau)$ given by

$$\big(\mathtt{WLS}_i(\boldsymbol{\Sigma}(k)_\sigma, \operatorname{col}_1(\boldsymbol{X}(k)), \boldsymbol{X}(k)), \dots, \dots, \mathtt{WLS}_i(\boldsymbol{\Sigma}(k)_\sigma, \operatorname{col}_p(\boldsymbol{X}(k)), \boldsymbol{X}(k))\big)(\tau),$$

and with the property that

$$\mathtt{WLS}_i(\boldsymbol{\Sigma}(k)_\sigma, \boldsymbol{X}(k), \boldsymbol{X}(k))(\tau) \to \frac{1}{n} \boldsymbol{X}(k)^T \boldsymbol{\Sigma}(k)_\sigma^{-1} \boldsymbol{X}(k)$$

as $\tau \to +\infty$, for all $i \in \{1, \dots, n\}$. With this information, agent i can compute the covariance matrix

$$\frac{1}{n} \big(\mathtt{WLS}_i(\boldsymbol{\Sigma}(k)_\sigma, \boldsymbol{X}(k), \boldsymbol{X}(k))(\tau)\big)^{-1} \to \operatorname{cov}(\widehat{\beta}_{\text{LS}}(k)),$$

as $\tau \to +\infty$, for all $i \in \{1, \dots, n\}$.

2) Distributed computation of the weighted least squares estimate: The network computes (18a) as follows. Agent i has access to $\operatorname{row}_i(\boldsymbol{\Sigma}(k)_\sigma)$, $\boldsymbol{Y}(k)_i = Y_i(k)$ and $\operatorname{row}_i(\boldsymbol{X}(k)) = x(p_i(k))^T$. Therefore, for $\mathcal{G}_{R\text{-disk}}(p_1(k), \dots, p_n(k))$ connected, the execution of the WEIGHTED LEAST SQUARES ALGORITHM with $B = \boldsymbol{\Sigma}(k)_\sigma$, $c = \boldsymbol{Y}(k)$, and $M = \boldsymbol{X}(k)$ guarantees, cf. Proposition 5.2,

$$\mathtt{WLS}_i(\boldsymbol{\Sigma}(k)_\sigma, \boldsymbol{Y}(k), \boldsymbol{X}(k))(\tau) \to \frac{1}{n} \boldsymbol{X}(k)^T \boldsymbol{\Sigma}(k)_\sigma^{-1} \boldsymbol{Y}(k),$$

as $\tau \to +\infty$, for all $i \in \{1, \dots, n\}$. With this information, agent i can compute the weighted least squares estimate

$$\big(\mathtt{WLS}_i(\boldsymbol{\Sigma}(k)_\sigma, \boldsymbol{X}(k), \boldsymbol{X}(k))(\tau)\big)^{-1} \mathtt{WLS}_i(\boldsymbol{\Sigma}(k)_\sigma, \boldsymbol{Y}(k), \boldsymbol{X}(k))(\tau) \longrightarrow \widehat{\beta}_{\text{LS}}(k),$$

as $\tau \to +\infty$, for all $i \in \{1, \dots, n\}$. Each agent fuses its knowledge $\beta \sim N_p(\omega_{\leq t-1}, W_{\leq t-1})$ with the information obtained as described in Sections VI-A.1 and VI-A.2 to compute the posterior predictive distribution $\beta \sim N_p(\omega_{\leq t}, W_{\leq t})$.

B. Distributed sequential simple kriging

Here, we describe the strategy that network agents implement to compute the sequential simple kriging of Section IV-B. According to Lemma 4.2, to compute the means of the conditional distributions of the spatial field and its gradient at s, we are interested in the distributed calculation of

$$\boldsymbol{\gamma}(s,k) \in \mathbb{R}^n, \qquad \boldsymbol{\nabla\gamma}(s,k) \in \mathbb{R}^{n \times d}, \tag{19a}$$

$$\boldsymbol{\Sigma}(k)_\sigma^{-1} \boldsymbol{Y}(k) \in \mathbb{R}^n, \quad \boldsymbol{\Sigma}(k)_\sigma^{-1} \boldsymbol{X}(k) \in \mathbb{R}^{n \times p}. \tag{19b}$$

Regarding (19a), the jth components of $\boldsymbol{\gamma}(s,k)$ and $\boldsymbol{\nabla\gamma}(s,k)$ can only be nonvanishing if agent j is within r-distance of s, that is, $\|s - p_j(k)\| \leq r$. Therefore, any agent i can compute all the nonvanishing components in $\boldsymbol{\gamma}(s,k)$ and $\boldsymbol{\nabla\gamma}(s,k)$ if $B(s,r) \subset B(p_i(k), R)$. Noting that this is equivalent to $s \in B(p_i(k), R-r)$, we deduce that agent i can compute (19a) in a distributed way for any $s \in B(p_i(k), R-r)$.

Regarding (19b), as a by-product of the executions of the WEIGHTED LEAST SQUARES ALGORITHM performed in Section VI-A, at time $k \in \mathbb{Z}_{\geq 0}$, agent $i \in \{1, \dots, n\}$ has available the ith component of the functions

$$\mathtt{JOR}(\boldsymbol{\Sigma}(k)_\sigma, \boldsymbol{Y}(k)) : \mathbb{R}_{\geq 0} \to \mathbb{R}^n,$$
$$\mathtt{JOR}(\boldsymbol{\Sigma}(k)_\sigma, \operatorname{col}_j(\boldsymbol{X}(k))) : \mathbb{R}_{\geq 0} \to \mathbb{R}^n, \quad j \in \{1, \dots, p\}.$$

Let us define $\mathtt{JOR}(\boldsymbol{\Sigma}(k)_\sigma, \boldsymbol{X}(k)) : \mathbb{R}_{\geq 0} \to \mathbb{R}^{n \times p}$ by

$$\mathtt{JOR}(\boldsymbol{\Sigma}(k)_\sigma, \boldsymbol{X}(k))(\tau) = \big(\mathtt{JOR}(\boldsymbol{\Sigma}(k)_\sigma, \operatorname{col}_1(\boldsymbol{X}(k))), \dots, \mathtt{JOR}(\boldsymbol{\Sigma}(k)_\sigma, \operatorname{col}_p(\boldsymbol{X}(k)))\big)(\tau).$$

Note that agent i has access to the ith row of $\mathtt{JOR}(\boldsymbol{\Sigma}(k)_\sigma, \boldsymbol{X}(k))$. By Proposition 5.2, we have

$$\mathtt{JOR}(\boldsymbol{\Sigma}(k)_\sigma, \boldsymbol{Y}(k))(\tau) \longrightarrow \boldsymbol{\Sigma}(k)_\sigma^{-1} \boldsymbol{Y}(k) \in \mathbb{R}^n,$$
$$\mathtt{JOR}(\boldsymbol{\Sigma}(k)_\sigma, \boldsymbol{X}(k))(\tau) \longrightarrow \boldsymbol{\Sigma}(k)_\sigma^{-1} \boldsymbol{X}(k) \in \mathbb{R}^{n \times p}.$$

Finally, note that agent $i \in \{1, \dots, n\}$ has access to both $\mathtt{JOR}_j(\boldsymbol{\Sigma}(k)_\sigma, \boldsymbol{Y}(k)) \in \mathbb{R}$ and $\operatorname{row}_j(\mathtt{JOR}(\boldsymbol{\Sigma}(k)_\sigma, \boldsymbol{X}(k))) \in \mathbb{R}^p$ for all j such that $p_i(k)$ and $p_j(k)$ are neighbors in $\mathcal{G}_{R\text{-disk}}$. Therefore, we deduce the following result.

Proposition 6.2: For all $k \in \mathbb{Z}_{\geq 0}$ and all $s \in B(p_i(k), R - r)$, $E(Z(s,k) \mid (\boldsymbol{Y}_{\leq k}, \beta))$ and $E(\nabla Z(s,k) \mid (\boldsymbol{Y}_{\leq k}, \beta))$ can be casted as in `11:` of Table II, and therefore, are computable by agent i over $\mathcal{G}_{R\text{-disk}}$.

Remark 6.3: (Execution of DISTRIBUTED KRIGING ALGORITHM*)* It is reasonable to assume that the order of magnitude of the time required by individual agents to communicate and compute is smaller than the one required to move. Additionally, according to Section III, measurements are taken at instants of time in $\mathbb{Z}_{>0}$. Hence we assume that the computations described in Sections IV-A and IV-B run on a time scale τ much faster than the time scale t. •

VII. DISTRIBUTED SPATIAL GRADIENT ASCENT

This section presents a motion coordination algorithm that the network can implement to find local maxima of the spatial field. As exemplified in the introduction, this task has practical applications in a variety of scenarios. At any $t \in \mathbb{R}_{\geq 0}$, the DISTRIBUTED KRIGING ALGORITHM in Table II allows agent $i \in \{1, \ldots, n\}$ to compute the expected value of the spatial field and its gradient in $B(p_i(\lfloor t \rfloor), R-r)$. Each agent can then implement a gradient ascent strategy

$$\dot{p}_i(t) = E(\nabla Z(p_i(t), t) \mid \boldsymbol{Y}_{\leq t}). \tag{20}$$

Feedback is present through the dependence of the measurements $\boldsymbol{Y}_{\leq t}$ on the network configuration. Because new measurements are taken at times in $\mathbb{Z}_{\geq 0}$, the resulting trajectory of agent i is continuous and piecewise differentiable.

Remark 7.1: The convergence of the WEIGHTED LEAST SQUARES ALGORITHM is asymptotic, and hence the values of the estimation of the parameter, the spatial field, and its gradient are exact up to some numerical tolerance. Therefore, the network implements an approximation of (20). •

Next, we characterize the asymptotic convergence of the gradient ascent when no measurement errors are present.

Proposition 7.2: Assume the superlevel sets of Z are compact, and that there are no measurement errors. Then, any network trajectory under (20) starting from $\mathcal{S} = (\mathbb{R}^d)^n \setminus \{(p_1, \ldots, p_n) \in (\mathbb{R}^d)^n \mid p_i = p_j \text{ with } i \neq j\}$ and such that agents remain connected satisfies $E(\nabla Z(p_i(t)) \mid \boldsymbol{Z}_{\leq t}) \to 0$, $i \in \{1, \ldots, n\}$, as $t \to \infty$, with probability one.

Figure 1 shows an execution of the gradient ascent (20). Note that the network topology changes along the evolution.

VIII. CONCLUSIONS

We have considered a robotic sensor network taking successive measurements of a process of interest and trying to find its maxima. We have introduced a statistical framework to estimate the distribution of the spatial field and of its gradient. We have developed a distributed spatial estimation algorithm, and synthesized a motion coordination strategy that makes network agents find critical points of the field with probability one in case of no measurement noise.

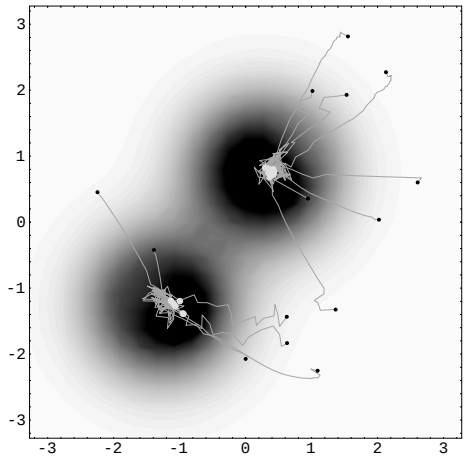

Fig. 1. Distributed gradient ascent cooperative strategy (20) implemented by a robotic sensor network with 14 agents. The spatial field has mean $\mu(s) = .3 + 1.2\,\mathrm{e}^{-\|s-(.25,.75)\|^2} + 1.1\,\mathrm{e}^{-\|s+(1.25,1.25)\|^2}$ and covariance structure determined by $K(s) = e^{-5\|s\|^2}$ for $\|s\| \leq r = 1.75$. We depict the contour plot of the posterior mean. Initially agents know $\beta \sim N_3(0, I_3)$. The communication radius is $R = 2.75$, the control authority of each agent is bounded by $u_{\max} = .25$, and the noise sensor error variance is $\sigma = .15$. The black disks depict the (randomly generated) initial agent positions and the gray disks depict the agent positions after 38 seconds.

REFERENCES

[1] P. Ögren, E. Fiorelli, and N. E. Leonard, "Cooperative control of mobile sensor networks: Adaptive gradient climbing in a distributed environment," *IEEE Transactions on Automatic Control*, vol. 49, no. 8, pp. 1292–1302, 2004.

[2] K. M. Lynch, I. B. Schwartz, P. Yang, and R. A. Freeman, "Decentralized environmental modeling by mobile sensor networks," *IEEE Transactions on Robotics*, 2006. Submitted.

[3] S. Martínez, "Distributed representation of spatial fields through an adaptive interpolation scheme," in *American Control Conference*, (New York City, NY), pp. 2750–2755, 2007.

[4] 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.

[5] D. P. Bertsekas and J. N. Tsitsiklis, *Parallel and Distributed Computation: Numerical Methods*. Belmont, MA: Athena Scientific, 1997.

[6] L. Xiao, S. Boyd, and S. Lall, "A scheme for asynchronous distributed sensor fusion based on average consensus," in *International Conference on Information Processing in Sensor Networks (IPSN'05)*, (Los Angeles, CA), pp. 63–70, Apr. 2005.

[7] D. P. Spanos, R. Olfati-Saber, and R. M. Murray, "Distributed sensor fusion using dynamic consensus," in *IFAC World Congress*, (Prague, CZ), July 2005. Electronic proceedings.

[8] D. P. Spanos, R. Olfati-Saber, and R. M. Murray, "Dynamic consensus for mobile networks," in *IFAC World Congress*, (Prague, Czech Republic), July 2005.

[9] R. A. Freeman, P. Yang, and K. M. Lynch, "Stability and convergence properties of dynamic average consensus estimators," in *IEEE Conf. on Decision and Control*, (San Diego, CA), pp. 398–403, 2006.

[10] N. A. C. Cressie, *Statistics for Spatial Data*. New York: Wiley, 1993. revised edition.

[11] M. L. Stein, *Interpolation of Spatial Data. Some Theory for Kriging*. Springer Series in Statistics, New York: Springer Verlag, 1999.

[12] S. Banerjee, A. E. Gelfand, and C. F. Sirmans, "Directional rates of change under spatial process models," *Journal of the American Statistical Association*, vol. 98, no. 464, pp. 946–954, 2003.

[13] L. C. G. Rogers and D. Williams, *Diffusions, Markov Processes and Martingales*, vol. 1. Cambridge, UK: Cambridge University Press, 2nd ed., 2000.

[14] H. J. Kushner, "On the stability of stochastic dynamical systems," *PNAS*, vol. 53, pp. 8–12, 1965.

[15] J. Cortés, "Distributed Kriged Kalman filter for spatial estimation," 2007. Submitted. Electronically available at http://www.soe.ucsc.edu/~jcortes.

[16] J. Cortés, S. Martínez, and F. Bullo, "Robust rendezvous for mobile autonomous agents via proximity graphs in arbitrary dimensions," *IEEE Transactions on Automatic Control*, vol. 51, no. 8, pp. 1289–1298, 2006.

[17] F. E. Udwadia, "Some convergence results related to the JOR iterative method for symmetric, positive-definite matrices," *Applied Mathematics and Computation*, vol. 47, no. 1, pp. 37–45, 1992.