Distributed Kriged Kalman filter for spatial estimation

Jorge Cortés

***Abstract*—This paper considers robotic sensor networks performing spatially-distributed estimation tasks. A robotic sensor network is deployed in an environment of interest, and takes successive point measurements of a dynamic physical process modeled as a spatio-temporal random field. Taking a Bayesian perspective on the Kriging interpolation technique from geostatistics, we design the DISTRIBUTED KRIGED KALMAN FILTER for predictive inference 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 average consensus algorithms. As an application of the proposed algorithm, we design a gradient ascent cooperative strategy and analyze its convergence properties in the absence of measurement errors via stochastic Lyapunov functions. We illustrate our results in simulation.**

I. INTRODUCTION

Consider a robotic sensor network taking successive measurements of a dynamic physical process modeled as a spatio-temporal random field. Our objective is to design a distributed estimation algorithm that enables the network to obtain consistent and statistically sound representations of the spatial field. Arguably, the availability of such representations to the network agents is necessary to tackle other sensing tasks related with the physical process, such as optimal estimation, localization of critical points, or identification of areas of rapid variability. These tasks are relevant in multiple scenarios, including environmental monitoring, oceanographic exploration, and atmospheric research, when one might be interested in finding higher pollutant concentrations, areas of maximum salinity, or locations where algae are abundant.

Literature review: In geostatistics, spatial processes modeled as random fields are estimated via Kriging interpolation techniques [1], [2]. Simple Kriging assumes that the mean of the random field is constant and known a priori. Universal Kriging, instead, considers the setup where the mean function is an unknown combination of known basis functions. For processes that evolve in time, [3], see also [4], develops a universal Kriging approach termed Kriged Kalman filter that combines the time and spatial components of the field. The work [5] presents an inferential framework for directional gradients of spatial fields based on point-referenced data.

In cooperative control, [6] proposes a decentralized information filter for parameter estimation based on all-to-all communication, and applies it to tracking, localization, and map building. [7] designs network coordination strategies to seek out local optima of a deterministic, static field using noisy measurements and all-to-all network communication. The field is represented by an affine function. Instead, [8] considers a dynamic field represented by an unknown linear combination of known functions whose coefficients evolve stochastically driven by white noise. [8] develops distributed optimal estimation techniques for networks with connected communication topology and sensor measurements corrupted by white noise. The measurements taken by individual network agents are uncorrelated. Objective analysis techniques are employed in [9] to find, in restricted parameterized families of curves, network trajectories that optimize the off-line, centralized estimation of an environmental field whose mean is a priori known and whose covariance is separable. During the evolution, individual agents do not communicate field measurements to other neighbors or possess a representation of the spatial field. Instead, the fusion of the data is performed at the end of the experiment. The works [10], [11] introduce distributed data fusion algorithms based on averaging consensus that work under the assumption that sensor measurements are uncorrelated. Dynamic consensus algorithms that allow to track the average of a given time-varying signal are studied in [12], [13], [14]. Other related works include [15], [16], where decentralized Kalman filtering procedures are developed that work under the assumption of all-to-all communication. Parallel and distributed algorithms for static networks are thoroughly studied in [17]. Finally, stability analysis tools for stochastic systems include the supermartingale convergence theorem [18] and stochastic Lyapunov functions [19].

Statement of contributions: The contributions of this paper are the following: (i) the formulation of the spatio-temporal field estimation via Bayesian universal Kriging, and the incorporation of statistically sound gradient information 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 the DISTRIBUTED KRIGED KALMAN FILTER for predictive inference 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.

Submitted May 3, 2007, revised January 11, 2009, final version June 16, 2009. An early version of this work appeared in the 2007 IEEE Conference on Decision and Control

Jorge Cortés is with the Department of Mechanical and Aerospace Engineering, University of California, San Diego, USA, cortes@ucsd.edu, http://tintoretto.ucsd.edu/jorge

Organization: The paper is organized as follows. Section II presents basic notions on random spatial fields. 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 via Bayesian universal Kriging. Section V presents a distributed algorithm to compute weighted least squares estimates when sensor measurements are correlated. This algorithm is then used in Section VI to design a distributed implementation of the sequential estimation discussed in Section IV. Section VII proposes a cooperative strategy to localize critical points of spatial fields and analyzes its convergence properties. Section VIII presents our conclusions and ideas for future work.

Notation: Let $\mathbb{Z}$, $\mathbb{Z}_{>0}$, $\mathbb{Z}_{\geq 0}$, $\mathbb{R}$, $\mathbb{R}_{>0}$ and $\mathbb{R}_{\geq 0}$ denote, respectively, the set of integer, positive integer, non-negative integer, real, positive real, and non-negative real numbers. Let $\delta : \mathbb{R} \to \{0, 1\}$ denote the Dirac delta function defined by $\delta(t) = 0$ for $t \neq 0$, and $\delta(0) = 1$. 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 $A \in \mathbb{R}^{d_1 \times d_2}$, let $\text{row}_i(A) \in \mathbb{R}^{d_2}$ and $\text{col}_j(A) \in \mathbb{R}^{d_1}$ 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, we take $V = \{1, \dots, n\}$. The adjacency matrix of G is the matrix $\mathcal{A}(G) = (a_{ij}) \in \mathbb{R}^{n \times n}$ defined by $a_{ij} = 1$ if $(i, j) \in E$, and $a_{ij} = 0$ otherwise. We will often simply denote it by $\mathcal{A}$. Throughout the paper, we use the math boldface font to emphasize the dependence of the corresponding quantity on the specific network configuration where it is evaluated. This allows us to write more concise expressions.

II. Random spatial fields

In this section we review important notions on random spatial fields. The interested reader is referred to [1], [2] for more details. Let us start with some basic definitions. Let Z be a random spatial field on $\mathbb{R}^d$, $d \in \mathbb{Z}_{>0}$, with positive definite covariance function $C : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}_{\geq 0}$,

$$\text{Cov}(Z(s), Z(s')) = C(s, s').$$

The field Z is *stationary* on $\mathbb{R}^d$ if $C(s, s') = K(s - s')$, for $K : \mathbb{R}^d \to \mathbb{R}_{\geq 0}$, and *isotropic* on $\mathbb{R}^d$ if $C(s, s') = \widetilde{K}(\|s - s'\|)$, for $\widetilde{K} : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$. Throughout the paper, we deal with stationary random fields. When modeling physical processes, it is common for a random field to be stationary over a strict subset of $\mathbb{R}^d$ instead of the whole Euclidean space. For instance, the assumption of stationarity is reasonable for a temperature field considered over a small enough region of the ocean. However, over larger spatial domains, other physical phenomena might cause smaller correlation ranges in particular areas that invalidate the stationarity assumption. The ensuing discussion is also valid for random fields that are stationary on an open subset of $\mathbb{R}^d$.

Predictive inference of a spatial field Z at arbitrary points given measurements at arbitrary locations can be done via the joint distribution. For concreteness, let $Z(\cdot) \sim GP(\mu(\cdot), K(\cdot))$ be a stationary Gaussian process. Given measurements $Z(p_1), \dots, Z(p_n)$ of the spatial field Z at locations $p_1, \dots, p_n$, and $s \in \mathbb{R}^d$, define the following shorthand notation for convenience,

$$\begin{aligned} \boldsymbol{Z} &= (Z(p_1), \dots, Z(p_n)) \in \mathbb{R}^n, \\ \boldsymbol{\mu} &= (\mu(p_1), \dots, \mu(p_n)) \in \mathbb{R}^n, \\ \boldsymbol{\Sigma} &= (K(p_i - p_j)) \in \mathbb{R}^{n \times n}, \\ \boldsymbol{\gamma}(s)^T &= (K(s - p_1), \dots, K(s - p_n)) \in \mathbb{R}^n. \end{aligned}$$

Then $(\boldsymbol{Z}, Z(s))$ is distributed as the $n+1$ dimensional normal distribution

$$N_{n+1}\left(\begin{pmatrix} \boldsymbol{\mu} \\ \mu(s) \end{pmatrix}, \begin{pmatrix} \boldsymbol{\Sigma} & \boldsymbol{\gamma}(s) \\ \boldsymbol{\gamma}(s)^T & K(0) \end{pmatrix} \right).$$

Consequently, the conditional predictive distribution of the spatial field at s given observations at $p_1, \dots, p_n$ is the normal distribution

$$\begin{aligned} Z(s) \mid \boldsymbol{Z} \sim N\big(&\mu(s) + \boldsymbol{\gamma}(s)^T \boldsymbol{\Sigma}^{-1} (\boldsymbol{Z} - \boldsymbol{\mu}), \\ &K(0) - \boldsymbol{\gamma}(s)^T \boldsymbol{\Sigma}^{-1} \boldsymbol{\gamma}(s)\big). \end{aligned} \tag{1}$$

The conditional mean in (1) is known in the geostatistics literature as the simple Kriging predictor, and the conditional variance in (1) is the corresponding mean-squared prediction error. In general, perfect observations of the spatial field are not available, and the mean and the covariance structure are only known up to a certain number of parameters. We will discuss the estimation problem in these more general terms in Section IV.

When considering dynamic processes, we restrict our attention to spatio-temporal random fields on $\mathbb{R}^d \times \mathbb{R}_{\geq 0}$ with separable covariance functions, i.e., of the form

$$\text{Cov}(Z(s, t), Z(s', t')) = C_1(s, s')\, C_2(t, t'),$$

where $C_1 : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}_{\geq 0}$ and $C_2 : \mathbb{R}_{\geq 0} \times \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$. A stationary spatio-temporal random field of this form verifies that $C_1(s, s') = K_1(s - s')$ and $C_2(t, t') = K_2(t - t')$, for $K_1 : \mathbb{R}^d \to \mathbb{R}_{\geq 0}$ and $K_2 : \mathbb{R} \to \mathbb{R}_{\geq 0}$. Note that the above discussion is also valid for predictive inference of a spatio-temporal field at a fixed instant of time.

A. Gradient random spatial fields

The discussion here follows [5] and can be easily extended for spatio-temporal fields. For concreteness, we restrict our attention to stationary random fields. 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, for $s \in \mathbb{R}^d$,

$$D_u Z(s) = \lim_{h \to 0} \frac{Z(s + hu) - Z(s)}{h},$$

if the limit understood in the L_2 sense exists. 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, if Z is mean square differentiable at s_0, then $D_u Z(s_0) = \nabla Z(s_0)^T u$, for all $u \in \mathbb{R}^d$ (where the equality should be understood in the L_2-sense). In particular,

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

Throughout the paper, we will 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

$$\begin{aligned}\operatorname{Cov}((Z(s), \nabla Z(s)), (Z(s'), \nabla Z(s'))) = \\ \begin{pmatrix} K(s-s') & -(\nabla K(s-s'))^T \\ \nabla K(s-s') & -\operatorname{H}(K)(s-s') \end{pmatrix}, \end{aligned} \quad (2)$$

where $\nabla K : \mathbb{R}^d \to \mathbb{R}^d$ and $\operatorname{H}(K) : \mathbb{R}^d \to \mathbb{R}^{d \times d}$ denote, respectively, the gradient and the Hessian of the function K. This joint distribution allows predictive inference for the gradient at arbitrary points given measurements of the random field at arbitrary locations. For concreteness, let $Z(s) \sim GP(\mu(s), K(\cdot))$, with $\mu : \mathbb{R}^d \to \mathbb{R}$ continuously differentiable. Given measurements $Z(p_1), \dots, Z(p_n)$ of the spatial field Z at locations $p_1, \dots, p_n$, and $s \in \mathbb{R}^d$, according to (2), $(\boldsymbol{Z}, \nabla Z(s))$ is distributed as the $n+d$ dimensional normal distribution

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

where $\boldsymbol{\nabla\gamma}(s)^T = (\nabla K(s-p_1), \dots, \nabla K(s-p_n)) \in \mathbb{R}^{d \times n}$. Consequently, the conditional predictive distribution for the gradient is the d-dimensional normal distribution

$$\begin{aligned}\nabla Z(s) \mid \boldsymbol{Z} \sim N_d\big(&\nabla \mu(s) + \boldsymbol{\nabla\gamma}(s)^T \boldsymbol{\Sigma}^{-1}(\boldsymbol{Z} - \boldsymbol{\mu}), \\ &-\operatorname{H}(K)(0) - \boldsymbol{\nabla\gamma}(s)^T \boldsymbol{\Sigma}^{-1} \boldsymbol{\nabla\gamma}(s)\big).\end{aligned} \quad (3)$$

Section IV studies the conditional predictive distribution of the gradient when the mean of the spatial field is unknown, and perfect observations are not available.

A *critical point* of the spatial field Z is a location $s_* \in \mathbb{R}^d$ such that $\nabla Z(s_*) = 0$. Note that 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 set-up

The objective of this paper is to design distributed estimation algorithms that enable a robotic sensor network to obtain consistent and statistically sound representations of a physical process of interest. In the following, we detail the specific models for the process and the robotic network.

A. Physical process model

We consider a dynamic physical process, i.e., a process that evolves in time, modeled as a spatio-temporal Gaussian random field Z of the form

$$Z(s,k) = \mu(s,k) + \nu(s,k), \quad (4a)$$

$$\mu(s,k) = \int \omega_s(u) \mu(u, k-1) du + \eta(s,k), \quad (4b)$$

where $(s,k) \in \mathbb{R}^d \times \mathbb{Z}_{>0}$ and $\mu : \mathbb{R}^d \times \mathbb{Z}_{>0} \to \mathbb{R}$ is continuously differentiable with respect to its first argument. Here, ν captures small-scale variability of the physical process, and the evolution of the mean is determined by the interaction function $\omega_s : \mathbb{R}^d \to \mathbb{R}$ and the stochastic component η. Both ν and η are stationary spatial fields that exhibit temporal variability but have no temporal dynamics associated with them. Formally, both are zero-mean Gaussian random fields with separable covariance structure

$$\begin{aligned}\operatorname{Cov}(\nu(s,k), \nu(s',k')) &= K_\nu(s-s')\,\delta(k-k'), \\ \operatorname{Cov}(\eta(s,k), \eta(s',k')) &= K_\eta(s-s')\,\delta(k-k'),\end{aligned}$$

where δ denotes the Dirac delta function. Note that both ν and η are uncorrelated in time. We assume that the functions $K_\nu, K_\eta : \mathbb{R}^d \to \mathbb{R}_{\geq 0}$ have finite range. Without loss of generality, both ranges are considered equal, that is, there exists $r \in \mathbb{R}_{>0}$ such that

$$K_\nu(s-s') = 0 = K_\eta(s-s') \quad \text{for} \quad \|s-s'\| > r. \quad (5)$$

The approach taken in [3], [4] to deal with the time evolution (4b) of the spatial field mean μ is to consider a truncated expansion. Specifically, if $\{\phi_j : \mathbb{R}^d \to \mathbb{R}\}_{j=1}^\infty$ is a complete and orthonormal sequence of continuously differentiable functions, the mean admits a representation of the form

$$\mu(s,k) = \sum_{j=1}^{\infty} \beta_j(k) \phi_j(s),$$

where, for each $j \in \mathbb{Z}_{>0}$, $\{\beta_j(k)\}_{k=1}^\infty$ is a random time series. Likewise, ω_s admits a decomposition

$$\omega_s(u) = \sum_{l=1}^{\infty} b_l(s) \phi_l(u).$$

The standard procedure is then to truncate the representation of μ and ω_s to, say, the first $m \in \mathbb{Z}_{>0}$ basis elements, and use the orthonormality of the basis to rewrite (4b) as

$$\phi(s)^T \beta(k) = b(s)^T \beta(k-1) + \eta(s,k), \quad (6)$$

where, for simplicity, we use the notation

$$\begin{aligned}\beta(k) &= (\beta_1(k), \dots, \beta_m(k))^T \in \mathbb{R}^m, \\ b(s) &= (b_1(s), \dots, b_m(s))^T \in \mathbb{R}^m, \\ \phi(s) &= (\phi_1(s), \dots, \phi_m(s))^T \in \mathbb{R}^m.\end{aligned}$$

Alternatively, one can set up the problem by directly assuming that the mean μ of Z in (4a) is a linear combination of known functions whose coefficients evolve in time according to (6).

B. Network model

Consider a network of n agents evolving in $\mathbb{R}^d$ according to the first-order dynamics

$$\dot{p}_i = u_i, \quad i \in \{1, \dots, n\}.$$

The control action is bounded $\|u_i\| \leq u_{\max} \in \mathbb{R}_{>0}$, so that an agent can move at most $u_{\max}$ in one second. Agents are equipped with identical sensors, and can take point measurements at their location of the spatial field of interest Z at times

$k \in \mathbb{Z}_{>0}$. The measurement taken by agent i located at p_i at time k is corrupted by white noise according to

$$Y_i(k) = Z(p_i, k) + \epsilon_i, \tag{7}$$

where $\epsilon_i \sim N(0, \sigma)$. Measurement errors are assumed to be independent. For simplicity, the variance σ is assumed to be the same for all agents, although the forthcoming discussion can be generalized to the case of different noise variances for each agent.

Each agent can communicate with other agents located within a distance $R \in \mathbb{R}_{>0}$ from its current position. As we will show later, each agent can construct a distributed representation of the spatial field and of its gradient in a ball of radius $R - r$. Therefore, we will make the assumption that

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

The communication capabilities of the agents induce the network topology corresponding to the R-disk graph $\mathcal{G}_{R\text{-disk}}$. At each network configuration $(p_1, \dots, p_n) \in (\mathbb{R}^d)^n$, the R-disk graph $\mathcal{G}_{R\text{-disk}}(p_1, \dots, p_n)$ is an undirected graph with vertex set $\{p_1, \dots, p_n\}$ and edge set $\{(p_i, p_j) \mid \|p_i - p_j\| \leq R\}$. The R-disk graph is a particular example of the notion of proximity graph, see e.g., [20]. We assume that either the number of network 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 provide a formal notion of the concept of distributed computation of functions and vector fields, see e.g., [20]. For simplicity, here we only use an informal version of this notion, where we characterize a computation as distributed over an undirected graph if each node can perform the computation using only information provided by its neighbors in the graph. •

IV. Sequential estimation of the spatial field and of its gradient

In this section, we take a Bayesian perspective to incorporate previous knowledge into the estimation of the spatial field and of its gradient. The scheme followed here recovers the predictive distribution of the spatial field presented in [3], see also [4], and yields novel information regarding the predictive distribution of the gradient of the spatial field. We consider the spatial field estimation when measurements are taken at multiple time instants, or *sequentially*. The physical process model in Section III-A together with the data model in Section III-B give rise to the evolution, for $k \in \mathbb{Z}_{>0}$

$$Y_i(k) = Z(p_i(k), k) + \epsilon_i, \tag{8a}$$

$$Z(s, k) = \phi(s)^T \beta(k) + \nu(s, k), \tag{8b}$$

$$\beta(k) = \boldsymbol{H}(k)\beta(k-1) + \boldsymbol{J}(k)\boldsymbol{\eta}(k), \tag{8c}$$

where, for convenience, we have introduced the notation $\boldsymbol{H}(k) = \boldsymbol{J}(k)\boldsymbol{B}(k)$, $\boldsymbol{J}(k) = (\boldsymbol{\Phi}(k)^T \boldsymbol{\Phi}(k))^{-1} \boldsymbol{\Phi}(k)^T$, and

$$\boldsymbol{B}(k) = [b(p_1(k)), \dots, b(p_n(k))]^T \in \mathbb{R}^{n \times m},$$
$$\boldsymbol{\Phi}(k) = [\phi(p_1(k)), \dots, \phi(p_n(k))]^T \in \mathbb{R}^{n \times m},$$
$$\boldsymbol{\eta}(k) = (\eta(p_1(k), k), \dots, \eta(p_n(k), k))^T \in \mathbb{R}^n.$$

Notice that the matrices $\boldsymbol{H}$ and $\boldsymbol{J}$ driving the evolution of the parameter β change from one time instant to another only if agent positions change.

The natural Bayesian solution for making predictions about the spatial field at time $k \in \mathbb{Z}_{>0}$ is to use the conditional distribution of Z given the data up to time k and the parameter β, but marginalizing over the posterior distribution of β given the data up to time k. This viewpoint also allows us to integrate into the picture prior information on the distribution of β. Therefore, 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). The decomposition into the three steps will be conveniently used in Section VI to design a distributed implementation.

A. Sequential parameter estimation via Kalman filtering

With the model (8), the parameter β can be optimally predicted via a Kalman filter. Here, instead of considering the usual Kalman filter recursion equations, we use the equivalent information filter formulation, see for instance [21]. We also note that alternative forms of the information filter, like the Joseph form [22], can be computed in a distributed fashion along the same lines described later in Sections V and VI.

Assume β is initially distributed according to a multivariate normal distribution $\beta(0) \sim N_p(\xi, \Xi)$. Given $t, s \in \mathbb{R}_{\geq 0}$, let $\widehat{\beta}(t|s)$ denote the estimator of β at time t with data collected up to time s, and let $P(t|s)$ denote the associated mean-squared error. The usual Kalman filter equations are written in the variables $(\widehat{\beta}(k|k-1), P(k|k-1))$ and $(\widehat{\beta}(k|k), P(k|k))$. Instead, we define

$$\widehat{a}(t|s) = P(t|s)^{-1} \widehat{\beta}(t|s), \tag{9}$$

and write the information filter equations in the variables $(\widehat{a}(k|k-1), P(k|k-1)^{-1})$ and $(\widehat{a}(k|k), P(k|k)^{-1})$. Note that, initially,

$$\widehat{a}(0|0) = \Xi^{-1}\xi, \quad P(0|0)^{-1} = \Xi^{-1}.$$

The information filter equations have two steps. The first step corresponds to a prediction of the parameter at time $k \in \mathbb{Z}_{>0}$ given data up to time $k-1$, and the second step incorporates the measurements taken at time k into the picture.

Prediction: Using (8c), the one-step-ahead prediction at time $k \in \mathbb{Z}_{>0}$ with data collected up to time $k-1$ is

$$\begin{aligned} &\widehat{a}(k|k-1) \\ &\quad = P(k|k-1)^{-1} \boldsymbol{H}(k) P(k-1|k-1) \widehat{a}(k-1|k-1), \end{aligned} \tag{10a}$$

with information matrix

$$\begin{aligned} &P(k|k-1)^{-1} \\ &\quad = (\boldsymbol{H}(k) P(k-1|k-1) \boldsymbol{H}(k)^T + \boldsymbol{J}(k) \boldsymbol{Q}(k) \boldsymbol{J}(k)^T)^{-1}, \end{aligned} \tag{10b}$$

where $\boldsymbol{Q}(k) = \operatorname{Var}(\boldsymbol{\eta}(k)) = [K_\eta(p_i(k) - p_j(k))] \in \mathbb{R}^{n \times n}$.

Correction: Using (8a), the optimal prediction at time $k \in \mathbb{Z}_{>0}$ with data collected up to time k can be recursively expressed as

$$\widehat{a}(k|k) = \widehat{a}(k|k-1) + \boldsymbol{\Phi}(k)^T (\boldsymbol{\Sigma}(k) + \sigma I_n)^{-1} \boldsymbol{Y}(k), \tag{11a}$$

with information matrix

$$\begin{aligned} &P(k|k)^{-1} \\ &\quad = P(k|k-1)^{-1} + \boldsymbol{\Phi}(k)^T(\boldsymbol{\Sigma}(k)+\sigma I_n)^{-1}\boldsymbol{\Phi}(k), \end{aligned} \tag{11b}$$

where $\boldsymbol{Y}(k) = (Y_1(k), \dots, Y_n(k)) \in \mathbb{R}^n$ denotes the measurements taken by the network agents at time k,

$$\boldsymbol{\Sigma}(k) = [K_\nu(p_i(k) - p_j(k))] \in \mathbb{R}^{n\times n}$$

is the variance corresponding to the spatial field ν, and σI_n is the variance corresponding to the sensor errors $\epsilon_1, \dots, \epsilon_n$.

The information filter equations provide an iterative fashion of computing $\widehat{a}(t|t)$ and $P(t|t)^{-1}$ that is appropriate for a distributed implementation by the robotic network. We describe this in Section VI-A.

B. Sequential simple Kriging

For $k \in \mathbb{Z}_{>0}$, let $\boldsymbol{Y}_{\leq k} = (\boldsymbol{Y}(0), \dots, \boldsymbol{Y}(k))$ denote the data available up to time k. For $s \in \mathbb{R}^d$, let

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

The covariance structure of the spatial field (cf. Section III-A) has some important consequences. On the one hand, the ith components of $\boldsymbol{\gamma}(s,k)$ and $\boldsymbol{\nabla\gamma}(s,k)$ can only be nonvanishing if $\|s-p_i(k)\| \leq r$. More importantly, the decorrelation in time of the spatial field and the sensor errors imply that only the observations collected at exactly time k play a role in the construction of the conditional predictive distribution of Z and ∇Z with observations collected up to time k. This is formalized in the following result.

Lemma 4.1 (Sequential simple Kriging): For all $k \in \mathbb{Z}_{>0}$ and all $s \in \mathbb{R}^d$, one has, conditionally on the data collected up to time k and the parameter $\beta(k)$,

(i) the normal distribution $Z(s,k) \,|\, (\boldsymbol{Y}_{\leq k}, \beta(k))$ with mean

$$\phi(s)^T\beta(k) + \boldsymbol{\gamma}(s,k)^T\boldsymbol{\Sigma}(k)_\sigma^{-1}(\boldsymbol{Y}(k) - \boldsymbol{\Phi}(k)\beta(k)),$$

and variance

$$K(0) - \boldsymbol{\gamma}(s,k)^T\boldsymbol{\Sigma}(k)_\sigma^{-1}\boldsymbol{\gamma}(s,k),$$

(ii) the normal distribution $\nabla Z(s,k) \;|\, (\boldsymbol{Y}_{\leq k}, \beta(k))$ with mean

$$\nabla\phi(s)^T\beta(k) + \boldsymbol{\nabla\gamma}(s,k)^T\boldsymbol{\Sigma}(k)_\sigma^{-1}(\boldsymbol{Y}(k) - \boldsymbol{\Phi}(k)\beta(k)),$$

and variance

$$-\operatorname{H}(K)(0) - \boldsymbol{\nabla\gamma}(s,k)^T\boldsymbol{\Sigma}(k)_\sigma^{-1}\boldsymbol{\nabla\gamma}(s,k),$$

where, for brevity, we let $\boldsymbol{\Sigma}(k)_\sigma = \boldsymbol{\Sigma}(k) + \sigma I_n$.

Proof: Given the evolution equations (8) and the covariance structure for ν and ϵ detailed in Section III, we have that for $k \in \mathbb{Z}_{>0}$ and $s \in \mathbb{R}^d$, the conditional distribution of $Z(s,k)$ given $(\boldsymbol{Y}_{\leq k}, \beta(k))$ is

$$\begin{aligned} Z(s,k) \,|\, (\boldsymbol{Y}_{\leq k}, \beta(k)) &\sim N\big(\phi(s)^T\beta(k) \\ &+ \boldsymbol{\gamma}_{\leq k}(s)^T(\boldsymbol{\Sigma}_{\leq k} + \sigma I_{n(k+1)})^{-1}(\boldsymbol{Y}_{\leq k} - \boldsymbol{\Phi}_{\leq k}\beta(k)), \\ &K(0) - \boldsymbol{\gamma}_{\leq k}(s)^T(\boldsymbol{\Sigma}_{\leq k} + \sigma I_{n(k+1)})^{-1}\boldsymbol{\gamma}_{\leq k}(s)\big), \end{aligned}$$

where $\boldsymbol{\Sigma}_{\leq k} = \operatorname{diag}(\boldsymbol{\Sigma}(0), \dots, \boldsymbol{\Sigma}(k))$, $\boldsymbol{\Phi}_{\leq k}^T = \big(\boldsymbol{\Phi}(0), \dots, \boldsymbol{\Phi}(k)\big)^T$, and $\boldsymbol{\gamma}_{\leq k}(s)^T = \big(0, \dots, 0, \boldsymbol{\gamma}(s,k)^T\big)$. Then, fact (i) follows by noting that

$$\begin{aligned} &(\boldsymbol{\Sigma}_{\leq k} + \sigma I_{n(k+1)})^{-1} \\ &\qquad = \operatorname{diag}((\boldsymbol{\Sigma}(0) + \sigma I_n)^{-1}, \dots, (\boldsymbol{\Sigma}(k) + \sigma I_n)^{-1}). \end{aligned}$$

Fact (ii) can be established analogously using (2). ■

In the absence of measurement errors, Lemma 4.1(i) corresponds to the simple Kriging predictor and variance of the spatio-temporal field in Section II.

C. Sequential Bayesian universal Kriging

Finally, we construct the Bayesian universal Kriging predictor of the spatial field and of its gradient by putting together Section IV-A and Section IV-B. Specifically, at time $k \in \mathbb{Z}_{>0}$, for each $s \in \mathbb{R}^d$, the posterior predictive distributions of $Z(s,k)$ and $\nabla Z(s,k)$ given the data $\boldsymbol{Y}_{\leq k}$ are obtained by marginalizing the conditional distributions in Lemma 4.1 over the posterior distribution $\beta(k) \;|\boldsymbol{Y}_{\leq k} \sim N(\widehat{\beta}(k|k), P(k|k))$ obtained with the combination of the information filter equations in Section IV-A and equation (9). Accordingly, we obtain the following result.

Lemma 4.2 (Sequential Bayesian universal Kriging): For all $k \in \mathbb{Z}_{>0}$ and all $s \in \mathbb{R}^d$, one has, conditionally on the data collected up to time k,

(i) the normal distribution $Z(s,k) \,|\, \boldsymbol{Y}_{\leq k}$ with mean

$$\phi(s)^T\widehat{\beta}(k|k) + \boldsymbol{\gamma}(s,k)^T\boldsymbol{\Sigma}(k)_\sigma^{-1}(\boldsymbol{Y}(k) - \boldsymbol{\Phi}(k)\widehat{\beta}(k|k))$$

and variance

$$\begin{aligned} K(0) &- \boldsymbol{\gamma}(s,k)^T\boldsymbol{\Sigma}(k)_\sigma^{-1}\boldsymbol{\gamma}(s,k) \\ &+ \big(\phi(s) - \boldsymbol{\Phi}(k)^T\boldsymbol{\Sigma}(k)_\sigma^{-1}\boldsymbol{\gamma}(s,k)\big)^T P(k|k) \\ &\qquad \big(\phi(s) - \boldsymbol{\Phi}(k)^T\boldsymbol{\Sigma}(k)_\sigma^{-1}\boldsymbol{\gamma}(s,k)\big). \end{aligned}$$

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

$$\begin{aligned} &\nabla\phi(s)^T\widehat{\beta}(k|k) \\ &\qquad + \boldsymbol{\nabla\gamma}(s,k)^T\boldsymbol{\Sigma}(k)_\sigma^{-1}(\boldsymbol{Y}(k) - \boldsymbol{\Phi}(k)\widehat{\beta}(k|k), \end{aligned}$$

and variance

$$\begin{aligned} -\operatorname{H}(K)(0) &- \boldsymbol{\nabla\gamma}(s,k)^T\boldsymbol{\Sigma}(k)_\sigma^{-1}\boldsymbol{\nabla\gamma}(s,k) \\ &+ \big(\nabla\phi(s) - \boldsymbol{\Phi}(k)^T\boldsymbol{\Sigma}(k)_\sigma^{-1}\boldsymbol{\nabla\gamma}(s,k)\big)^T P(k|k) \\ &\qquad \big(\nabla\phi(s) - \boldsymbol{\Phi}(k)^T\boldsymbol{\Sigma}(k)_\sigma^{-1}\boldsymbol{\nabla\gamma}(s,k)\big). \end{aligned}$$

The normal distribution in Lemma 4.2(i) corresponds to the spatial estimation obtained from the Kriged Kalman filter proposed in [3]. The normal distribution in Lemma 4.2(ii) gives us information about the gradient of the spatial field. Our next objective is to design a distributed coordination algorithm that allows network agents to compute these quantities.

V. Distributed average weighted least squares

This section presents distributed algorithms to compute average weighted least squares estimates. The capability to compute such estimates will be instrumental in Section VI to synthesize a distributed implementation of the estimation procedure described in Section IV.

Given a network of n agents with interaction topology described by an undirected graph G, matrices $F \in \mathbb{R}^{n\times n}$ invertible and $M \in \mathbb{R}^{n\times m}$, and a vector $c \in \mathbb{R}^n$, we introduce here an algorithm to compute the quantity

$$\frac{1}{n} M^T F^{-1} c, \tag{16}$$

that is distributed over G. The idea is to combine a Jacobi iteration and a dynamic consensus algorithm into a single procedure that we term the Weighted Least Squares Algorithm. The reason behind this terminology is the following: consider a linear observation model (determined by M) of an unknown parameter, and let c represent measured data with associated covariance F. Then, (16) corresponds to the average weighted least squares estimate of the parameter. Let us start by presenting the individual ingredients of the Weighted Least Squares Algorithm.

A. Jacobi overrelaxation algorithm

Given an invertible matrix $F \in \mathbb{R}^{n\times n}$ and a vector $c \in \mathbb{R}^n$, consider the linear system $Fy = c$. The Jacobi overrelaxation (JOR) algorithm [17] is an iterative procedure to compute the unique solution $y = F^{-1}c \in \mathbb{R}^n$. It is formulated as the discrete-time dynamical system

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

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 in terms of the eigenvalues of the matrix describing the linear iteration, see [17]. Here, instead, we will use the following sufficient convergence criteria from [23, Theorem 2].

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

As long as (i) agent i has access to c_i, and (ii) if $f_{ij} \neq 0$, 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 $y = F^{-1}c$ with information provided by its neighbors in G.

Remark 5.2: (Robustness to agents' arrivals and departures): In the scenario where the matrix F, the vector c, and the interaction topology G are a function of agents' positions, the fact that the JOR algorithm converges from any initial condition implies that it is robust to a finite number of agents' arrivals and departures. In other words, if the number of agents after addition and deletion is $\tilde{n}$, with corresponding $\tilde{F}$ and $\tilde{c}$, then as long as $h < 2/\tilde{n}$, the convergence of the JOR algorithm to $\tilde{F}^{-1}\tilde{c}$ is guaranteed. •

B. Dynamic average consensus algorithms

Dynamic average consensus filters [12], [13], [14] are distributed algorithms that allow the network to track the average of a given time-varying signal. Under suitable conditions on the evolution of the signal, one can guarantee asymptotic convergence. Here, we use a particular instance of the proportional-integral dynamic consensus estimators studied in [14] but formulated for higher-dimensional signals.

Let $\tau \in \mathbb{R}_{\geq 0} \mapsto u(\tau) \in (\mathbb{R}^m)^n$ 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

$$\frac{dv_i}{d\tau} = \gamma(u_i(\tau) - v_i(\tau)) - \sum_{j\neq i} a_{ij}(v_i(\tau) - v_j(\tau)) + \sum_{j\neq i} a_{ij}(w_i(\tau) - w_j(\tau)), \tag{17a}$$

$$\frac{dw_i}{d\tau} = -\sum_{j\neq i} a_{ij}(v_i(\tau) - v_j(\tau)), \tag{17b}$$

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 neighboring agents in the graph G. As we see next, v_i can be interpreted as the estimate that agent i possess of the average of the time-varying signal u. It can be proved [14] that if G is connected, for any $\gamma > 0$, any constant signal $\tau \in \mathbb{R}_{\geq 0} \mapsto u(\tau) = u \in (\mathbb{R}^m)^n$, and any initial states $v(0), w(0) \in (\mathbb{R}^m)^n$, the algorithm (17) satisfies

$$v_i(\tau) - \frac{1}{n}\sum_{i=1}^{n} u_i(\tau) \to 0 \quad \text{as } \tau \to +\infty \tag{18}$$

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

Remark 5.3: (Robustness to agents' arrivals and departures): When the signal u and the interaction topology G are a function of agents' positions, one can show [14] that the estimator (17) guarantees zero steady-state error under a finite number of agents' arrivals and departures. •

C. The Weighted Least Squares Algorithm

Here, we combine the JOR algorithm and the dynamic average consensus algorithm to synthesize the Weighted Least Squares Algorithm described in Table I.

Proposition 5.4: Consider the Weighted Least Squares Algorithm described in Table I. For $F \in \mathbb{R}^{n\times n}$ invertible, $c \in \mathbb{R}^n$, and $M \in \mathbb{R}^{n\times m}$, define the output functions $\texttt{WLS}(F,c,M) : \mathbb{R}_{\geq 0} \to (\mathbb{R}^m)^n$ and $\texttt{JOR}(F,c) : \mathbb{R}_{\geq 0} \to \mathbb{R}^n$ by, respectively,

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

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 $i \in \{1,\dots,n\}$

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

```
Initialization:
1: $y(0) = c \in \mathbb{R}^n$
2: $v(0) = w(0) = (\text{row}_1(M)z_1, \dots, \text{row}_n(M)z_n) \in (\mathbb{R}^m)^n$,
   with $z \in \mathbb{R}^n$ arbitrary
3: $\gamma \in \mathbb{R}_{>0}$ and $h \in (0, 2/n)$

Agent $i \in \{1,\dots,n\}$ executes concurrently
1: Jacobi overrelaxation algorithm, for $\ell \in \mathbb{Z}_{\geq 0}$
     $y_i(\ell+1) = (1-h)y_i(\ell) - h\frac{1}{f_{ii}}\Big(\sum_{j\neq i} f_{ij}y_j(\ell) - c_i\Big)$
2: Dynamic average consensus algorithm, for $\tau \in \mathbb{R}_{\geq 0}$
     $\frac{dv_i}{d\tau} = \gamma(u_i(\tau) - v_i(\tau)) - \sum_{j\neq i} a_{ij}(v_i(\tau) - v_j(\tau))$
              $+ \sum_{j\neq i} a_{ij}(w_i(\tau) - w_j(\tau)),$
     $\frac{dw_i}{d\tau} = -\sum_{j\neq i} a_{ij}(v_i(\tau) - v_j(\tau)),$
   where $\mathcal{A} = (a_{ij})$ is the adjacency matrix of $G$, and
   $\tau \mapsto u(\tau) \in (\mathbb{R}^m)^n$ 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.

can compute $\texttt{WLS}_i(F,c,M)$ and $\texttt{JOR}_i(F,c)$ with information provided by its neighboring agents in G;

(ii) the function $\texttt{JOR}(F,c)$ is such that

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

exponentially fast;

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

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

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

Proof: The WEIGHTED LEAST SQUARES ALGORITHM is distributed over G by design. The statement on the limit of $\texttt{JOR}(F,c)(\tau)$ follows from Lemma 5.1 (the linear rate of convergence of the discrete-time algorithm translates into an exponential rate of convergence for the continuous-time function). Regarding the limit of $\texttt{WLS}_i(F,c,M)$, $i \in \{1,\dots,n\}$, note that, with the notation of Table I,

$$\sum_{i=1}^n u_i(\tau) = \sum_{i=1}^n \text{row}_i(M)y_i(\lfloor\tau\rfloor) \to \sum_{i=1}^n \text{row}_i(M)y_i = \boldsymbol{M}^T F^{-1}c,$$

where we have used that the JOR algorithm converges to the solution $y = (y_1,\dots,y_n)$ of $Fy = c$. The result now follows from the convergence properties (18) of the dynamic average consensus algorithm. ■

Remark 5.5: (Execution of JOR, dynamic average consensus, and WEIGHTED LEAST SQUARES *algorithms)*: In the forthcoming discussion, we will make use of the fast convergence properties of the JOR, dynamic average consensus, and WEIGHTED LEAST SQUARES algorithms and use the exact asymptotic limit of these algorithms in our derivations. In practice, after a few iterations, the values obtained by the execution of these algorithms are very close to the exact asymptotic limit because the convergence rate of the JOR algorithm is linear and the convergence rates of the dynamic average consensus algorithms and the WEIGHTED LEAST SQUARES ALGORITHM are exponential. Using this fact, it is not difficult to characterize the precise number of iterations needed to achieve a desired level of convergence. •

VI. DISTRIBUTED IMPLEMENTATION OF SEQUENTIAL FIELD ESTIMATION

In this section we introduce the DISTRIBUTED KRIGED KALMAN FILTER. This algorithm allows each network agent to compute, at any time, the posterior predictive distribution of the spatial field and its gradient on a neighborhood of its current location obtained in Section IV, and only requires communication with neighboring agents in $\mathcal{G}_{R\text{-disk}}$.

The algorithm is described in Table II. 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 algorithms presented in Section V, and in particular on the WEIGHTED LEAST SQUARES ALGORITHM. It should be noted that every time the algorithms in Section V are invoked in Table II, we use their exact asymptotic limit, cf. Remarks 5.5 and 6.2.

In the following, we explain in detail the algorithm steps outlined in Table II.

A. Distributed sequential parameter estimation

Here, we describe the strategy that network agents implement in order to compute the sequential parameter estimation described in Section IV-A.

Distributed prediction: At time $k \in \mathbb{Z}_{>0}$, assume $\widehat{a}(k-1|k-1)$ and $P(k-1|k-1)^{-1}$ are known to all network agents from the previous iteration (initially, we set $\widehat{a}(0|0) = \Xi^{-1}\xi$,

$P(0|0)^{-1} = \Xi^{-1}$). According to (10), agent i can compute the one-step-ahead prediction $\widehat{a}(k|k-1)$ with information matrix $P(k|k-1)^{-1}$ (step `9:` of Table II) if it has access to the matrices

$$\begin{aligned}&\boldsymbol{H}(k) = (\boldsymbol{\Phi}(k)^T\boldsymbol{\Phi}(k))^{-1}\boldsymbol{\Phi}(k)^T B(k),\\ &\boldsymbol{J}(k)\boldsymbol{Q}(k)\boldsymbol{J}(k)^T\\ &\quad = (\boldsymbol{\Phi}(k)^T\boldsymbol{\Phi}(k))^{-1}\boldsymbol{\Phi}(k)^T\boldsymbol{Q}(k)\boldsymbol{\Phi}(k)(\boldsymbol{\Phi}(k)^T\boldsymbol{\Phi}(k))^{-1}.\end{aligned}$$

We break down this task into the computation of the matrices

$$\boldsymbol{\Phi}(k)^T\boldsymbol{\Phi}(k),\ \boldsymbol{\Phi}(k)^T B(k),\ \boldsymbol{\Phi}(k)^T\boldsymbol{Q}(k)\boldsymbol{\Phi}(k) \in \mathbb{R}^{p\times m}. \quad (19)$$

Let us show how the network performs a distributed computation of the matrices in (19). This corresponds to step `3:` of Table II. Specifically, if agent i has access to the matrix $C_i \in \mathbb{R}^{m\times m}$ for $i \in \{1,\dots,n\}$, define $\mathtt{DAC}(C_1,\dots,C_n) : \mathbb{R}_{\geq 0} \to \mathbb{R}^{m\times m}$ by

$$\mathtt{DAC}(C_1,\dots,C_n)(\tau) = v(\tau), \quad (20)$$

where v is determined by the execution of the dynamic average consensus algorithm (17) over $\mathcal{G}_{R\text{-disk}}$ with constant signal $\tau \mapsto u(\tau) \in (\mathbb{R}^{m\times m})^n$ given by

$$u_i(\tau) = C_i.$$

According to Section V-B, if $\mathcal{G}_{R\text{-disk}}$ is connected, then

$$\mathtt{DAC}(C_1,\dots,C_n)(\tau) \to \frac{1}{n}\sum_{i=1}^{n} C_i,$$

as $\tau \to \infty$ exponentially fast.

At time $k \in \mathbb{Z}_{>0}$, agent $i \in \{1,\dots,n\}$ has access to

$$\phi(p_i(k))^T\phi(p_i(k)) \quad \text{and} \quad \phi(p_i(k))^T b(p_i(k)).$$

Moreover, the assumption (5) on the finite correlation range of η guarantees that agent $i \in \{1,\dots,n\}$ can compute $\operatorname{row}_i(\boldsymbol{Q}(k)) = (K_\eta(p_i(k)-p_1(k)),\dots,K_\eta(p_i(k)-p_n(k))) \in \mathbb{R}^n$ by knowing the position of its neighbors in $\mathcal{G}_{R\text{-disk}}$ at time k. Hence, agent i has access to

$$\phi(p_i(k))^T\operatorname{row}_i(\boldsymbol{Q}(k)\boldsymbol{\Phi}(k)),$$

by communicating with its R-disk neighbors. Therefore, in order to compute the matrices in (19), the network executes three dynamic average consensus algorithms over $\mathcal{G}_{R\text{-disk}}$ to obtain

$$\begin{aligned}&\mathtt{DAC}_i(\phi(p_1(k))^T\phi(p_1(k)),\dots,\phi(p_n(k))^T\phi(p_n(k)))(\tau) \to\\ &\quad \frac{1}{n}\boldsymbol{\Phi}(k)^T\boldsymbol{\Phi}(k),\\ &\mathtt{DAC}_i(\phi(p_1(k))^T b(p_1(k)),\dots,\phi(p_n(k))^T b(p_n(k)))(\tau) \to\\ &\quad \frac{1}{n}\boldsymbol{\Phi}(k)^T B(k),\\ &\mathtt{DAC}_i(\phi(p_1(k))^T\operatorname{row}_1(\boldsymbol{Q}(k)\boldsymbol{\Phi}(k)),\dots,\\ &\qquad \phi(p_n(k))^T\operatorname{row}_n(\boldsymbol{Q}(k)\boldsymbol{\Phi}(k)))(\tau) \to \frac{1}{n}\boldsymbol{\Phi}(k)^T\boldsymbol{Q}(k)\boldsymbol{\Phi}(k).\end{aligned}$$

Distributed correction: At time $k \in \mathbb{Z}_{>0}$, assume $\widehat{a}(k|k-1)$ and $P(k|k-1)^{-1}$ are known to all network agents from the distributed prediction computation. According to (11), agent i can compute the prediction $\widehat{a}(k|k)$ with information matrix $P(k|k)^{-1}$ (step `10:` of Table II) if it has access to

$$\boldsymbol{\Phi}(k)^T\boldsymbol{\Sigma}(k)_\sigma^{-1}\boldsymbol{Y}(k) \in \mathbb{R}^m, \quad (21a)$$

$$\boldsymbol{\Phi}(k)^T\boldsymbol{\Sigma}(k)_\sigma^{-1}\boldsymbol{\Phi}(k) \in \mathbb{R}^{m\times m}. \quad (21b)$$

Let us describe how the network performs a distributed computation of (21).

1) Distributed computation of the weighted least squares estimate: This discussion refers to step `4:` of Table II. The network computes the vector (21a) by invoking the WEIGHTED LEAST SQUARES ALGORITHM once. Specifically, agent i has access to $\operatorname{row}_i(\boldsymbol{\Sigma}(k)_\sigma)$, $\boldsymbol{Y}(k)_i = Y_i(k)$ and to $\operatorname{row}_i(\boldsymbol{\Phi}(k)) = \phi(p_i(k))^T$. Therefore, assuming that $\mathcal{G}_{R\text{-disk}}(p_1(k),\dots,p_n(k))$ is connected, the execution of the WEIGHTED LEAST SQUARES ALGORITHM with $F = \boldsymbol{\Sigma}(k)_\sigma$, $c = \boldsymbol{Y}(k)$, and $M = \boldsymbol{\Phi}(k)$ guarantees, according to Proposition 5.4,

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

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

2) Distributed computation of the covariance matrix of the weighted least squares estimate: This discussion refers to steps `5:-8:` of Table II. The network computes the matrix (21b) by invoking p instances of the WEIGHTED LEAST SQUARES ALGORITHM (one instance per matrix column). Specifically, for each $j \in \{1,\dots,p\}$, agent i has access to the ith component of the vector $\operatorname{col}_j(\boldsymbol{\Phi}(k)) \in \mathbb{R}^n$, and to $\operatorname{row}_i(\boldsymbol{\Phi}(k)) = \phi(p_i(k))^T$. Moreover, the assumption (5) on the finite correlation range of the spatial field ν guarantees that agent i can compute $\operatorname{row}_i(\boldsymbol{\Sigma}(k)_\sigma)$ by knowing the position of its neighbors in $\mathcal{G}_{R\text{-disk}}$ at time k. Therefore, assuming that $\mathcal{G}_{R\text{-disk}}(p_1(k),\dots,p_n(k))$ is connected, the execution of the WEIGHTED LEAST SQUARES ALGORITHM with $F = \boldsymbol{\Sigma}(k)_\sigma$, $c = \operatorname{col}_j(\boldsymbol{\Phi}(k))$, and $M = \boldsymbol{\Phi}(k)$ guarantees, according to Proposition 5.4,

$$\begin{aligned}&\mathtt{WLS}_i(\boldsymbol{\Sigma}(k)_\sigma, \operatorname{col}_j(\boldsymbol{\Phi}(k)), \boldsymbol{\Phi}(k))(\tau) \longrightarrow\\ &\qquad\qquad \frac{1}{n}\boldsymbol{\Phi}(k)^T\boldsymbol{\Sigma}(k)_\sigma^{-1}\operatorname{col}_j(\boldsymbol{\Phi}(k)),\end{aligned}$$

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 $\mathtt{WLS}_i(\boldsymbol{\Sigma}(k)_\sigma, \boldsymbol{\Phi}(k), \boldsymbol{\Phi}(k))$ defined by

$$\begin{aligned}&\mathtt{WLS}_i(\boldsymbol{\Sigma}(k)_\sigma, \boldsymbol{\Phi}(k), \boldsymbol{\Phi}(k))(\tau) =\\ &\qquad \big(\mathtt{WLS}_i(\boldsymbol{\Sigma}(k)_\sigma, \operatorname{col}_1(\boldsymbol{\Phi}(k)), \boldsymbol{\Phi}(k)),\dots,\\ &\qquad\qquad \dots,\mathtt{WLS}_i(\boldsymbol{\Sigma}(k)_\sigma, \operatorname{col}_p(\boldsymbol{\Phi}(k)), \boldsymbol{\Phi}(k))\big)(\tau),\end{aligned}$$

with the property that

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

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

```
Name:     DISTRIBUTED KRIGED KALMAN FILTER
Goal:     Compute Bayesian universal kriging predictors of the spatial field and of its gradient
Assumes:  (i) $\mathcal{G}_{R\text{-disk}}$ is connected along evolution
          (ii) Initially all agents know $\beta \sim N(\xi, \Xi)$

Initialization:
  $\widehat{a}(0|0) = \Xi^{-1}\xi$, $P(0|0)^{-1} = \Xi^{-1}$
  For $t \in [0,1)$ and $s \in \mathbb{R}^d$, each agent evaluates
      $E(Z(s,t) \mid \boldsymbol{Y}_{\leq 0}) = \phi(s)^T \xi$ and $E(\nabla Z(s,t) \mid \boldsymbol{Y}_{\leq 0}) = \nabla\phi(s)^T \xi$.

At time $k \in \mathbb{Z}_{>0}$, agent $i \in \{1, \dots, n\}$
  takes measurement $\boldsymbol{Y}_i(k) = Y_i(k)$, computes $\text{row}_i(\boldsymbol{\Phi}(k)) = \phi(p_i(k))^T$ and $\text{row}_i(B(k)) = b(p_i(k))^T$
  acquires location of $\mathcal{G}_{R\text{-disk}}(p_1(k), \dots, p_n(k))$-neighbors and computes $\text{row}_i(\boldsymbol{\Sigma}(k)_\sigma)$ and $\text{row}_i(\boldsymbol{Q}(k))$
  executes three dynamic consensus algorithms DAC over $\mathcal{G}_{R\text{-disk}}(p_1(k), \dots, p_n(k))$ and sets
      $\boldsymbol{\Phi}(k)^T \boldsymbol{\Phi}(k) = n\,\mathtt{DAC}_i(\phi(p_1(k))^T \phi(p_1(k)), \dots, \phi(p_n(k))^T \phi(p_n(k)))(\infty)$,
      $\boldsymbol{\Phi}(k)^T B(k) = n\,\mathtt{DAC}_i(\phi(p_1(k))^T b(p_1(k)), \dots, \phi(p_n(k))^T b(p_n(k)))(\infty)$,
      $\boldsymbol{\Phi}(k)^T \boldsymbol{Q}(k) \boldsymbol{\Phi}(k) = n\,\mathtt{DAC}_i(\phi(p_1(k))^T \text{row}_1(\boldsymbol{Q}(k)\boldsymbol{\Phi}(k)), \dots, \phi(p_n(k))^T \text{row}_n(\boldsymbol{Q}(k)\boldsymbol{\Phi}(k)))(\infty)$,
      $\boldsymbol{H}(k) = (\boldsymbol{\Phi}(k)^T \boldsymbol{\Phi}(k))^{-1} \boldsymbol{\Phi}(k)^T B(k)$,
      $\boldsymbol{J}(k) \boldsymbol{Q}(k) \boldsymbol{J}(k)^T = (\boldsymbol{\Phi}(k)^T \boldsymbol{\Phi}(k))^{-1} \boldsymbol{\Phi}(k)^T \boldsymbol{Q}(k) \boldsymbol{\Phi}(k) (\boldsymbol{\Phi}(k)^T \boldsymbol{\Phi}(k))^{-1}$.
  executes WEIGHTED LEAST SQUARES ALGORITHM over $\mathcal{G}_{R\text{-disk}}(p_1(k), \dots, p_n(k))$ for $(\boldsymbol{\Sigma}(k)_\sigma, \boldsymbol{Y}(k), \boldsymbol{\Phi}(k))$ and sets
      $\boldsymbol{\Phi}(k)^T \boldsymbol{\Sigma}(k)_\sigma^{-1} \boldsymbol{Y}(k) = n\,\mathtt{WLS}_i(\boldsymbol{\Sigma}(k)_\sigma, \boldsymbol{Y}(k), \boldsymbol{\Phi}(k))(\infty)$
  for $j = 1$ to $p$ do
    executes WEIGHTED LEAST SQUARES ALGORITHM over $\mathcal{G}_{R\text{-disk}}(p_1(k), \dots, p_n(k))$ for $(\boldsymbol{\Sigma}(k)_\sigma, \text{col}_j(\boldsymbol{\Phi}(k)), \boldsymbol{\Phi}(k))$
  end for
  sets $\boldsymbol{\Phi}(k)^T \boldsymbol{\Sigma}(k)_\sigma^{-1} \boldsymbol{\Phi}(k) = n\,\mathtt{WLS}_i(\boldsymbol{\Sigma}(k)_\sigma, \boldsymbol{\Phi}(k), \boldsymbol{\Phi}(k))(\infty)$
  Parameter estimation: prediction
  computes estimate and information matrix at time $k$ with data up to time $k-1$
      $P(k|k-1)^{-1} = (\boldsymbol{H}(k) P(k-1|k-1) \boldsymbol{H}(k)^T + \boldsymbol{J}(k) \boldsymbol{Q}(k) \boldsymbol{J}(k)^T)^{-1}$,
      $\widehat{a}(k|k-1) = P(k|k-1)^{-1} \boldsymbol{H}(k) P(k-1|k-1) \widehat{a}(k-1|k-1)$.
  Parameter estimation: correction
  computes estimate and information matrix at time $k$ with data up to time $k$
      $P(k|k)^{-1} = P(k|k-1)^{-1} - \boldsymbol{\Phi}(k)^T (\boldsymbol{\Sigma}(k)_\sigma)^{-1} \boldsymbol{\Phi}(k)$,
      $\widehat{a}(k|k) = \widehat{a}(k|k-1) + \boldsymbol{\Phi}(k)^T (\boldsymbol{\Sigma}(k)_\sigma)^{-1} \boldsymbol{Y}(k)$.
  Simple kriging
  sets
      $(\boldsymbol{\Sigma}(k)_\sigma^{-1} \boldsymbol{Y}(k))_i := \mathtt{JOR}_i(\boldsymbol{\Sigma}(k)_\sigma, \boldsymbol{Y}(k))(\infty)$,
      $\text{row}_i(\boldsymbol{\Sigma}(k)_\sigma^{-1} \boldsymbol{\Phi}(k)) := \text{row}_i(\mathtt{JOR}(\boldsymbol{\Sigma}(k)_\sigma, \boldsymbol{\Phi}(k))(\infty))$.
  computes predictors at $s \in B(p_i(k), R-r)$
      $E(Z(s,k) \mid (\boldsymbol{Y}_{\leq k}, \beta(k))) = \phi(s)^T \beta(k) + \sum_{\|s-p_j(k)\| \leq r} \boldsymbol{\gamma}_j(s,k) \big( (\boldsymbol{\Sigma}(k)_\sigma^{-1} \boldsymbol{Y}(k))_j - \text{row}_j(\boldsymbol{\Sigma}(k)_\sigma^{-1} \boldsymbol{\Phi}(k)) \beta(k) \big)$,
      $E(\nabla Z(s,k) \mid (\boldsymbol{Y}_{\leq k}, \beta(k))) = \nabla\phi(s)^T \beta(k) + \sum_{\|s-p_j(k)\| \leq r} \boldsymbol{\nabla\gamma}_j(s,k) \big( (\boldsymbol{\Sigma}(k)_\sigma^{-1} \boldsymbol{Y}(k))_j - \text{row}_j(\boldsymbol{\Sigma}(k)_\sigma^{-1} \boldsymbol{\Phi}(k)) \beta(k) \big)$.
  Bayesian universal kriging
  for $t \in [k, k+1)$ and $s \in B(p_i(k), R-r)$, combines parameter estimation and simple kriging to compute (cf. Section IV-C)
      $E(Z(s,t) \mid \boldsymbol{Y}_{\leq k})$ and $E(\nabla Z(s,t) \mid \boldsymbol{Y}_{\leq k})$.
```

TABLE II
THE DISTRIBUTED KRIGED KALMAN FILTER

B. Distributed sequential simple Kriging

Here, we describe the strategy that network agents implement in order to compute the sequential simple Kriging described in Section IV-B. This strategy makes use of the special covariance structure of the spatial field. The discussion refers to steps `11:-12:` of Table II. According to Lemma 4.1, to compute the means of the conditional distributions of the spatial field and its gradient at $s \in \mathbb{R}^d$ and $k \in \mathbb{Z}_{>0}$, we look for the distributed calculation of

$$\boldsymbol{\gamma}(s,k) \in \mathbb{R}^n, \ \boldsymbol{\nabla\gamma}(s,k) \in \mathbb{R}^{n\times d}, \tag{22a}$$

$$\boldsymbol{\Sigma}(k)_\sigma^{-1}\boldsymbol{Y}(k) \in \mathbb{R}^n, \tag{22b}$$

$$\boldsymbol{\Sigma}(k)_\sigma^{-1}\boldsymbol{\Phi}(k) \in \mathbb{R}^{n\times m}. \tag{22c}$$

Regarding (22a), note that 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 the ball centered at s of radius r is contained in the area within communication range of agent i – since in this case the agent will have access to the location of all other agents contained in $B(s,r)$. Noting that

$$B(s,r) \subset B(p_i(k),R) \quad \text{iff} \quad s \in B(p_i(k), R-r),$$

we deduce that agent i can compute (22a) in a distributed way for any $s \in B(p_i(k), R-r)$.

Regarding (22b) and (22c), note that, as a by-product of the executions of the WEIGHTED LEAST SQUARES ALGORITHM performed in the distributed sequential parameter estimation of Section VI-A, at time $k \in \mathbb{Z}_{>0}$, agent $i \in \{1,\dots,n\}$ has available the ith component of the functions

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

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

$$\begin{aligned}\texttt{JOR}(\boldsymbol{\Sigma}(k)_\sigma, \boldsymbol{\Phi}(k))(\tau) = \big(&\texttt{JOR}(\boldsymbol{\Sigma}(k)_\sigma, \mathrm{col}_1(\boldsymbol{\Phi}(k))),\\ &\dots, \texttt{JOR}(\boldsymbol{\Sigma}(k)_\sigma, \mathrm{col}_p(\boldsymbol{\Phi}(k)))\big)(\tau).\end{aligned}$$

Note that agent i has access to the ith row of $\texttt{JOR}(\boldsymbol{\Sigma}(k)_\sigma, \boldsymbol{\Phi}(k))$. By Proposition 5.4, we have

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

Finally, note that agent $i \in \{1,\dots,n\}$ has access to both $\texttt{JOR}_j(\boldsymbol{\Sigma}(k)_\sigma, \boldsymbol{Y}(k)) \in \mathbb{R}$ and $\mathrm{row}_j(\texttt{JOR}(\boldsymbol{\Sigma}(k)_\sigma, \boldsymbol{\Phi}(k))) \in \mathbb{R}^m$ 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 6.1: For all $k \in \mathbb{Z}_{>0}$ 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 `12:` of Table II, and therefore, are computable by agent i over $\mathcal{G}_{R\text{-disk}}$.

Proof: If $s \in B(p_i(k), R-r)$, then agent i can compute all the non-vanishing components of $\boldsymbol{\gamma}(s,k)$ and $\boldsymbol{\nabla\gamma}(s,k)$, since they correspond to the identifiers of other agents that must be its neighbors in $\mathcal{G}_{R\text{-disk}}$. Since agent i has also access to the corresponding components of $\texttt{JOR}_j(\boldsymbol{\Sigma}(k)_\sigma, \boldsymbol{Y}(k)) \in \mathbb{R}$ and $\mathrm{row}_j(\texttt{JOR}(\boldsymbol{\Sigma}(k)_\sigma, \boldsymbol{\Phi}(k))) \in \mathbb{R}^m$, the result follows. ■

Remark 6.2: (Execution of DISTRIBUTED KRIGED KALMAN FILTER*)*: 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 the robotic sensor network model in Section III-B, measurements are only taken at instants of time in $\mathbb{Z}_{>0}$. These considerations, together with the observations made in Remark 5.5, lead us to assume that the distributed computations described in Sections IV-A and IV-B run on a time scale τ which is much faster than the time scale t. These observations provide justification for the asymptotic limits taken in steps `3:-8:` and `11:` of Table II. We are currently addressing the characterization of the communication requirements for the algorithm execution. However, it should be noted that, with regards to other message-passing algorithms, the present approach needs minimal memory requirements at each agent, provides all agents with the same global information, and handles without any modification evolving network interaction topologies. •

Remark 6.3: (Robustness to agents' arrivals and departures): The requirement in DISTRIBUTED KRIGED KALMAN FILTER that $\mathcal{G}_{R\text{-disk}}$ is connected along the evolution can be relaxed as follows. From Remarks 5.2 and 5.3, it is clear that both the dynamic average consensus algorithms and the WEIGHTED LEAST SQUARES ALGORITHM are robust to changing numbers of network agents. As long as each agent knows the exact number of agents in its connected component of $\mathcal{G}_{R\text{-disk}}$, it can perform the distributed data fusion steps described in Table II. Regarding the parameter estimation, different connected components will use different measurements, and hence will have different mean and covariance estimates about the parameter. As currently stated, the algorithm is then robust to agent deletion, while it is not robust to the addition of new agents that can join the connected component with possibly different parameter mean and covariance estimates. Regarding simple Kriging, since the spatial correlation range of the random field is smaller than the communication radius, each agent computes the same estimate of the spatial field and of its gradient on a neighborhood around its current location. Hence, at this stage, the algorithm is robust to agents' arrivals and departures. •

VII. DISTRIBUTED GRADIENT ASCENT OF SPATIAL FIELDS

The distributed estimation algorithm developed in the previous section can be used in conjunction with the motion capabilities of the robotic agents to perform a number of coordination tasks. In this section, we illustrate these possibilities by designing a distributed gradient ascent coordination algorithm to find the maxima of a spatial field.

At any instant of time $t \in \mathbb{R}_{\geq 0}$, the DISTRIBUTED KRIGED KALMAN FILTER described in Table II allows agent $i \in \{1,\dots,n\}$ to compute the expected value of both the spatial field and its gradient in the neighborhood $B(p_i(\lfloor t \rfloor), R-r)$ of its location. With the information provided by the filter, each agent can then implement a gradient ascent strategy of the form

$$\dot{p}_i(t) = E(\nabla Z(p_i(t), t) \mid \boldsymbol{Y}_{\leq t}). \tag{23}$$

Note that, because new measurements are taken at time instants in $\mathbb{Z}_{>0}$, the resulting trajectory of agent i is continuous and piecewise differentiable. The next result characterizes the asymptotic convergence properties of the distributed gradient ascent strategy when no measurement errors are present.

Proposition 7.1: Let Z be a spatial Gaussian random field with continuously differentiable mean function and with compact superlevel sets. Consider a robotic sensor network that measures Z with no error, that is, $\epsilon_i = 0$ for $i \in \{1, \dots, n\}$ in equation (7). Then, any network trajectory evolving under (23) that starts from $\mathcal{S} = (\mathbb{R}^d)^n \setminus \{(p_1, \dots, p_n) \in (\mathbb{R}^d)^n \mid p_i = p_j \text{ with } i \neq j\}$ satisfies

$$E(\nabla Z(p_i(t)) \mid \boldsymbol{Z}_{\leq t}) \to 0, \quad i \in \{1, \dots, n\},$$

as $t \to \infty$, with probability one.

Proof: Let $(p_1(0), \dots, p_n(0)) \in \mathcal{S}$. Note that $\mathcal{S}$ is invariant. This is a consequence of the fact that, for any given set of measurements of the spatial field $\boldsymbol{Y}_{\leq t} = \boldsymbol{Z}_{\leq t}$, the vector field in (23) is continuously differentiable, and hence no two trajectories intersect. Let $\mathcal{D}_i = \{p \in \mathbb{R}^d \mid Z(p) \geq Z(p_i(0))\}$, and define

$$\mathcal{D} = \mathcal{D}_1 \times \cdots \times \mathcal{D}_n.$$

By hypothesis, $\mathcal{D}$ is compact. For each $k \in \mathbb{Z}_{\geq 0}$, agent $i \in \{1, \dots, n\}$ is guaranteed to increase the expected value of the spatial field along the time interval $[k, k+1]$ by following the gradient flow (23). Equivalently,

$$E(Z(p_i(k+1)) \mid \boldsymbol{Z}_{\leq k}) \geq E(Z(p_i(k)) \mid \boldsymbol{Z}_{\leq k}).$$

Because by hypothesis there are no measurement errors, we have $E(Z(p_i(k)) \mid \boldsymbol{Z}_{\leq k}) = Z(p_i(k))$. Therefore, the sequence $\{V_k = V(p_1(k), \dots, p_n(k))\}_{k=0}^{\infty}$, where $V(p_1, \dots, p_n) = \sum_{i=1}^{n} Z(p_i)$ is a submartingale [18], that is,

$$E(V_{k+1} \mid \boldsymbol{Z}_{\leq k}) \geq V_k.$$

Using now [24, Corollary 2], we conclude the result. ■

We have implemented the gradient ascent (23) in Mathematica® to illustrate its performance. The Distributed Kriged Kalman Filter is implemented as a single centralized program. Agents evolve according to the robotic network model described in Section III-B, with communication radius $R = 2.5$, agent control authority bounded by $u_{\max} = .25$, and noise sensor error variance $\sigma = .25$. During the execution, each agent makes use of the expected value of the gradient of the spatial field computed in step 13: of Distributed Kriged Kalman Filter to follow the gradient ascent direction as specified in (23). We illustrate the performance of the closed-loop system in Figure 1 with a static (i.e., not evolving in time) spatial field with mean $\mu(s) = .3 + 1.2\,\mathrm{e}^{-\|s-(.5,1)\|^2} + \mathrm{e}^{-\|s+(1.5,1.5)\|^2}$, zero-mean small-scale variability ν, and covariance structure determined by $K_\nu(s) = e^{-5\|s\|^2}$ if $\|s\| \leq r = 1.5$ and $K_\nu(s) = 0$ otherwise. In the simulation, agents initially know $\beta \sim N_3(0, I_3)$.

According to Proposition 7.1, individual agents converge asymptotically to the set of expected critical points of the spatial field. However, if two or more agents tend to the same point in $\mathbb{R}^d$, then the numerical implementation becomes problematic because the Bayesian universal Kriging computations are, in general, ill-posed on configurations in $\mathcal{S}$, where two or more points coincide. Our simulation did not exhibit this problem because we did not run it for a time long enough. One way to resolve this is by specifying a threshold in how close the individual agents need to get to the set of critical points. Once agents that are converging to the same critical point satisfy the threshold, they can decide, for instance, to form a circle and uniformly deploy around it.

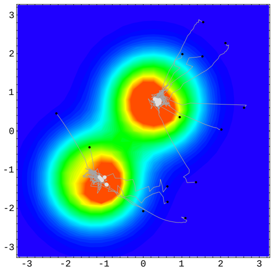

Fig. 1. Distributed gradient ascent cooperative strategy (23) implemented by a robotic sensor network consisting of 14 agents. We depict the contour plot of the posterior mean of the spatial field. The black disks depict the (randomly generated) initial network configuration, while the gray disks depict the network configuration after 36 seconds.

VIII. Conclusions

We have considered a scenario where a robotic sensor network takes successive measurements of a dynamic physical process of interest model as a spatio-temporal random field. We have introduced a statistical framework to estimate the distribution of the random field and of its gradient. Under the assumptions of uncorrelation in time and limited-range correlation in space, we have developed the Distributed Kriged Kalman Filter that enables the network to compute the predictive mean functions of the random field and of its gradient. We have illustrated the usefulness of the proposed algorithm by synthesizing a motion coordination strategy that makes network agents find critical points of the field with probability one in case of no measurement noise. Numerous avenues of research appear worth pursuing. Among them, we highlight the consideration of more general statistical assumptions on the spatio-temporal field, (e.g., measurements correlated in time, unknown parameters in the covariance structure that must also be estimated from the data), the quantification of the communication requirements of the proposed algorithm, the extension of the convergence results of the gradient ascent algorithm to the case of measurement errors, or the design of distributed algorithms that maximize the information content of retrieved data.

Acknowledgments

This research was supported in part by NSF CAREER Award ECS-0546871. The author wishes to thank Prof. Athanasios Kottas for enlightening conversations.

References

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

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

[3] K. V. Mardia, C. Goodall, E. J. Redfern, and F. J. Alonso, "The Kriged Kalman filter," *Test*, vol. 7, no. 2, pp. 217–285, 1998. With discussion.

[4] N. A. C. Cressie and C. K. Wikle, "Space-time Kalman filter," in *Encyclopedia of Environmetrics* (A. H. El-Shaarawi and W. W. Piegorsch, eds.), vol. 4, pp. 2045–2049, New York: John Wiley, 2002.

[5] 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.

[6] S. Sukkarieh, E. Nettleton, J.-H. Kim, M. Ridley, A. Goktogan, and H. Durrant-Whyte, "The ANSER project: Data fusion across multiple uninhabited air vehicles," *International Journal of Robotics Research*, vol. 22, no. 7-8, pp. 505–539, 2003.

[7] 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.

[8] 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.

[9] 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. 95, no. 1, pp. 48–74, 2007.

[10] L. Xiao, S. Boyd, and S. Lall, "A scheme for robust distributed sensor fusion based on average consensus," in *Symposium on Information Processing of Sensor Networks*, (Los Angeles, CA), pp. 63–70, Apr. 2005.

[11] 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.

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

[13] R. Olfati-Saber and J. S. Shamma, "Consensus filters for sensor networks and distributed sensor fusion," in *IEEE Conf. on Decision and Control*, (Seville, Spain), pp. 6698–6703, 2005.

[14] 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.

[15] B. S. Y. Rao, H. F. Durrant-Whyte, and J. S. Sheen, "A fully decentralized multi-sensor system for tracking and surveillance," *International Journal of Robotics Research*, vol. 12, no. 1, pp. 20–44, 1993.

[16] J. L. Speyer, "Computation and transmission requirements for a decentralized linear-quadratic-gaussian control problem," *IEEE Transactions on Automatic Control*, vol. 24, no. 2, pp. 266–269, 1979.

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

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

[19] H. J. Kushner and G. G. Yin, *Stochastic Approximation and Recursive Algorithms and Applications*, vol. 35 of *Applications of Mathematics: Stochastic Modelling and Applied Probability*. New York: Springer, 2nd ed., 2003.

[20] F. Bullo, J. Cortés, and S. Martínez, *Distributed Control of Robotic Networks*. Applied Mathematics Series, Princeton University Press, 2009. Electronically available at http://coordinationbook.info.

[21] B. D. O. Anderson and J. B. Moore, *Optimal Filtering*. Prentice Hall, 1979.

[22] P. S. Maybeck, *Stochastic Models, Estimation, and Control*, vol. 1. New York: Academic Press, 1979.

[23] 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.

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