Mobile Sensor Network Navigation Using Gaussian Processes With Truncated Observations

Yunfei Xu, *Member, IEEE*, Jongeun Choi, *Member, IEEE*, and Songhwai Oh, *Member, IEEE*

***Abstract*—In this paper, we consider mobile sensor networks that use spatiotemporal Gaussian processes to predict a wide range of spatiotemporal physical phenomena. Nonparametric Gaussian process regression that is based on truncated observations is proposed for mobile sensor networks with limited memory and computational power. We first provide a theoretical foundation of Gaussian process regression with truncated observations. In particular, we demonstrate that prediction using all observations can be well approximated by prediction using truncated observations under certain conditions. Inspired by the analysis, we then propose a centralized navigation strategy for mobile sensor networks to move in order to reduce prediction error variances at points of interest. For the case in which each agent has a limited communication range, we propose a distributed navigation strategy. Particularly, we demonstrate that mobile sensing agents with the distributed navigation strategy produce an emergent, swarming-like, collective behavior for communication connectivity and are coordinated to improve the quality of the collective prediction capability.**

***Index Terms*—Distributed algorithms, Gaussian processes, mobile sensor networks.**

I. Introduction

In RECENT years, because of global climate changes, more environmental scientists have become interested in changing ecosystems over vast regions on land and in oceans and lakes. In order to meet such demands, it is necessary to develop autonomous robotic systems that can perform a series of tasks, such as estimation, prediction, monitoring, tracking, and removal of a scalar field of interest undergoing often complex transport phenomena. In this paper, we consider the problem of the prediction of spatiotemporal fields by the use of mobile sensor networks.

Manuscript received April 29, 2010; revised October 20, 2010 and March 14, 2011; accepted July 13, 2011. This paper was recommended for publication by Associate Editor D. Song and Editor W. K. Chung upon evaluation of the reviewers' comments. This work was supported in part by the National Science Foundation through CAREER Award CMMI-0846547. The work of S. Oh was supported by the Basic Science Research Program through the National Research Foundation of Korea funded by the Ministry of Education, Science, and Technology under Grant 2010-0013354.

Y. Xu is with the Department of Mechanical Engineering, Michigan State University, East Lansing, MI 48824 USA (e-mail: xuyunfei@egr.msu.edu).

J. Choi is with the Department of Mechanical Engineering and the Department of Electrical and Computer Engineering, Michigan State University, East Lansing, MI 48824 USA (e-mail: jchoi@egr.msu.edu).

S. Oh is with the School of Electrical Engineering and Computer Science and the Automation and Systems Research Institute, Seoul National University, Seoul 151-744, Korea (e-mail: songhwai@snu.ac.kr).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TRO.2011.2162766

Significant enhancements have been made in the area of mobile sensor networks. Emerging technologies have been reported on the coordination of mobile sensing agents [1]–[6]. Mobile sensing agents form an *ad hoc* wireless communication network in which each agent usually operates under a short communication range, with limited memory and computational power. Mobility in a sensor network can increase its sensing coverage both in space and time and robustness against dynamic changes in the environment. The mobility of mobile agents can be designed for the optimal sampling of the field of interest. Recently, in [7], mobile sensor networks that optimize ocean sampling performance that is defined in terms of uncertainty in a model estimate of a sampled field have been developed. However, this approach optimized the collective patterns of mobile agents that are parameterized by a restricted number of parameters rather than optimizing individual trajectories. In [8], distributed learning and cooperative control were developed for multiagent systems to discover peaks of the unknown field based on the recursive estimation of an unknown field. A typical sensor-placement technique [9], which puts sensors at the locations where the entropy is high, tends to place sensors along the borders of the area of interest [10]. In [10], Krause *et al.* showed that seeking sensor placements that are most informative about unsensed locations is NP-hard, and they presented a polynomial-time approximation algorithm by exploitation of the submodularity of mutual information. In a similar approach, in [11], an efficient planning of informative paths for multiple robots that maximizes the mutual information has been presented.

To find optimal locations that predict the phenomenon best, one needs a model of the spatiotemporal phenomenon. To this end, we use the Gaussian process to model fields that are undergoing transport phenomena. Nonparametric Gaussian process regression (or kriging in geostatistics) has been widely used as a nonlinear regression technique to estimate and predict geostatistical data [12]–[15]. A Gaussian process with an infinite number of random variables over a continuum space can be viewed as a generalization of a Gaussian probability distribution with a finite number of random variables. Gaussian process regression enables us to predict physical values, such as temperature and plume concentration, at any point with a predicted uncertainty level efficiently. For instance, near-optimal static-sensor placements with a mutual information criterion in Gaussian processes were proposed in [10], [16]. A distributed kriged Kalman filter for spatial estimation that is based on a mobile sensor network was developed in [17]. Multiagent systems that are versatile for various tasks by the exploitation of predictive posterior statistics of Gaussian processes were developed in [18] and [19].

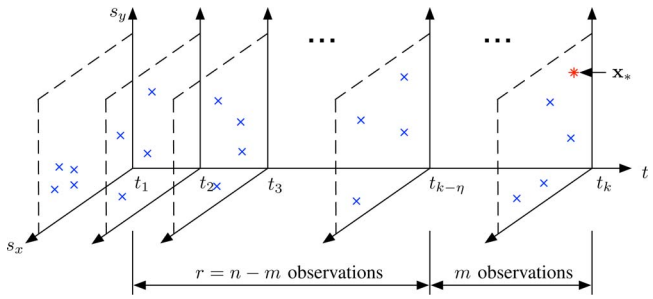


Fig. 1. Robot is supposed to predict a scalar value at $\mathbf{x}_*$ (denoted by a red star) based on cumulative n spatiotemporal observations (denoted by blue crosses). Near-optimal prediction can be obtained by the use of truncated observations, e.g., the last m observations. In this case, $\mathbf{x} = [s_x \;\; s_y \;\; t]^T$.

The motivation of our study is twofold. First, the main reason why the nonparametric prediction by the use of Gaussian processes is not popular for resource-constrained multiagent systems is the fact that the optimal prediction must use all cumulatively measured values [13], [14]. In this case, a robot needs to compute the inverse of the covariance matrix whose size grows as it collects more measurements. With this operation, the robot will run out of memory quickly. Therefore, it is necessary to develop a class of prediction algorithms by the use of spatiotemporal Gaussian processes under a fixed memory size. The spacetime Kalman filter model that is proposed in [20] and [21] and utilized in [19] partially solved this problem by modeling the spatiotemporal field as a sum of a zero-mean Gaussian process, which is uncorrelated in time, and a time-varying mean function (see [21, eqs. (6) and (12)]). The zero-mean Gaussian process represents a spatial structure that is independent from one time point to the next as described in [21] by the assumption that the dynamical environmental process is governed by a relatively large time scale. This formulation in turn provides the Markov property in time, which makes the optimal prediction recursive in time. However, the value of a temporal mean function at a point (realized by a stable linear system) consists of a linear sum of colored white noises and transient responses that converge to zero values exponentially fast [19], which cannot represent a wide range of spatiotemporal phenomena in a fully nonparametric manner [15]. A simple way to cope with this dilemma is to design a robot so that it predicts a spatiotemporal Gaussian process at the current (or future) time based on truncated observations, e.g., the last m observations from a total of n observations as shown in Fig. 1. This seems intuitive in the sense that the last m observations are more correlated than the other $r = n - m$ observations (see Fig. 1) in order to predict values at current or future time. Therefore, it is very important to analyze the performance degradation and tradeoff effects of prediction based on truncated observations compared with the one based on all cumulative observations.

The second motivation is to design and analyze distributed sampling strategies for resource-constrained mobile sensor networks. To develop distributed estimation and coordination algorithms for multiagent systems by the use of only local information from local neighboring agents has been one of the most fundamental problems in mobile sensor networks [1], [3]–[6], [8], [17]. To emphasize practicality and usefulness, it is critical to synthesize and analyze distributed sampling strategies under practical constraints, such as measurement noise and a limited communication range.

The contribution of this paper is as follows. We first present a theoretical foundation of Gaussian process regression with truncated observations. In particular, we show that the quality of prediction based on truncated observations does not deteriorate much as compared with that of prediction based on all cumulative data under certain conditions. Inspired by the analysis, we then propose a centralized navigation strategy by the use of truncated observations for resource-constrained mobile sensor networks to move in order to minimize a network-performance cost function. Under a limited communication range, a distributed navigation algorithm in which each agent uses only local information has been proposed. For the distributed strategy, a continuously differentiable network-performance cost function has been synthesized to avoid hybrid system dynamics [22] and/or chattering behaviors when agents lose or gain neighbors. We demonstrate that the distributed navigation strategy produces an emergent, swarming-like, collective behavior to maintain communication connectivity among mobile sensing agents.

This paper is organized as follows. In Section II, we introduce spatiotemporal Gaussian processes, and provide the notations for mobile sensor networks. In Section III, we review the Gaussian process regression and propose to use only truncated observations to bound the computational complexity. The error bounds to use truncated observations are analyzed for prediction at a single point. A way of selection of a temporal truncation size is also discussed. To improve the prediction quality, centralized and distributed navigation strategies for mobile sensor networks are proposed in Section IV. In Section V, simulation results illustrate the usefulness of our schemes under different conditions and parameters.

The standard notation will be used in this paper. Let $\mathbb{R}$, $\mathbb{R}_{\geq 0}$, and $\mathbb{Z}_{>0}$ denote, respectively, the set of real numbers, the set of non-negative real numbers, and the set of positive integers. The positive definiteness and the positive semidefiniteness of a matrix $\mathbf{A}$ are denoted by $\mathbf{A} \succ 0$ and $\mathbf{A} \succeq 0$, respectively. $\mathbb{E}$ denotes the expectation operator and Corr denotes the correlation operator. Let $\|\mathbf{x}\|$ denote the standard Euclidean norm (or 2-norm) of a vector $\mathbf{x}$. The induced 2-norm of a matrix $\mathbf{A}$ is denoted by $\|\mathbf{A}\|$. $\|\mathbf{y}\|_\infty$ denotes the infinity norm of a vector $\mathbf{y}$. The union of sets $\mathcal{A}$ and $\mathcal{B}$ is denoted by $\mathcal{A} \cup \mathcal{B}$. $|\mathcal{A}|$ denotes the cardinality of a set $\mathcal{A}$. Other notation will be explained in due course.

II. Preliminaries

In this section, we introduce (spatiotemporal) Gaussian processes and robotic sensor networks.

A. Gaussian Processes

A Gaussian process defines a distribution over a space of functions and it is completely specified by its mean function and covariance function. A Gaussian process is formally defined as follows.

Definition 2.1 (Gaussian process [15]): A *Gaussian process* is a collection of random variables, any finite number of which have a joint Gaussian distribution.

We consider a zero-mean Gaussian process[1] $z(\mathbf{x}) \in \mathbb{R}$ that is written as

$$z(\mathbf{x}) \sim \mathcal{GP}(0, \sigma_f^2 \mathcal{K}(\mathbf{x}, \mathbf{x}')) \tag{1}$$

where $\mathbf{x}, \mathbf{x}' \in \mathbb{R}^d$ are the inputs. The mean function is assumed to be zero and the covariance function is defined as $\sigma_f^2 \mathcal{K}(\mathbf{x}, \mathbf{x}')$. The signal variance σ_f^2, which is assumed to be constant across the input space, gives the overall vertical scale relative to the mean of the Gaussian process in the output space. The correlation between $z(\mathbf{x})$ and $z(\mathbf{x}')$, i.e., $\mathrm{Corr}(z(\mathbf{x}), z(\mathbf{x}'))$, is given by[2]

$$\mathcal{K}(\mathbf{x}, \mathbf{x}') = \exp\left(-\frac{\|\mathbf{x} - \mathbf{x}'\|^2}{2\sigma_\ell^2}\right) \tag{2}$$

where σ_ℓ is the length scale that determines the decreasing rate of the correlation between two inputs as the distance between them increases.

A spatiotemporal Gaussian process $z(\mathbf{s}, t)$ is a special case of the Gaussian process that is defined in (1), where $\mathbf{x} = [\mathbf{s}^T \quad t]^T = [s_x \quad s_y \quad t]^T \in \mathbb{R}^2 \times \mathbb{R}_{\geq 0}$. As in our previous work [23], we use the following generalized correlation function $\mathcal{K}(\mathbf{x}, \mathbf{x}')$ with a hyperparameter vector $\boldsymbol{\theta} := [\sigma_f^2 \quad \sigma_x \quad \sigma_y \quad \sigma_t]^T$:

$$\begin{aligned}\mathcal{K}(\mathbf{x}, \mathbf{x}') &= \mathcal{K}_s(\mathbf{s}, \mathbf{s}')\mathcal{K}_t(t, t') \\ &= \exp\left(-\sum_{\ell \in \{x,y\}} \frac{(s_\ell - s'_\ell)^2}{2\sigma_\ell^2}\right) \exp\left(-\frac{(t-t')^2}{2\sigma_t^2}\right)\end{aligned} \tag{3}$$

where $\mathbf{s}, \mathbf{s}' \in \mathbb{R}^2$ are the space locations, and $t, t' \in \mathbb{R}_{\geq 0}$ are the time indices. $\{\sigma_x, \sigma_y\}$ and σ_t are length scales for space and time, respectively. Equation (3) shows that points close in the measurement space and time indices are strongly correlated and produce similar values. A spatially isotropic version of the correlation function in (3) has been used in [7].

The hyperparameters of a Gaussian process can be estimated *a priori* by the maximization of the likelihood function as shown in [23]. In this paper, we assume that the hyperparameters are known *a priori*.

B. Mobile Sensor Networks

Let N be the number of sensing agents that are distributed over a 2-D surveillance region $\mathcal{Q} \subset \mathbb{R}^2$. Assume that $\mathcal{Q}$ is a compact set. The identity of each agent is indexed by $i \in \mathcal{I} := \{1, 2, \ldots, N\}$. Let $\mathbf{q}_i(t) \in \mathcal{Q}$ be the position of the ith sensing agent at time t.

Suppose, at time $t_k \in \mathcal{T} := \{t_1, t_2, \ldots\} \subset \mathbb{R}_{\geq 0}$, agent i takes a noise-corrupted measurement $y_i(t_k)$ at its current position $\mathbf{q}_i(t_k)$, i.e.

$$y_i(t_k) = z(\mathbf{q}_i(t_k), t_k) + w_i$$

where w_i is a zero-mean Gaussian white noise with variance σ_w^2.

III. Gaussian Process Regression With Truncated Observations

In this section, we review Gaussian process regression and point out the main hurdle to use it for mobile sensor networks. We propose to use truncated observations to effectively address this issue and analyze the error bounds to use truncated observations. A way to select a temporal truncation size for spatiotemporal Gaussian process regression is also presented.

A. Gaussian Process Regression

Suppose we have n noise-corrupted observations $\mathcal{D} = \{(\mathbf{x}^{(i)}, y^{(i)}) \,|\, i = 1, \ldots, n\}$. Then, the collection of observations $\mathbf{y} = [y^{(1)} \cdots y^{(n)}]^T \in \mathbb{R}^n$ has the Gaussian distribution

$$\mathbf{y} \sim \mathcal{N}(\mathbf{0}, \sigma_f^2 \mathbf{C})$$

where $\mathbf{C} = \mathrm{Corr}(\mathbf{y}, \mathbf{y}) \in \mathbb{R}^{n \times n}$ is the correlation matrix of $\mathbf{y}$, which is obtained by $(\mathbf{C})_{ij} = \mathcal{K}(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}) + \delta_{ij}/\gamma$, where $\gamma = \sigma_f^2/\sigma_w^2$ is the SNR, and δ_{ij} denotes the Dirac delta function. We can predict the value, i.e., $z_* := z(\mathbf{x}_*)$, of the Gaussian process at a point $\mathbf{x}_*$ as [15]

$$\hat{z}_* = \mathbf{k}^T \mathbf{C}^{-1} \mathbf{y} \tag{4a}$$

with a prediction error variance that is given by

$$\sigma_{\hat{z}_*}^2 = \sigma_f^2 (1 - \mathbf{k}^T \mathbf{C}^{-1} \mathbf{k}) \tag{4b}$$

where $\mathbf{k} = \mathrm{Corr}(\mathbf{y}, z_*) \in \mathbb{R}^n$ is the correlation vector between $\mathbf{y}$ and z_*, which is obtained by $(\mathbf{k})_j = \mathcal{K}(\mathbf{x}^{(j)}, \mathbf{x}_*)$. Notice that the prediction mean in (4a) and its prediction error variance in (4b) require the inversion of the correlation matrix $\mathbf{C}$, whose size depends on the number of observations n.

As mentioned in Section I, one drawback of Gaussian process regression is that its computational complexity and memory space increase as more measurements are collected, which makes the method prohibitive for robots with limited memory and computing power. To overcome this increase in complexity, a number of approximation methods for Gaussian process regression have been proposed. In particular, the sparse greedy approximation method [24], the Nystrom method [25], the informative vector machine [26], the likelihood approximation [27], and the Bayesian committee machine [28] have been shown to be effective for many problems. However, these approximation methods have been proposed without theoretical justifications.

In general, if measurements are taken from nearby locations (or spacetime locations), correlation between measurements is strong and correlation exponentially decays as the distance between locations increases. If the correlation function of a

[1] A Gaussian process with a nonzero mean can be treated by a change of variables. Even without a change of variables, this is not a drastic limitation, since the mean of the posterior process is not confined to zero [15].

[2] The squared exponential correlation function is used in this paper. However, the analysis and algorithms are not restricted to the squared exponential correlation function. Any correlation function can be used instead if it exhibits the property that the correlation decays as the distance between input points increases.

Gaussian process has this property, intuitively, we can make a good prediction at a point of interest by the use of only measurements nearby. In the next Section III-B, we formalize this idea and provide a theoretical foundation to justify Gaussian process regression with truncated observations that is proposed in this paper.

B. Error Bounds in Using Truncated Observations

Without loss of generality, we assume that the first m out of n observations are used to predict z_*. Let $r = n - m$, $\mathbf{y}_m = [y^{(1)} \cdots y^{(m)}]^T$, and $\mathbf{y}_r = [y^{(m+1)} \cdots y^{(n)}]^T$. Then, $\mathbf{C}$ and $\mathbf{k}$ can be represented as

$$\mathbf{C} = \begin{bmatrix} \mathbf{C}_m & \mathbf{K}_{mr} \\ \mathbf{K}_{mr}^T & \mathbf{C}_r \end{bmatrix} \quad \text{and} \quad \mathbf{k} = \begin{bmatrix} \mathbf{k}_m \\ \mathbf{k}_r \end{bmatrix}.$$

By the use of truncated observations, we can predict the value z_* as

$$\hat{z}'_* = \mathbf{k}_m^T \mathbf{C}_m^{-1} \mathbf{y}_m \tag{5a}$$

with a prediction error variance that is given by

$$\sigma^2_{\hat{z}'_*} = \sigma_f^2 (1 - \mathbf{k}_m^T \mathbf{C}_m^{-1} \mathbf{k}_m). \tag{5b}$$

The following result shows the gap between predicted values by the use of truncated measurements and all measurements.

Theorem 3.1: Consider a Gaussian process that is defined in (1); we have

$$\begin{aligned} \hat{z}_* - \hat{z}'_* = {} & (\mathbf{k}_r - \mathbf{K}_{mr}^T \mathbf{C}_m^{-1} \mathbf{k}_m)^T \\ & \times (\mathbf{C}_r - \mathbf{K}_{mr}^T \mathbf{C}_m^{-1} \mathbf{K}_{mr})^{-1} \\ & \times (\mathbf{y}_r - \mathbf{K}_{mr}^T \mathbf{C}_m^{-1} \mathbf{y}_m) \end{aligned} \tag{6a}$$

and

$$\begin{aligned} \sigma^2_{\hat{z}_*} - \sigma^2_{\hat{z}'_*} = {} & -\sigma_f^2 (\mathbf{k}_r - \mathbf{K}_{mr}^T \mathbf{C}_m^{-1} \mathbf{k}_m)^T \\ & \times (\mathbf{C}_r - \mathbf{K}_{mr}^T \mathbf{C}_m^{-1} \mathbf{K}_{mr})^{-1} \\ & \times (\mathbf{k}_r - \mathbf{K}_{mr}^T \mathbf{C}_m^{-1} \mathbf{k}_m) \\ < {} & 0. \end{aligned} \tag{6b}$$

Proof: See Appendix A. ∎

Corollary 3.2: The prediction error variance $\sigma^2_{\hat{z}'_*}$ is a nonincreasing function of m.

Proof: The proof is straightforward from Theorem 3.1 by letting $n = m + 1$. ∎

By the consideration of an ideal case in which the measurements $\mathbf{y}_m$ are not correlated with the remaining measurements $\mathbf{y}_r$, we have the following result.

Proposition 3.3: Under the assumptions that are used in Theorem 3.1 and for given $\mathbf{y}_r \sim \mathcal{N}(\mathbf{0}, \mathbf{C}_r)$, if $\mathbf{K}_{mr} = \mathbf{0}$, then $\hat{z}_* - \hat{z}'_* = \mathbf{k}_r^T \mathbf{C}_r^{-1} \mathbf{y}_r$ and $\sigma^2_{\hat{z}_*} - \sigma^2_{\hat{z}'_*} = -\sigma_f^2 \mathbf{k}_r^T \mathbf{C}_r^{-1} \mathbf{k}_r$. In addition, we also have

$$|\hat{z}_* - \hat{z}'_*| \le \|\mathbf{k}_r^T \mathbf{C}_r^{-1}\| \sqrt{r} \bar{y}(p_1)$$

with a nonzero probability p_1. For a desired p_1, we can find $\bar{y}(p_1)$ by solving

$$p_1 = \prod_{1 \le i \le r} \left(1 - 2\phi\left(-\frac{\bar{y}(p_1)}{\lambda_i^{1/2}}\right)\right) \tag{7}$$

where ϕ is the cumulative normal distribution, and $\{\lambda_i \,|\, i = 1, \ldots, r\}$ are the eigenvalues of $\mathbf{C}_r = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^T$ with a unitary matrix $\mathbf{U}$, i.e., $\mathbf{\Lambda} = \text{diag}(\lambda_1, \ldots, \lambda_r)$.

Proof: See Appendix B. ∎

Hence, if the magnitude of $\mathbf{K}_{mr}$ is small, then the truncation error from the usage of truncated measurements will be close to $\mathbf{k}_r^T \mathbf{C}_r^{-t} \mathbf{y}_r$. Furthermore, if we want to reduce this error, we want $\mathbf{k}_r$ to be small, i.e., when the covariance between z_* and the remaining measurements $\mathbf{y}_r$ is small. In summary, if 1) the correlation between the measurements $\mathbf{y}_m$ and the remaining measurements $\mathbf{y}_r$ is small and 2) the correlation between z_* and the remaining measurements $\mathbf{y}_r$ is small, then the truncation error is small, and $\hat{z}'_*$ can be a good approximation to $\hat{z}_*$. This idea is formalized in a more general setting in the following theorem.

Theorem 3.4: Consider a zero-mean Gaussian process that is defined in (1) with the covariance function (2) and assume that we have collected n observations, i.e., $y^{(1)}, \ldots, y^{(n)}$. Suppose that $\mathbf{K}_{mr}$ is small enough, such that $\|\mathbf{K}_{mr}^T \mathbf{C}_m^{-1} \mathbf{k}_m\| \le \|\mathbf{k}_r\|$ and $\|\mathbf{K}_{mr}^T \mathbf{C}_m^{-1} \mathbf{y}_m\| \le \delta_2 \|\mathbf{y}_r\|$, for some $\delta_2 > 0$. Given $0 < p_2 < 1$, choose $\bar{y}(p_2)$, such that $\max_{i=m+1}^n |y^{(i)}| < \bar{y}(p_2)$ with probability p_2 and $\epsilon > 0$ such that $\epsilon < 2\gamma r(1 + \delta_2)\bar{y}(p_2)$, where γ is the SNR. For $\mathbf{x}_*$, if the last $r = n - m$ data points satisfy

$$\|\mathbf{x}^{(i)} - \mathbf{x}_*\|^2 > 2\sigma_\ell^2 \log\left(2\gamma \frac{1}{\epsilon} r(1 + \delta_2)\bar{y}(p_2)\right)$$

then, with probability p_2, we have

$$|\hat{z}_* - \hat{z}'_*| < \epsilon.$$

Proof: See Appendix C. ∎

Remark 3.5: The last part of Proposition 3.3 and Theorem 3.4 seek a bound for the difference between predicted values by the use of all and truncated observations with a given probability since the difference is a random variable.

Example 3.6: We provide an illustrative example to show how to use the result of Theorem 3.4 as follows. Consider a Gaussian process that is defined in (1) and (2) with $\sigma_f^2 = 1$, $\sigma_\ell = 0.2$, and $\gamma = 100$. If we have any randomly chosen ten samples ($m = 10$) within $[0 \quad 1]^2$ and we want to make a prediction at $\mathbf{x}_* = [1 \quad 1]^T$, we choose $\bar{y}(p_2) = 2\sigma_f = 2$, such that $\max_{i=m+1}^n |y^{(i)}| < \bar{y}(p_2)$ with probability $p_2 = 0.95$. According to Theorem 3.4, if we have an extra sample $\mathbf{x}^{(11)}$ ($r = 1$) at $[2.5 \quad 2.5]^T$, which satisfies the condition $\|\mathbf{x}^{(11)} - \mathbf{x}_*\| > 0.92$, then the difference in prediction using with and without the extra sample is less than $\epsilon = 0.01$ with probability $p_2 = 0.95$.

Example 3.7: Motivated by the results presented, we take a closer look at the usefulness of using a subset of observations from a sensor network for a particular realization of the Gaussian process. We consider a particular realization that is shown in Fig. 2, where crosses represent the sampling points of a Gaussian process that is defined in (1) and (2) with $\sigma_f^2 = 1$, $\sigma_\ell = 0.2$, and

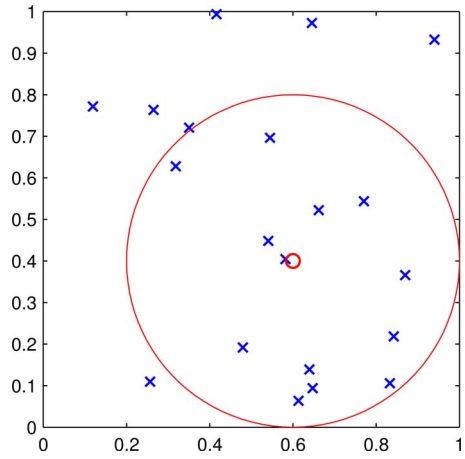


Fig. 2. Example of the selection of truncated observations. The parameters used in the example are $\sigma_f^2 = 1$, $\sigma_\ell = 0.2$, $\sigma_w = 0.1$.

TABLE I
PREDICTION MEANS AND VARIANCES USING $\mathbf{y}$, $\mathbf{y}_m$, AND $\mathbf{y}_r$

	$n = 20$	$m = 13$	$r = 7$
$\hat{z}_*$	0.531	0.52	-0.232
$\sigma^2_{\hat{z}_*}$	0.0109	0.0113	0.952

$\gamma = 100$ over $[0 \quad 1]^2$. We have selected $\mathbf{y}_m$ as the collection of observations (blue crosses) within the red circle of a radius $R = 2\sigma_\ell = 0.4$ centered at a point (a red star) located at $\mathbf{x}_* = [0.6 \quad 0.4]^T$. If a measurement is taken outside the red circle, the correlation between this measurement and the value at $\mathbf{x}_*$ decreases to 0.135. The rest of observations (blue crosses outside of the red circle) are selected as $\mathbf{y}_r$. The prediction results are shown in Table I. In this particular realization, we have $z_* = 0.539$. It can be seen that the prediction means and variances using only $\mathbf{y}_m$ are close to the one using all observations. We also compute the prediction at $\mathbf{x}_*$ with $\mathbf{y}_r$ which is far from the true value with a large variance.

The result of Theorem 3.4 and Examples 3.6 and 3.7 all suggest the usage of observations that are highly correlated with the point of interest.

C. Selecting a Temporal Truncation Size

In Section III-B, we have obtained the error bounds for the prediction at a single point. In general, the observations that are made close to that point are more informative than the others.

For a spatiotemporal Gaussian process, we define η as the truncation size, and our objective is to use only the observations made during the last η time steps, i.e., from time $t_{k-\eta+1}$ to t_k, to make prediction at time t_k. In general, a small η yields faster computation but lower accuracy and a large η yields slower computation but higher accuracy. Thus, the truncation size η should be selected according to a tradeoff relationship between accuracy and efficiency.

Next, we show an approach to select the truncation size η in an averaged performance sense. Given the observations and associated sampling locations and times (denoted by $\mathcal{D}$, which depends on η), the generalization error $\epsilon(\mathbf{x}_*, \mathcal{D})$ at a point $\mathbf{x}_* = [\mathbf{s}_*^T \quad t_*]^T$ is defined as the prediction error variance $\sigma^2_{\hat{z}_*}$ [29], [30]. For a given t_* not knowing user specific $\mathbf{s}_*$ *a priori*, we seek to find η that guarantees a low prediction error variance uniformly over the entire space $\mathcal{Q}$, i.e., we want $\epsilon(\mathcal{D}) = \mathbb{E}_{\mathbf{s}_*}[\sigma^2_{\hat{z}_*}]$ to be small [29], [30]. Here, $\mathbb{E}_{\mathbf{s}_*}$ denotes the expectation with respect to the uniform distribution of $\mathbf{s}_*$.

According to Mercer's theorem, we know that the kernel function $\mathcal{K}_s$ can be decomposed into

$$\mathcal{K}_s(\mathbf{s}, \mathbf{s}') = \sum_{i=1}^{\infty} \lambda_i \phi_i(\mathbf{s}) \phi_i(\mathbf{s}')$$

where $\{\lambda_i\}$ and $\{\phi_i(\cdot)\}$ are the eigenvalues and corresponding eigenfunctions, respectively [30]. In a similar way shown in [30], the input-dependent generalization error $\epsilon(\mathcal{D})$ for our spatiotemporal Gaussian process can be obtained as

$$\begin{aligned}\epsilon(\mathcal{D}) &= \mathbb{E}_{\mathbf{s}_*}\left[\sigma_f^2\left(1 - \mathrm{tr}\left(\mathbf{k}\mathbf{k}^T(\mathbf{K} + 1/\gamma\mathbf{I})^{-1}\right)\right)\right] \\ &= \sigma_f^2\left(1 - \mathrm{tr}\left(\mathbb{E}_{\mathbf{s}_*}[\mathbf{k}\mathbf{k}^T](\mathbf{K} + 1/\gamma\mathbf{I})^{-1}\right)\right). \qquad (8)\end{aligned}$$

We have

$$\mathbb{E}_{\mathbf{s}_*}[\mathbf{k}\mathbf{k}^T] = \mathbf{\Psi}\mathbf{\Lambda}^2\mathbf{\Psi}^T \circ \mathbf{k}_t\mathbf{k}_t^T \qquad (9)$$

and

$$\mathbf{K} = \mathbf{\Psi}\mathbf{\Lambda}\mathbf{\Psi}^T \circ \mathbf{K}_t\mathbf{K}_t^T \qquad (10)$$

where $(\mathbf{\Psi})_{ij} = \phi_j(\boldsymbol{s}_i)$, $(\mathbf{k}_t)_j = \mathcal{K}_t(t^{(j)}, t_*)$, $(\mathbf{K}_t)_{ij} = \mathcal{K}_t(t^{(i)}, t^{(j)})$, and $(\mathbf{\Lambda})_{ij} = \lambda_i\delta_{ij}$. δ_{ij} denotes the Dirac delta function. "$\circ$" denotes the Hadamard (element-wise) product [30]. Hence, the input-dependent generalization error $\epsilon(\mathcal{D})$ can be computed analytically by plugging (9) and (10) into (8). Notice that $\epsilon(\mathcal{D})$ is a function of inputs (i.e., the sampling locations and times). To obtain an averaged performance level without the knowledge of the algorithmic sampling strategy *a priori*, we use an appropriate sampling distribution, which models the stochastic behavior of the sampling strategy. Thus, further averaging over the observation set $\mathcal{D}$ with the sampling distribution yields $\epsilon(\eta) = \mathbb{E}_{\mathcal{D}}[\epsilon(\mathcal{D})]$, which is a function of the truncation size η only. This averaging process can be done by the use of Monte Carlo methods. Then, η can be chosen based on the averaged performance measure $\epsilon(\eta)$ under the sampling distribution.

An alternative way, without the usage of the eigenvalues and eigenfunctions, is to directly and numerically compute $\epsilon(\mathcal{D}) = \mathbb{E}_{\mathbf{s}_*}[\sigma^2_{\hat{z}_*}]$ uniformly over the entire space $\mathcal{Q}$ with random sampling positions at each time step. An averaged generalization error with respect to the temporal truncation size can be plotted by the usage of such Monte Carlo methods. Then, the temporal truncation size η can be chosen such that a given level of the averaged generalization error is achieved.

Example 3.8: Consider a problem of selection of a temporal truncation size η for spatiotemporal Gaussian process regression using observations from nine agents. The spatiotemporal Gaussian process is defined in (1) and (3) with $\sigma_f^2 = 1$, $\sigma_x = \sigma_y = 0.2$, $\sigma_t = 5$, and $\gamma = 100$ over $[0 \quad 1]^2$. The Monte Carlo simulation result is shown in Fig. 3. The achieved generalization error $\epsilon(\mathcal{D})$ is plotted in blue circles with error bars

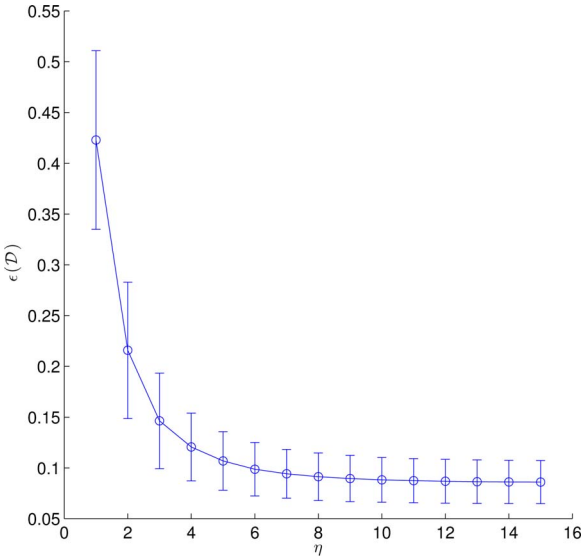


Fig. 3. Example of selecting a temporal truncation size η. The parameters used in the example are $\sigma_f^2 = 1, \sigma_x = \sigma_y = 0.2, \sigma_t = 5, \gamma = 100$.

with respect to the temporal truncation size η. As can be seen, an averaged generalization error (in blue circles) under 0.1 can be achieved by the use of observations taken from last ten time steps.

Notice that the prediction error variances can be significantly minimized by the optimal selection of the sampling positions. Hence, the selected η guarantees at least the averaged performance level of the sensor network when the optimal sampling strategy is used.

By the use of a fixed truncation size η, the computational complexity and memory space required to make prediction [i.e., to evaluate (4a) and (4b)] do not increase as more measurements are collected. Our next objective is to improve the quality of the prediction by the careful selection of the future sampling positions for the mobile sensor network.

IV. Navigation Strategies

At time t_k, the goal of the mobile sensor network is to make prediction at prespecified points of interest $\{(\mathbf{v}_j, \tau_j) \,|\, j \in \mathcal{J}\}$ indexed by $\mathcal{J} := \{1, \ldots, M\}$. From here on, points of interest will be denoted as *target points*. The introduction of target points is motivated by the fact that the potential environmental concerns should be frequently monitored. For instance, the target points can be assigned at the interface of a factory and a lake, sewage systems, or polluted beaches. Thus, the introduction of target points, which can be arbitrarily specified by a user, provides a flexible way to define a geometrical shape of a subregion of interest in a surveillance region. Notice that the target points can be changed by a user at any time. In particular, we allow that the number of target points M can be larger than that of agents N, which is often the case in practice. The prediction of $z_j := z(\mathbf{v}_j, \tau_j)$ of the Gaussian process at a target point $(\mathbf{v}_j, \tau_j)$ can be obtained as in (4a) and (4b).

A. Centralized Navigation Strategy

Consider a case in which a central station receives collective measurements from all N mobile sensors and performs the prediction. We denote the collection of positions of all N agents at time t as $\mathbf{q}(t)$, i.e.,

$$\mathbf{q}(t) = [\mathbf{q}_1(t)^T \cdots \mathbf{q}_N(t)^T]^T.$$

The collective measurements from all N mobile sensors at time $t \in \mathcal{T}$ is denoted by $\mathbf{y}_k := [y_1(t_k) \cdots y_N(t_k)]^T$. For notational simplicity, we also define the cumulative measurements that have been taken from time $t_{k-\eta+1}$ to t_k as

$$\mathbf{y}_{k-\eta+1:k} = [\mathbf{y}_{k-\eta+1}^T \cdots \mathbf{y}_k^T]^T.$$

Let the central station discard the oldest set of measurements $\mathbf{y}_{k-\eta+1}$ after making the prediction at time t_k. At the next time index t_{k+1}, by the usage of the remained observations $\mathbf{y}_{k-\eta+2:k}$ in the memory along with new measurements $\mathbf{y}_{k+1}$ from all N agents at time t_{k+1}, the central station will predict $z(\mathbf{s}_*, t_*)$ evaluated at target points $\{(\mathbf{v}_j, \tau_j)\}_{j=1}^M$. Hence, agents should move to the most informative locations to take measurements at time t_{k+1} [10].

For notational simplicity, let $\bar{\mathbf{y}}$ be the remained observations, i.e., $\bar{\mathbf{y}} := \mathbf{y}_{k-\eta+2:k}$, and $\tilde{\mathbf{y}}$ be the measurements that will be taken at positions $\tilde{\mathbf{q}} = [\tilde{\mathbf{q}}_1^T \cdots \tilde{\mathbf{q}}_N^T]^T \in \mathcal{Q}^N$ and time t_{k+1}. In contrast with the information-theoretic control strategies using the conditional entropy or the mutual information criterion [10], [31], in this paper, the mobility of the robotic sensors will be designed such that they directly minimize the average of the prediction error variances over target points, i.e.,

$$J_c(\tilde{\mathbf{q}}) = \frac{1}{|\mathcal{J}|} \sum_{j \in \mathcal{J}} \sigma_{\hat{z}_j}^2(\tilde{\mathbf{q}}) \tag{11}$$

where $|\mathcal{J}| = M$ is the cardinality of $\mathcal{J}$. The prediction error variance at each of M target points is given by

$$\sigma_{\hat{z}_j}^2(\tilde{\mathbf{q}}) = \sigma_f^2 \left(1 - \mathbf{k}_j(\tilde{\mathbf{q}})^T \mathbf{C}(\tilde{\mathbf{q}})^{-1} \mathbf{k}_j(\tilde{\mathbf{q}})\right) \quad \forall j \in \mathcal{J}$$

where $\mathbf{k}_j(\tilde{\mathbf{q}})$ and $\mathbf{C}(\tilde{\mathbf{q}})$ are defined as

$$\mathbf{k}_j(\tilde{\mathbf{q}}) = \begin{bmatrix} \mathrm{Corr}(\bar{\mathbf{y}}, z_j) \\ \mathrm{Corr}(\tilde{\mathbf{y}}, z_j) \end{bmatrix}$$

$$\mathbf{C}(\tilde{\mathbf{q}}) = \begin{bmatrix} \mathrm{Corr}(\bar{\mathbf{y}}, \bar{\mathbf{y}}) & \mathrm{Corr}(\bar{\mathbf{y}}, \tilde{\mathbf{y}}) \\ \mathrm{Corr}(\tilde{\mathbf{y}}, \bar{\mathbf{y}}) & \mathrm{Corr}(\tilde{\mathbf{y}}, \tilde{\mathbf{y}}) \end{bmatrix}.$$

In order to reduce the average of prediction error variances over target points $\{(\mathbf{v}_j, \tau_j)\}_{j=1}^M$, the central station solves the following optimization problem:

$$\mathbf{q}(t_{k+1}) = \arg \min_{\tilde{\mathbf{q}} \in \mathcal{Q}^N} J_c(\tilde{\mathbf{q}}). \tag{12}$$

Notice that in this problem setup, we only consider the constraint that robots should move within the region $\mathcal{Q}$. However, the mobility constraints, such as the maximum distance that a robot can move between two time indices or the maximum speed with which a robot can travel, can be incorporated as additional constraints in the optimization problem [17].

The sensor network configuration $\mathbf{q}(t)$ can be controlled by a gradient descent algorithm such that $\mathbf{q}(t)$ can move to a local

minimum of J_c for the prediction at time t_{k+1}:

$$\frac{d\mathbf{q}(t)}{dt} = -\nabla_{\mathbf{q}} J_c(\mathbf{q}(t)) \tag{13}$$

where $\nabla_{\mathbf{x}} J_c(\mathbf{x})$ denotes the gradient of $J_c(\mathbf{x})$ at $\mathbf{x}$. A critical point of $J_c(\mathbf{q})$ that is obtained in (13) will be $\mathbf{q}(t_{k+1})$. The analytical form of $\partial\sigma^2_{\hat{z}_j}(\tilde{\mathbf{q}})/\partial\tilde{\mathbf{q}}_{i,\ell}$, where $\tilde{\mathbf{q}}_{i,\ell}$ is the ℓth element in $\tilde{\mathbf{q}}_i \in \mathcal{Q}$, can be obtained by

$$\frac{\partial\sigma^2_{\hat{z}_j}(\tilde{\mathbf{q}})}{\partial\tilde{\mathbf{q}}_{i,\ell}} = \mathbf{k}_j^T \mathbf{C}^{-1}\left(\frac{\partial\mathbf{C}}{\partial\tilde{\mathbf{q}}_{i,\ell}}\mathbf{C}^{-1}\mathbf{k}_j - 2\frac{\partial\mathbf{k}_j}{\partial\tilde{\mathbf{q}}_{i,\ell}}\right) \qquad \forall i \in \mathcal{I}, \quad \ell \in \{1,2\}.$$

Other more advanced nonlinear optimization techniques may be applied to solve the optimization problem in (12) [32].

The centralized sampling strategy for the mobile sensor network with the cost function J_c in (11) is summarized in Table II. Notice that the prediction in the centralized sampling strategy uses temporally truncated observations. A decentralized version of the centralized sampling strategy in Table II may be developed by the usage of the approach proposed in [33], in which each robot incrementally refines its decision while intermittently communicating with the rest of the robots.

TABLE II
CENTRALIZED SAMPLING STRATEGY AT TIME t_k

Input:
(1) Number of agents N
(2) Positions of agents $\{\mathbf{q}_i(t_k)\}_{i=1}^N$
(3) Hyperparameters of the Gaussian process $\boldsymbol{\theta} = [\ \sigma_f^2 \quad \sigma_x \quad \sigma_y \quad \sigma_t\ ]^T$
(4) Target points $\{(\mathbf{v}_j, \tau_j)\}_{j=1}^M$
(5) Truncation size η

Output:
(1) Prediction at target points $\{\hat{z}_j\}_{j=1}^M$
(2) Prediction error variance at target points $\{\sigma^2_{\hat{z}_j}\}_{j=1}^M$

For $i \in \mathcal{I}$, agent i performs:
1: make an observation at current position $\mathbf{q}_i(t_k)$, i.e., $y_i(t_k)$
2: transmit the observation $y_i(t_k)$ to the central station

The central station performs:
1: collect the observations from all N agents, i.e., $\mathbf{y}_k = [\ y_1(t_k) \quad \cdots \quad y_N(t_k)\ ]^T$
2: obtain the cumulative measurements, i.e., $\mathbf{y}_{k-\eta+1:k} = [\ \mathbf{y}_{k-\eta+1}^T \quad \cdots \quad \mathbf{y}_k^T\ ]^T$
3: **for** $j \in \mathcal{J}$ **do**
4: make prediction at a target point $(\mathbf{v}_j, \tau_j)$

$$\hat{z}_j = \mathbf{k}^T\mathbf{C}^{-1}\mathbf{y},$$

with a prediction error variance given by

$$\sigma^2_{\hat{z}_j} = \sigma_f^2(1 - \mathbf{k}^T\mathbf{C}^{-1}\mathbf{k}),$$

where $\mathbf{y} = \mathbf{y}_{k-\eta+1:k}$, $\mathbf{k} = \mathrm{Corr}(\mathbf{y}, z_j)$, and $\mathbf{C} = \mathrm{Corr}(\mathbf{y}, \mathbf{y})$
5: **end for**
6: **if** $k \geq \eta$ **then**
7: discard the oldest set of measurements taken at time $t_{k-\eta+1}$, i.e., $\mathbf{y}_{k-\eta+1}$
8: **end if**
9: compute the control with the remained data $\mathbf{y}_{k-\eta+2:k}$

$$\mathbf{q}(t_{k+1}) = \arg\min_{\tilde{\mathbf{q}}\in\mathcal{Q}^N} J_c(\tilde{\mathbf{q}}).$$

via

$$\frac{d\mathbf{q}(t)}{dt} = -\nabla_{\mathbf{q}} J_c(\mathbf{q}(t))$$

10: send the next sampling positions $\{\mathbf{q}_i(t_{k+1})\}_{i=1}^N$ (a critical point of $J_c(\tilde{\mathbf{q}})$) to all N agents

For $i \in \mathcal{I}$, agent i performs:
1: receive the next sampling position $\mathbf{q}_i(t_{k+1})$ from the central station
2: move to $\mathbf{q}_i(t_{k+1})$ before time t_{k+1}

B. Distributed Navigation Strategy

Now, we consider a case in which each agent in the sensor network can only communicate with other agents within a limited communication range R. In addition, no central station exists. In this section, we present a distributed navigation strategy for mobile agents that uses only local information in order to minimize a collective network-performance cost function.

The communication network of mobile agents can be represented by an undirected graph. Let $\mathcal{G}(t) := (\mathcal{I}, \mathcal{E}(t))$ be an undirected communication graph such that an edge $(i,j) \in \mathcal{E}(t)$ if and only if agent i can communicate with agent j at time t. We define the neighborhood of agent i at time t by $\mathcal{N}_i(t) := \{j \in \mathcal{I} \,|\, (i,j) \in \mathcal{E}(t),\ j \neq i\}$. In particular, we have

$$\mathcal{N}_i(t) = \{j \in \mathcal{I} \,|\, \|\mathbf{q}_i(t) - \mathbf{q}_j(t)\| < R,\ j \neq i\}.$$

Note that in our definition, "$<$ is used instead of "$\leq$ to decide the communication range.

At time $t_k \in \mathcal{T}$, agent i collects measurements $\{y_j(t_k) \,|\, j \in \{i\} \cup \mathcal{N}_i(t_k)\}$ sampled at $\{\mathbf{q}_j(t_k) \,|\, j \in \{i\} \cup \mathcal{N}_i(t_k)\}$ from its neighbors and itself. The collection of these observations and the associated sampling positions in vector forms are denoted by $\mathbf{y}_k^{[i]}$ and $\mathbf{q}^{[i]}(t_k)$, respectively. Similarly, for notational simplicity, we also define the cumulative measurements that have been collected by agent i from time $t_{k-\eta+1}$ to t_k as

$$\mathbf{y}^{[i]}_{k-\eta+1:k} = [(\mathbf{y}^{[i]}_{k-\eta+1})^T \cdots (\mathbf{y}^{[i]}_k)^T]^T.$$

In contrast with the centralized scheme, in the distributed scheme, each agent determines the sampling points based on the local information from neighbors. After making the prediction at time t_k, agent i discards the oldest set of measurements $\mathbf{y}^{[i]}_{k-\eta+1}$. At time t_{k+1}, by the usage of the remained observations $\mathbf{y}^{[i]}_{k-\eta+2:k}$ in the memory along with new measurements $\mathbf{y}^{[i]}_{k+1}$ from its neighbors in $\mathcal{N}_i(t_{k+1})$, agent i will predict $z(\mathbf{s}_*, t_*)$ evaluated at target points $\{(\mathbf{v}_j, \tau_j)\}_{j=1}^M$.

For notational simplicity, let $\bar{\mathbf{y}}^{[i]}$ be the remained observations of agent i, i.e., $\bar{\mathbf{y}}^{[i]} := \mathbf{y}^{[i]}_{k-\eta+2:k}$. Let $\tilde{\mathbf{y}}^{[i]}$ be the new measurements that will be taken at positions of agent i and its neighbors $\tilde{\mathbf{q}}^{[i]} \in \mathcal{Q}^{|\mathcal{N}_i|+1}$, and at time t_{k+1}, where $|\mathcal{N}_i|$ is the number of neighbors of agent i at time t_{k+1}. The prediction error variance that is obtained by agent i at each of M target points (indexed by $\mathcal{J}$) is given by

$$\sigma^{2[i]}_{\hat{z}_j}(\tilde{\mathbf{q}}^{[i]}) = \sigma_f^2\left(1 - \mathbf{k}_j^{[i]}(\tilde{\mathbf{q}}^{[i]})^T\mathbf{C}^{[i]}(\tilde{\mathbf{q}}^{[i]})^{-1}\mathbf{k}_j^{[i]}(\tilde{\mathbf{q}}^{[i]})\right) \qquad \forall j \in \mathcal{J}$$

where $\mathbf{k}_j^{[i]}(\tilde{\mathbf{q}}^{[i]})$ and $\mathbf{C}^{[i]}(\tilde{\mathbf{q}}^{[i]})$ are defined as

$$\mathbf{k}_j^{[i]}(\tilde{\mathbf{q}}^{[i]}) = \begin{bmatrix} \mathrm{Corr}(\bar{\mathbf{y}}^{[i]}, z_j) \\ \mathrm{Corr}(\tilde{\mathbf{y}}^{[i]}, z_j) \end{bmatrix}$$

$$\mathbf{C}^{[i]}(\tilde{\mathbf{q}}^{[i]}) = \begin{bmatrix} \text{Corr}(\bar{\mathbf{y}}^{[i]}, \bar{\mathbf{y}}^{[i]}) & \text{Corr}(\bar{\mathbf{y}}^{[i]}, \tilde{\mathbf{y}}^{[i]}) \\ \text{Corr}(\tilde{\mathbf{y}}^{[i]}, \bar{\mathbf{y}}^{[i]}) & \text{Corr}(\tilde{\mathbf{y}}^{[i]}, \tilde{\mathbf{y}}^{[i]}) \end{bmatrix}. \quad (14)$$

The performance of agent i can be evaluated by the average of the prediction error variances over target points, i.e.,

$$J^{[i]}(\tilde{\mathbf{q}}^{[i]}) = \frac{1}{|\mathcal{J}|} \sum_{j \in \mathcal{J}} \sigma^{2[i]}_{\hat{z}_j}(\tilde{\mathbf{q}}^{[i]}) \quad \forall i \in \mathcal{I}.$$

One criterion to evaluate the network performance is the average of individual performance, i.e.,

$$J(\tilde{\mathbf{q}}) = \frac{1}{|\mathcal{I}|} \sum_{i \in \mathcal{I}} J^{[i]}(\tilde{\mathbf{q}}^{[i]}). \quad (15)$$

However, the discontinuity of the function J occurs at the moment of gaining or losing neighbors, e.g., at the set

$$\{\tilde{\mathbf{q}} \,|\, \|\tilde{\mathbf{q}}_i - \tilde{\mathbf{q}}_j\| = R\}.$$

A gradient decent algorithm for mobile robots that minimizes such J may produce hybrid system dynamics and/or chattering behaviors when robots lose or gain neighbors.

Therefore, we seek to minimize an upper bound of J that is continuously differentiable. Consider the following function:

$$\bar{\sigma}^{2[i]}_{\hat{z}_j}(\tilde{\mathbf{q}}^{[i]}) = \sigma_f^2 \left(1 - \mathbf{k}_j^{[i]}(\tilde{\mathbf{q}}^{[i]})^T \bar{\mathbf{C}}^{[i]}(\tilde{\mathbf{q}}^{[i]})^{-1} \mathbf{k}_j^{[i]}(\tilde{\mathbf{q}}^{[i]})\right) \quad \forall j \in \mathcal{J} \quad (16)$$

where $\bar{\mathbf{C}}^{[i]}(\tilde{\mathbf{q}}^{[i]})$ is defined as

$$\bar{\mathbf{C}}^{[i]}(\tilde{\mathbf{q}}^{[i]}) = \begin{bmatrix} \text{Corr}(\bar{\mathbf{y}}^{[i]}, \bar{\mathbf{y}}^{[i]}) & \text{Corr}(\bar{\mathbf{y}}^{[i]}, \tilde{\mathbf{y}}^{[i]}) \\ \text{Corr}(\tilde{\mathbf{y}}^{[i]}, \bar{\mathbf{y}}^{[i]}) & \text{Corr}(\tilde{\mathbf{y}}^{[i]}, \tilde{\mathbf{y}}^{[i]}) + \tilde{\mathbf{C}}^{[i]}(\tilde{\mathbf{q}}^{[i]}) \end{bmatrix}.$$

Notice that $\bar{\mathbf{C}}^{[i]}(\tilde{\mathbf{q}}^{[i]})$ is obtained by adding a positive semidefinite matrix $\tilde{\mathbf{C}}^{[i]}(\tilde{\mathbf{q}}^{[i]})$ to the lower right block of $\mathbf{C}^{[i]}(\tilde{\mathbf{q}}^{[i]})$ in (14), where

$$\tilde{\mathbf{C}}^{[i]}(\tilde{\mathbf{q}}^{[i]}) = \text{diag}\left(\Phi(d_{i1})^{-1}, \ldots, \Phi(d_{i(|\mathcal{N}_i|+1)})^{-1}\right) - \frac{1}{\gamma}\mathbf{I}$$

where $d_{ij} := \|\tilde{\mathbf{q}}_i - \tilde{\mathbf{q}}_j\|$ is the distance between agents i and j $\forall j \in \{i\} \cup \mathcal{N}_i$. $\Phi : [0, R) \mapsto (0, \gamma]$ is a continuously differentiable function that is defined as

$$\Phi(d) = \gamma\phi\left(\frac{d + d_0 - R}{d_0}\right) \quad (17)$$

where

$$\phi(h) = \begin{cases} 1, & h \le 0 \\ \exp\left(\frac{-h^2}{1-h^2}\right), & 0 < h < 1. \end{cases}$$

An example of $\Phi(d)$, where $\gamma = 100$, $R = 0.4$, and $d_0 = 0.1$, is shown as the red dotted line in Fig. 4. Notice that if $\Phi(d) = \gamma$ is used (the blue solid line in Fig. 4), we have $\bar{\mathbf{C}}^{[i]}(\tilde{\mathbf{q}}^{[i]}) = \mathbf{C}^{[i]}(\tilde{\mathbf{q}}^{[i]})$. We then have the following result.

Proposition 4.1: $\bar{\sigma}^{2[i]}_{\hat{z}_j}(\tilde{\mathbf{q}}^{[i]})$ is an upper bound of $\sigma^{2[i]}_{\hat{z}_j}(\tilde{\mathbf{q}}^{[i]})$ $\forall i \in \mathcal{I}$.

Proof: Let $\mathbf{A} := \mathbf{C}^{[i]}(\tilde{\mathbf{q}}^{[i]})$ and $\mathbf{B} := \text{diag}(\mathbf{0}, \tilde{\mathbf{C}}^{[i]}(\tilde{\mathbf{q}}^{[i]}))$. The result follows immediately from the fact that $(\mathbf{A} + \mathbf{B})^{-1} \preceq \mathbf{A}^{-1}$ for any $\mathbf{A} \succ 0$ and $\mathbf{B} \succeq 0$. ■

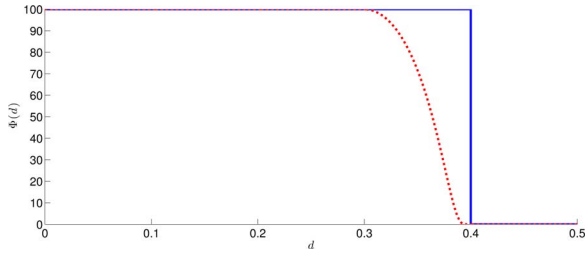


Fig. 4. Function $\Phi(d)$ in (17) with $\gamma = 100$, $R = 0.4$, and $d_0 = 0.1$ is shown as a red dotted line. The function $\Phi(d) = \gamma$ is shown as a blue solid line.

Hence, we construct a new cost function as

$$J_d(\tilde{\mathbf{q}}) = \frac{1}{|\mathcal{I}|} \sum_{i \in \mathcal{I}} \frac{1}{|\mathcal{J}|} \sum_{j \in \mathcal{J}} \bar{\sigma}^{2[i]}_{\hat{z}_j}(\tilde{\mathbf{q}}^{[i]}). \quad (18)$$

By Proposition 4.1, J_d in (18) is an upper bound of J in (15).

Next, we show that J_d is continuously differentiable when agents gain or lose neighbors. In doing so, we compute the partial derivative of J_d with respect to $\tilde{\mathbf{q}}_{i,\ell}$, where $\tilde{\mathbf{q}}_{i,\ell}$ is the ℓth element in $\tilde{\mathbf{q}}_i \in \mathcal{Q}$ as follows:

$$\begin{aligned} \frac{\partial J_d(\tilde{\mathbf{q}})}{\partial \tilde{\mathbf{q}}_{i,\ell}} &= \frac{1}{|\mathcal{I}|} \sum_{k \in \mathcal{I}} \frac{1}{|\mathcal{J}|} \sum_{j \in \mathcal{J}} \frac{\partial \bar{\sigma}^{2[k]}_{\hat{z}_j}(\tilde{\mathbf{q}}^{[k]})}{\partial \tilde{\mathbf{q}}_{i,\ell}} \\ &= \frac{1}{|\mathcal{I}|} \sum_{k \in \{i\} \cup \mathcal{N}_i} \frac{1}{|\mathcal{J}|} \sum_{j \in \mathcal{J}} \frac{\partial \bar{\sigma}^{2[k]}_{\hat{z}_j}(\tilde{\mathbf{q}}^{[k]})}{\partial \tilde{\mathbf{q}}_{i,\ell}} \\ &\forall i \in \mathcal{I}, \quad \ell \in \{1, 2\}. \end{aligned} \quad (19)$$

We then have the following.

Proposition 4.2: The cost function J_d in (18) is of class C^1, i.e., it is continuously differentiable.

Proof: We need to show that the partial derivatives of J_d with respect to $\tilde{\mathbf{q}}_{i,\ell}$ $\forall i \in \mathcal{I}, \ell \in \{1, 2\}$ exist and are continuous. Without loss of generality, we show that $\partial J_d / \partial \tilde{\mathbf{q}}_{i,\ell}$ $\forall \ell \in \{1, 2\}$ is continuous at any point $\tilde{\mathbf{q}}^*$ in the boundary set that is defined by

$$\mathcal{S}_{ik} := \{\tilde{\mathbf{q}} \,|\, d_{ik} = \|\tilde{\mathbf{q}}_i - \tilde{\mathbf{q}}_k\| = R\}.$$

First, we consider a case in which $\tilde{\mathbf{q}} \notin \mathcal{S}_{ik}$ and $d_{ik} < R$, i.e., $k \in \mathcal{N}_i$ and $i \in \mathcal{N}_k$. By the construction of $\bar{\sigma}^{2[i]}_{\hat{z}_j}$ in (16) using (17), when we take the limit of the partial derivative, as d_{ik} approaches R from below (as $\tilde{\mathbf{q}}$ approaches $\tilde{\mathbf{q}}^*$), we have that

$$\lim_{d_{ik} \to R_-} \frac{\partial \bar{\sigma}^{2[i]}_{\hat{z}_j}(\tilde{\mathbf{q}}^{[i]})}{\partial \tilde{\mathbf{q}}_{i,\ell}} = \frac{\partial \bar{\sigma}^{2[i]}_{\hat{z}_j}(\tilde{\mathbf{q}}^{[i]} \backslash \tilde{\mathbf{q}}_k)}{\partial \tilde{\mathbf{q}}_{i,\ell}}$$

$$\lim_{d_{ik} \to R_-} \frac{\partial \bar{\sigma}^{2[k]}_{\hat{z}_j}(\tilde{\mathbf{q}}^{[k]})}{\partial \tilde{\mathbf{q}}_{i,\ell}} = \frac{\partial \bar{\sigma}^{2[k]}_{\hat{z}_j}(\tilde{\mathbf{q}}^{[k]} \backslash \tilde{\mathbf{q}}_i)}{\partial \tilde{\mathbf{q}}_{i,\ell}} = 0$$

where $\tilde{\mathbf{q}}^{[a]} \backslash \tilde{\mathbf{q}}_b$ denotes the collection of locations of agent a and its neighbors excluding $\tilde{\mathbf{q}}_b$. Hence, we have

$$\lim_{d_{ik} \to R_-} \frac{\partial J_d(\tilde{\mathbf{q}})}{\partial \tilde{\mathbf{q}}_{i,\ell}} = \frac{\partial J_d(\tilde{\mathbf{q}}^*)}{\partial \tilde{\mathbf{q}}_{i,\ell}}. \quad (20)$$

Consider the other case in which $\tilde{\mathbf{q}} \notin \mathcal{S}_{ik}$ and $d_{ik} > R$, i.e., $k \notin \mathcal{N}_i$ and $i \notin \mathcal{N}_k$. When d_{ik} approaches R from above (as $\tilde{\mathbf{q}}$

approaches $\tilde{\mathbf{q}}^*$), we have

$$\lim_{d_{ik}\to R_+} \frac{\partial \bar{\sigma}^{2[i]}_{\hat{z}_j}(\tilde{\mathbf{q}}^{[i]})}{\partial \tilde{\mathbf{q}}_{i,\ell}} = \frac{\partial \bar{\sigma}^{2[i]}_{\hat{z}_j}(\tilde{\mathbf{q}}^{[i]})}{\partial \tilde{\mathbf{q}}_{i,\ell}}$$

and hence

$$\lim_{d_{ik}\to R_+} \frac{\partial J_d(\tilde{\mathbf{q}})}{\partial \tilde{\mathbf{q}}_{i,\ell}} = \frac{\partial J_d(\tilde{\mathbf{q}}^*)}{\partial \tilde{\mathbf{q}}_{i,\ell}}. \tag{21}$$

Therefore, from (20) and (21), we have

$$\lim_{d_{ik}\to R_-} \frac{\partial J_d(\tilde{\mathbf{q}})}{\partial \tilde{\mathbf{q}}_{i,\ell}} = \lim_{d_{ik}\to R_+} \frac{\partial J_d(\tilde{\mathbf{q}})}{\partial \tilde{\mathbf{q}}_{i,\ell}} = \frac{\partial J_d(\tilde{\mathbf{q}}^*)}{\partial \tilde{\mathbf{q}}_{i,\ell}}.$$

This completes the proof due to [34, Th. 4.6]. ■

By the use of J_d in (18), a gradient descent algorithm can be used to minimize the network-performance cost function J_d in (18) for the prediction at t_{k+1}:

$$\frac{d\mathbf{q}(t)}{dt} = -\nabla_{\mathbf{q}} J_d(\mathbf{q}(t)). \tag{22}$$

Note that the partial derivative in (19), which builds the gradient flow in (22), is a function of positions in $\cup_{j\in\mathcal{N}_i(t)}\mathcal{N}_j(t)$ only. This makes the algorithm distributed. A distributed sampling strategy for agent i with the network-performance cost function J_d in (18) is summarized in Table III. This way, each agent with the distributed sampling strategy uses spatially and temporally truncated observations.

TABLE III
DISTRIBUTED SAMPLING STRATEGY AT TIME t_k

Input:
(1) Number of agents N
(2) Positions of agents $\{\mathbf{q}_i(t_k)\}_{i=1}^N$
(3) Hyperparameters of the Gaussian process $\boldsymbol{\theta} = [\ \sigma_f^2 \quad \sigma_x \quad \sigma_y \quad \sigma_t\ ]^T$
(4) Target points $\{(\mathbf{v}_j, \tau_j)\}_{j=1}^M$
(5) Truncation size η

Output:
(1) Prediction at target points $\{\{\hat{z}_j^{[i]}\}_{j=1}^M\}_{i=1}^N$
(2) Prediction error variances at target points $\{\{\sigma^{2[i]}_{\hat{z}_j}\}_{j=1}^M\}_{i=1}^N$

For $i \in \{1, \cdots, N\}$, agent i performs:

1: make an observation at $\mathbf{q}_i(t_k)$, i.e., $y_i(t_k)$
2: transmit the observation to the neighbors in $\mathcal{N}_i(t_k)$
3: collect the observations from neighbors in $\mathcal{N}_i(t_k)$, i.e., $\mathbf{y}^{[i]}(t_k)$
4: obtain the cumulative measurements, i.e., $\mathbf{y}^{[i]}_{k-\eta+1:k} = [\ (\mathbf{y}^{[i]}_{k-\eta+1})^T \quad \cdots \quad (\mathbf{y}^{[i]}_k)^T\ ]^T$
5: **for** $j \in \mathcal{J}$ **do**
6: make prediction at a target point $(\mathbf{v}_j, \tau_j)$

$$\hat{z}_j^{[i]} = \mathbf{k}^T\mathbf{C}^{-1}\mathbf{y},$$

with a prediction error variance given by

$$\sigma^{2[i]}_{\hat{z}_j} = \sigma_f^2(1 - \mathbf{k}^T\mathbf{C}^{-1}\mathbf{k}),$$

where $\mathbf{y} = \mathbf{y}^{[i]}_{k-\eta+1:k}$, $\mathbf{k} = \mathrm{Corr}(\mathbf{y}, z_j)$, and $\mathbf{C} = \mathrm{Corr}(\mathbf{y}, \mathbf{y})$
7: **end for**
8: **if** $k \geq \eta$ **then**
9: discard the oldest set of measurements taken at time $t_{k-\eta+1}$, i.e., $\mathbf{y}^{[i]}_{k-\eta+1}$
10: **end if**
11: **while** $t_k \leq t \leq t_{k+1}$ **do**
12: compute $\nabla_{\mathbf{q}_\ell} J^{[i]}$ with the remained data $\mathbf{y}^{[i]}_{k-\eta+2}$
13: send $\nabla_{\mathbf{q}_\ell} J^{[i]}$ to agent ℓ in $\mathcal{N}_i(t)$
14: receive $\nabla_{\mathbf{q}_i} J^{[\ell]}$ from all neighbors in $\mathcal{N}_i(t)$
15: compute the gradient $\nabla_{\mathbf{q}_i} J_d = \sum_{\ell\in\mathcal{N}_i} \nabla_{\mathbf{q}_i} J^{[\ell]}/|\mathcal{I}|$
16: update position according to $\mathbf{q}_i(t+\delta t) = \mathbf{q}_i(t) - \alpha\nabla_{\mathbf{q}_i} J_d$ for a small step size α
17: **end while**

V. SIMULATION RESULTS

In this section, we apply our approach to a spatiotemporal Gaussian process with a covariance function in (3). The Gaussian process was numerically generated through circulant embedding of the covariance matrix for the simulation [35]. The hyperparameters used in the simulation were chosen to be $\boldsymbol{\theta} = [\sigma_f^2 \quad \sigma_x \quad \sigma_y \quad \sigma_t]^T = [1 \quad 0.2 \quad 0.2 \quad 5]^T$. The surveillance region $\mathcal{Q}$ is given by $\mathcal{Q} = [0 \quad 1]^2$. The SNR $\gamma = 100$ is used throughout the simulation, which is equivalent to a noise level of $\sigma_w = 0.1$. In our simulation, $N = 9$ agents start sampling at $t_1 = 1$ and make new observations at every integer time, i.e., $t_k = k\ \forall k \in \mathbb{Z}_{>0}$. The initial positions of the agents are randomly selected. The truncation size $\eta = 10$ is chosen by the use of the approach that is introduced in Section III-C that guarantees the averaged performance level $\epsilon(\eta = 10) < 0.1$ under a uniform sampling distribution (see Example 3.8).

In the figures of simulation results, the target positions, the initial positions of agents, the past sampling positions of agents, and the current positions of agents are represented by white stars, yellow crosses, pink dots, and white circles with agent indices, respectively.

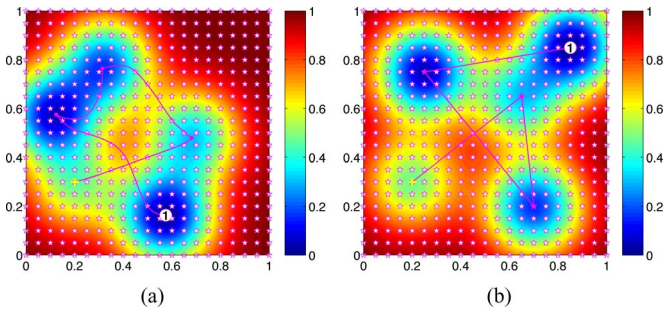


Fig. 5. Prediction error variances at t_5 achieved by the usage of (a) gradient-based algorithm and (b) exhaustive search algorithm. The trajectories of the agent are shown as solid lines.

A. Gradient-Based Algorithm Versus Exhaustive Search Algorithm

To evaluate the performance of the gradient-based algorithm that is presented in Section IV, we compare it with the exhaustive search algorithm over sufficiently many grid points, which guarantees the near optimum. Because of the exponential complexity of the grid-based exhaustive search algorithm as the number of agents increases, its usage for multiple robots is prohibitive. Hence, we consider a simple case in which only one mobile agent samples and makes prediction on 21×21 target points over $\mathcal{Q}$. The grid points used in the exhaustive search are the same as the target points, i.e., 21×21 grid points. The initial positions of the agents for both cases were set to $[0.2 \quad 0.3]^T$.

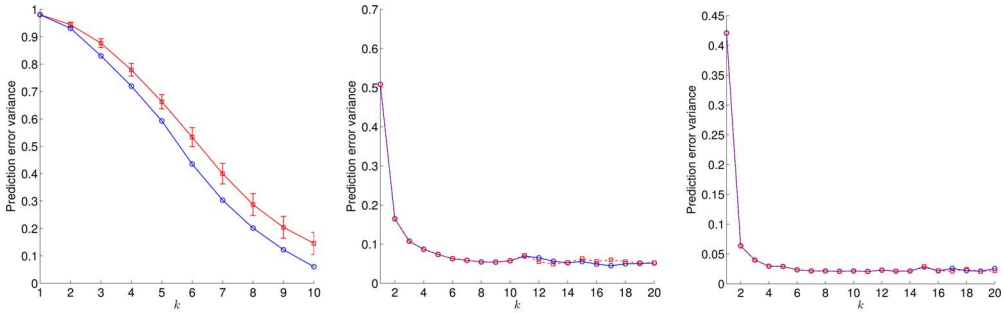


Fig. 6. Average of prediction error variances over target points (in blue circles) that is achieved by the centralized sampling scheme by the use of all collective observations for (a) case 1, (b) case 2, and (c) case 3. In case 1, the target points are fixed at time t_{10}, and the counterparts that are achieved by the benchmark random sampling strategy are shown as red squares with error bars. In cases 2 and 3, the target points are at t_{k+1} and change over time. The counterparts that are achieved by the use of truncated observations are shown as red squares.

The prediction error variances at t_5 for the proposed algorithm and the exhaustive search algorithm are shown in Fig. 5(a) and (b), respectively. At time t_5, the averaged prediction error variance over target points is 0.636, which is close to 0.613 that is achieved by the exhaustive search. Therefore, this simulation study shows that the performance of the gradient-based algorithm is comparable with that of the exhaustive search algorithm for the given problem.

B. Centralized Sampling Scheme

Consider a situation, where a central station has access to all measurements that are collected by agents. At each time, measurements that are sampled by agents are transmitted to the central station that uses the centralized navigation strategy and sends control commands back to individual agents.

Case 1: First, we consider a set of fixed target points, e.g., 6×6 grid points on $\mathcal{Q}$ at a fixed time t_{10}. At each time step, the cost function J_c in (11), which is the average of prediction error variances at target points, is minimized because of the proposed centralized navigation strategy in Section IV-A. As a benchmark strategy, we consider a random sampling scheme in which a group of nine agents takes observations at randomly selected positions within the surveillance region $\mathcal{Q}$.

In Fig. 6(a), the blue circles represent the average of prediction error variances over target points that are achieved by the centralized scheme, and the red squares indicate the average of prediction error variances over target points that are achieved by the benchmark strategy. Clearly, the proposed scheme produces lower averaged prediction error variances at target points as time increases, which demonstrates the usefulness of our scheme.

Case 2: Next, we consider the same 6×6 grid points on $\mathcal{Q}$ as in case 1. However, at time t_k, we are now interested in the prediction at the next sampling time t_{k+1}. At each time step, the cost function J_c is minimized. Fig. 6(b) shows the average of prediction error variances over target points that are achieved by the centralized scheme with truncation (red squares) and without truncation (blue circles). With truncated observations, i.e., with only observations that are obtained from latest $\eta = 10$ time steps, we are able to maintain the same level of the averaged prediction error variances [around 0.05 in Fig. 6(b)].

Fig. 7(a)–(c) shows the true field, the predicted field, and the prediction error variance at time t_1, respectively. To see the improvement, the counterparts of the simulation results at time t_5 are shown in Fig. 7(d)–(f). At time t_1, agents have little information about the field, and hence, the prediction is far away from the true field, which produces a large prediction error variance. As time increases, the prediction becomes close to the true field and the prediction error variances are reduced because of the proposed navigation strategy.

Case 3: Now, we consider another case in which 36 target points (which are plotted in Fig. 8 as white stars) are evenly distributed on three concentric circles to form a ring-shaped subregion of interest. As in case 2, we are interested in the prediction at the next time iteration t_{k+1}. The average of prediction error variances over these target points at each time step that is achieved by the centralized scheme with truncation (red squares) and without truncation (blue circles) are shown in Fig. 6(c). The prediction error variances at time t_1 and t_5 are shown in Fig. 8(a) and (b), respectively. It is shown that agents dynamically covered the ring-shaped region to minimize the average of prediction error variances over the target points.

C. Distributed Sampling Scheme

Consider a situation in which the sensor network has a limited communication range R, i.e., $\mathcal{N}_i(t) := \{j \in \mathcal{I} \,|\, \|\mathbf{q}_i(t) - \mathbf{q}_j(t)\| < R, j \neq i\}$. At each time step $k \in \mathbb{Z}_{>0}$, agent i collects measurements from itself and its neighbors $\mathcal{N}_i(t)$ and makes prediction in a distributed fashion. The distributed strategy is used to navigate itself to move to the next sampling position. To be comparable with the centralized scheme, the same target points as in case 2 of Section V-B are considered.

Fig. 9 shows that the cost function, which is an upper bound of the averaged prediction error variance over target points and agents, decreases smoothly from time t_1 to t_2 by the gradient descent algorithm with a communication range $R = 0.4$. Significant decreases occur whenever one of the agent gains a neighbor. Notice that the discontinuity of minimization of J in (15) that is caused by gaining or losing neighbors is eliminated because of the construction of J_d in (18). Hence, the proposed distributed algorithm is robust to gaining or losing neighbors.

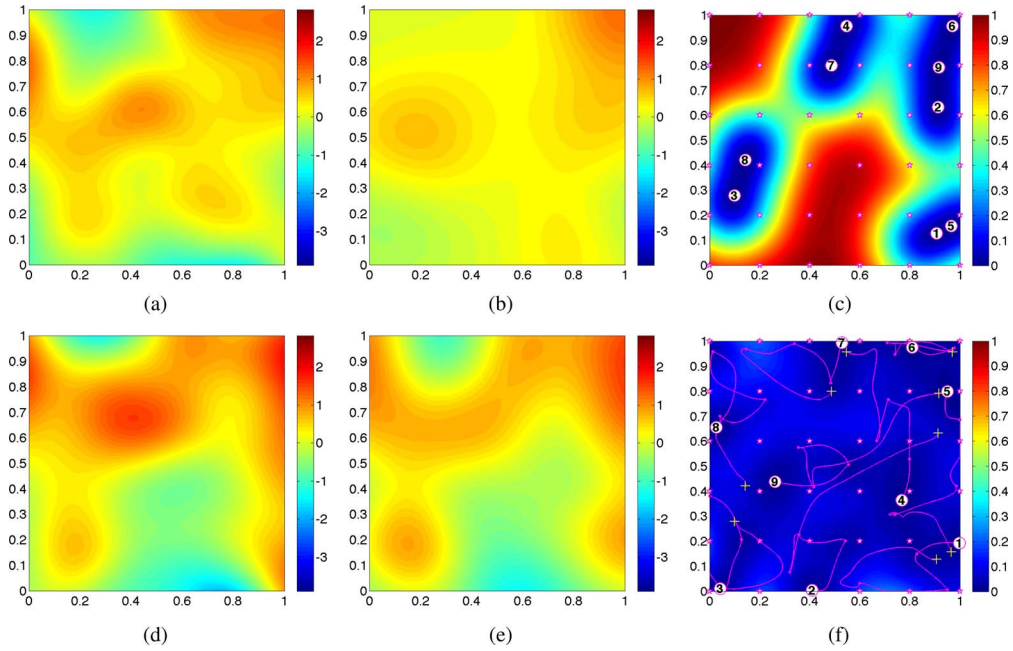


Fig. 7. Simulation results at t_1 and t_5 that are obtained by the centralized sampling scheme for case 2. (a) True field at t_1. (b) Predicted field at t_1. (c) Prediction error variance at t_1. (d) True field at t_5. (e) Predicted field at t_5. (f) Prediction error variance at t_5.

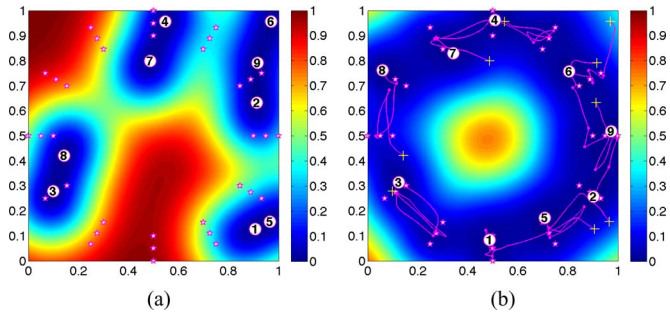


Fig. 8. Simulation results that are obtained by the centralized sampling scheme for case 3. The trajectories of agents are shown as solid lines. (a) Prediction error variance at t_1. (b) Prediction error variance at t_5.

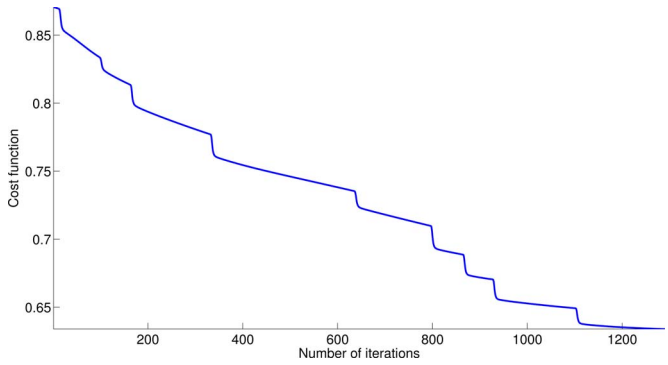


Fig. 9. Cost function $J_d(\tilde{\mathbf{q}})$ from t_1 to t_2 with a communication range $R = 0.4$.

The following study shows the effect of different communication ranges. Intuitively, the larger the communication range is, the more information can be obtained by the agent, and hence, the better prediction can be made. Fig. 10(a) and (b) shows the

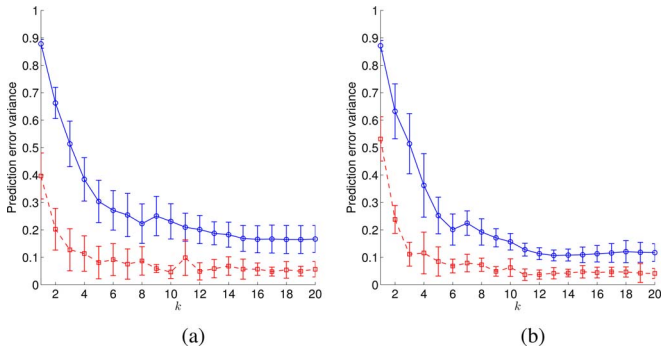


Fig. 10. Average of prediction error variances over all target points and agents that are achieved by the distributed sampling scheme with a communication range (a) $R = 0.3$ and (b) $R = 0.4$. The average of prediction error variances over all target points, and agents are shown as blue circles. The average of prediction error variance over local target points and agents are shown as red squares. The error bars indicate the standard deviation among agents.

average of prediction error variances over all target points and agents as blue circles with error bars indicating the standard deviation among agents for the cases $R = 0.3$ and $R = 0.4$, respectively. In both cases, $d_0 = 0.1$ in (17) was used. The average of prediction error variances is minimized quickly to a certain level. It can be seen that the level of the achieved averaged prediction error variance with $R = 0.4$ is lower than the counterpart with $R = 0.3$.

Now, assume that each agent only predict the field at target points within radius R (local target points). The average of prediction error variances, over only local target points and agents, are also plotted in Fig. 10 as red squares with the standard deviation among agents. As can be seen, the prediction error variances at local target points (red squares) are significantly lower than those for all target points (blue circles).

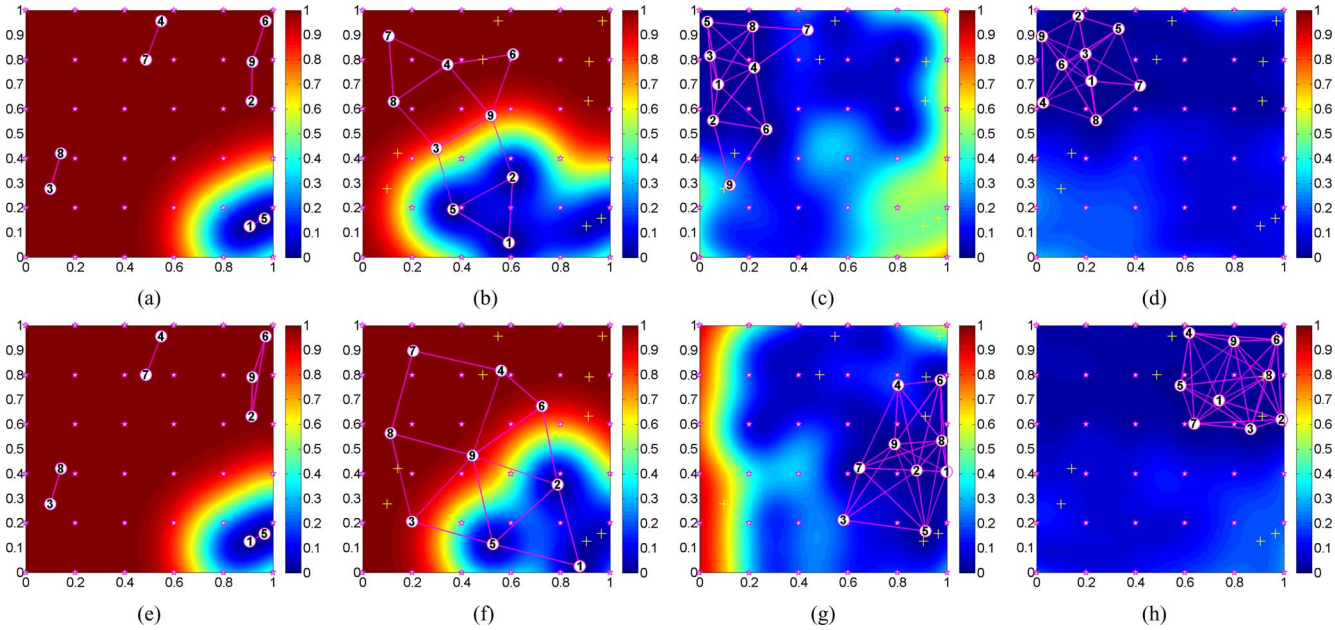


Fig. 11. Simulation results are obtained by the distributed sampling scheme with different communication ranges. The edges of the graph are shown as solid lines. (a) $R = 0.3$, $k = 1$. (b) $R = 0.3$, $k = 2$. (c) $R = 0.3$, $k = 5$. (d) $R = 0.3$, $k = 20$. (e) $R = 0.4$, $k = 1$. (f) $R = 0.4$, $k = 2$. (g) $R = 0.4$, $k = 5$. (h) $R = 0.4$, $k = 20$.

Fig. 11 shows the prediction error variances that are obtained by agent 1 along with the edges of the communication network for different communication ranges R and different time steps k. In Fig. 11, the target positions, the initial positions, and the current positions are represented by white stars, yellow crosses, and white circles, respectively. Surprisingly, the agents under the distributed navigation algorithm produce an emergent, swarm-like behavior to maintain communication connectivity among local neighbors. Notice that this collective behavior emerged naturally and was not generated by the flocking or swarming algorithm as in [8]. This interesting simulation study (see Fig. 11) shows that agents will not get too close to each other since the average of prediction error variances at target points can be reduced by spreading over and covering the target points that need to be sampled. However, agents will not move too far away from each other since the average of prediction error variances can be reduced by collecting measurements from a larger population of neighbors. This tradeoff is controlled by the communication range. With the intertwined dynamics of agents over the proximity graph, as shown in Fig. 11, mobile sensing agents are coordinated in each time iteration in order to dynamically cover the target positions for better collective prediction capability.

VI. Conclusion

For spatiotemporal Gaussian processes, in this paper, prediction based on truncated observations for mobile sensor networks has been justified. In particular, a theoretical foundation of Gaussian processes with truncated observations have been presented. Centralized and distributed navigation strategies have been proposed to minimize the average of prediction error variances at target points that can be arbitrarily chosen by a user. Simulation results demonstrated that mobile sensing agents under the distributed navigation strategy produce an emergent, collective behavior for communication connectivity and are coordinated to improve the quality of the collective prediction capability. Future work will consider the optimal coordination of the mobile sensor networks subject to energy constraints.

Appendix A
Proof of Theorem 3.1

Proof: We can rewrite (4a) as

$$\hat{z}_* = \begin{bmatrix} \mathbf{k}_m \\ \mathbf{k}_r \end{bmatrix}^T \begin{bmatrix} \mathbf{C}_m & \mathbf{K}_{mr} \\ \mathbf{K}_{mr}^T & \mathbf{C}_r \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{y}_m \\ \mathbf{y}_r \end{bmatrix} \tag{23a}$$

and (4b) as

$$\sigma_{\hat{z}_*}^2 = \sigma_f^2 \left(1 - \begin{bmatrix} \mathbf{k}_m \\ \mathbf{k}_r \end{bmatrix}^T \begin{bmatrix} \mathbf{C}_m & \mathbf{K}_{mr} \\ \mathbf{K}_{mr}^T & \mathbf{C}_r \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{k}_m \\ \mathbf{k}_r \end{bmatrix} \right). \tag{23b}$$

By the use of the identity that is based on matrix-inversion lemma

$$\begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{B}^T & \mathbf{C} \end{bmatrix}^{-1} = \begin{bmatrix} \mathbf{E} & \mathbf{F} \\ \mathbf{F}^T & \mathbf{G} \end{bmatrix}$$

where

$$\mathbf{E} = \mathbf{A}^{-1} + \mathbf{A}^{-1}\mathbf{B}\mathbf{C} - \mathbf{B}^T\mathbf{A}^{-1}\mathbf{B})^{-1}\mathbf{B}^T\mathbf{A}^{-1}$$
$$\mathbf{F} = -\mathbf{A}^{-1}\mathbf{B}(\mathbf{C} - \mathbf{B}^T\mathbf{A}^{-1}\mathbf{B})^{-1}$$
$$\mathbf{G} = (\mathbf{C} - \mathbf{B}^T\mathbf{A}^{-1}\mathbf{B})^{-1}$$

(23a) and (23b), respectively, become

$$\hat{z}_* = \mathbf{k}_m^T\mathbf{C}_m^{-1}\mathbf{y}_m + (\mathbf{k}_r - \mathbf{K}_{mr}^T\mathbf{C}_m^{-1}\mathbf{k}_m)^T$$

$$\times (\mathbf{C}_r - \mathbf{K}_{mr}^T \mathbf{C}_m^{-1} \mathbf{K}_{mr})^{-1}$$
$$\times (\mathbf{y}_r - \mathbf{K}_{mr}^T \mathbf{C}_m^{-1} \mathbf{y}_m)$$

and

$$\begin{aligned}\sigma_{\hat{z}_*}^2 = \sigma_f^2 \left(1 - \mathbf{k}_m^T \mathbf{C}_m^{-1} \mathbf{k}_m\right) \\ - \sigma_f^2 (\mathbf{k}_r - \mathbf{K}_{mr}^T \mathbf{C}_m^{-1} \mathbf{k}_m)^T \\ \times (\mathbf{C}_r - \mathbf{K}_{mr}^T \mathbf{C}_m^{-1} \mathbf{K}_{mr})^{-1} \\ \times (\mathbf{k}_r - \mathbf{K}_{mr}^T \mathbf{C}_m^{-1} \mathbf{k}_m).\end{aligned}$$

Hence, by the use of (5a) and (5b), we obtain (6a) and (6b). ■

APPENDIX B
PROOF OF THEOREM 3.3

Proof: The first statement is straightforward from Theorem 3.1.

For the second statement, we can represent $\mathbf{y}_r$ as $\mathbf{y}_r = \mathbf{C}_r^{1/2}\mathbf{u} = \mathbf{U}\mathbf{\Lambda}^{1/2}\mathbf{u} = \mathbf{U}\tilde{\mathbf{y}}$, where $\mathbf{u}$ is a vector of independent standard normals, and $\mathbf{C}_r = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^T$ and $\mathbf{C}_r^{1/2} = \mathbf{U}\mathbf{\Lambda}^{1/2}$. By the usage of the Cauchy–Schwarz inequality and norm inequalities, we have

$$\begin{aligned}|\hat{z}_* - \hat{z}'_*| &= |\mathbf{k}_r^T \mathbf{C}_r^{-1} \mathbf{y}_r| \\ &= |\mathbf{k}_r^T \mathbf{C}_r^{-1} \mathbf{U}\tilde{\mathbf{y}}| \\ &\le \|\mathbf{k}_r^T \mathbf{C}_r^{-1}\| \|\mathbf{U}\tilde{\mathbf{y}}\| \\ &= \|\mathbf{k}_r^T \mathbf{C}_r^{-1}\| \|\tilde{\mathbf{y}}\| \\ &\le \|\mathbf{k}_r^T \mathbf{C}_r^{-1}\| \sqrt{r} \|\tilde{\mathbf{y}}\|_\infty \\ &\le \|\mathbf{k}_r^T \mathbf{C}_r^{-1}\| \sqrt{r} \bar{y}.\end{aligned}$$

Recall that we have $\mathbf{u} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ and $\tilde{\mathbf{y}} \sim \mathcal{N}(\mathbf{0}, \mathbf{\Lambda})$, where $\mathbf{\Lambda} = \mathrm{diag}(\lambda_1, \ldots, \lambda_r)$. Then, we can compute the probability $p_1 = \Pr(\|\tilde{\mathbf{y}}\|_\infty \le \bar{y})$ as follows:

$$\begin{aligned}p_1 &= \Pr\left(\max_{1\le i\le r} |\tilde{y}^{(i)}| \le \bar{y}\right) \\ &= \Pr\left(\max_{1\le i\le r} |\lambda_i^{1/2} u_i| \le \bar{y}\right) \\ &= \prod_{1\le i\le r} \Pr\left(\lambda_i^{1/2} |u_i| \le \bar{y}\right) \\ &= \prod_{1\le i\le r} \Pr\left(|u_i| \le \frac{\bar{y}}{\lambda_i^{1/2}}\right) \\ &= \prod_{1\le i\le r} \left(1 - 2\phi\left(-\frac{\bar{y}}{\lambda_i^{1/2}}\right)\right)\end{aligned}$$

where ϕ is the cumulative standard normal distribution. ■

APPENDIX C
PROOF OF THEOREM 3.4

Proof: Let $\mathbf{A} = \mathbf{C}_m^{-1}\mathbf{K}_{mr}$ and $\mathbf{B} = \mathbf{K}_{mr}^T \mathbf{C}_m^{-1} \mathbf{K}_{mr}$ for notational convenience. Then

$$\begin{aligned}|\hat{z}_* - \hat{z}'_*| &= \|(\mathbf{k}_r^T - \mathbf{k}_m^T \mathbf{A})(\mathbf{C}_r - \mathbf{B})^{-1}(\mathbf{y}_r - \mathbf{A}^T \mathbf{y}_m)\| \\ &\le \|\mathbf{k}_r^T - \mathbf{k}_m^T \mathbf{A}\| \|(\mathbf{C}_r - \mathbf{B})^{-1}(\mathbf{y}_r - \mathbf{A}^T \mathbf{y}_m)\|.\end{aligned}$$

Since $\mathbf{K}_r = \mathbf{C}_r - 1/\gamma \mathbf{I}$ is positive semidefinite and $\mathbf{C}_m$ is positive definite, we have that $\mathbf{K}_r - \mathbf{B}$ is positive semidefinite. Then, we have

$$\begin{aligned}(\mathbf{C}_r - \mathbf{B})^{-1} &= (\mathbf{K}_r + 1/\gamma \mathbf{I} - \mathbf{B})^{-1} \\ &\preceq \gamma \mathbf{I}.\end{aligned}$$

Combining this result, we get

$$\begin{aligned}|\hat{z}_* - \hat{z}'_*| &\le 2\gamma \|\mathbf{k}_r\| (\|\mathbf{y}_r\| + \|\mathbf{A}^T \mathbf{y}_m\|) \\ &\le 2\gamma(1+\delta_2) \|\mathbf{k}_r\| \|\mathbf{y}_r\| \\ &\le 2\gamma(1+\delta_2) \sqrt{r} \mathcal{K}_{\max} \|\mathbf{y}_r\|\end{aligned}$$

where $\mathcal{K}(\mathbf{x}^{(i)}, \mathbf{x}_*) \le \mathcal{K}_{\max}$ for $i \in \{m+1, \ldots, n\}$. Define $\bar{y}(p_2)$ such that $\max_{i=m+1}^n |y^{(i)}| \le \bar{y}(p_2)$ with probability p_2. Then, with probability p_2, we have

$$|\hat{z}_* - \hat{z}'_*| \le 2\gamma r (1+\delta_2) \mathcal{K}_{\max} \bar{y}(p_2).$$

Hence, for $\epsilon > 0$, if

$$\mathcal{K}_{\max} < \frac{\epsilon}{2\gamma r (1+\delta_2) \bar{y}(p_2)} \tag{24}$$

with probability p_2, we have

$$|\hat{z}_* - \hat{z}'_*| < \epsilon. \tag{25}$$

Let $l^2 = \min \|\mathbf{x}^{(i)} - \mathbf{x}_*\|^2$ for any $i \in \{m+1, \ldots, n\}$. Then, (24) becomes, with probability p_2

$$\exp\left(-\frac{l^2}{2\sigma_\ell^2}\right) \le \mathcal{K}_{\max} < \frac{\epsilon}{2\gamma r (1+\delta_2) \bar{y}(p_2)}$$
$$l^2 > -2\sigma_\ell^2 \log\left(\frac{\epsilon}{2\gamma r (1+\delta_2) \bar{y}(p_2)}\right).$$

For $\epsilon < 2\gamma r (1+\delta_2) \bar{y}(p_2)$, we have

$$l^2 > 2\sigma_\ell^2 \log\left(2\gamma \frac{1}{\epsilon} r (1+\delta_2) \bar{y}(p_2)\right)$$

and this completes the proof. ■

ACKNOWLEDGMENT

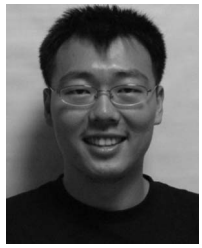
The authors would like to thank the associate editor and the anonymous reviewers for their valuable comments and suggestions.

REFERENCES

[1] M. M. Zavlanos and G. J. Pappas, "Distributed connectivity control of mobile networks," *IEEE Trans. Robot.*, vol. 24, no. 6, pp. 1416–1428, Dec. 2007.

[2] K. M. Lynch, I. B. Schwartz, P. Yang, and R. A. Freeman, "Decentralized environmental modeling by mobile sensor networks," *IEEE Trans. Robot.*, vol. 24, no. 3, pp. 710–724, Jun. 2008.

[3] M. M. Zavlanos and G. J. Pappas, "Dynamic assignment in distributed motion planning with local coordination," *IEEE Trans. Robot.*, vol. 24, no. 1, pp. 232–242, Feb. 2008.
[4] A. Jadbabaie, J. Lin, and A. S. Morse, "Coordination of groups of mobile autonomous agents using nearest neighbor rules," *IEEE Trans. Automat. Control*, vol. 48, no. 6, pp. 988–1001, Jun. 2003.
[5] R. Olfati-Saber, "Flocking for multi-agent dynamic systems: Algorithms and theory," *IEEE Trans. Automat. Control*, vol. 51, no. 3, pp. 401–420, Mar. 2006.
[6] W. Ren and R. W. Beard, "Consensus seeking in multiagent systems under dynamically changing interaction topologies," *IEEE Trans. Automat. Control*, vol. 50, no. 5, pp. 655–661, May 2005.
[7] N. E. Leonard, D. A. Paley, F. Lekien, R. Sepulchre, D. M. Fratantoni, and R. E. Davis, "Collective motion, sensor networks, and ocean sampling," *Proc. IEEE*, vol. 95, no. 1, pp. 48–74, Jan. 2007.
[8] J. Choi, S. Oh, and R. Horowitz, "Distributed learning and cooperative control for multi-agent systems," *Automatica*, vol. 45, no. 12, pp. 2802–2814, 2009.
[9] N. Cressie, *Statistics for Spatial Data*. New York: Wiley, 1991.
[10] A. Krause, A. Singh, and C. Guestrin, "Near-optimal sensor placements in Gaussian processes: Theory, efficient algorithms and empirical studies," *J. Mach. Learning Res.*, vol. 9, pp. 235–284, 2008.
[11] A. Singh, A. Krause, C. Guestrin, W. Kaiser, and M. Batalin, "Efficient planning of informative paths for multiple robots," in *Proc. Int. Joint Conf. Artif. Intell.*, 2007, pp. 2204–2211.
[12] N. Cressie, "Kriging nonstationary data," *J. Amer. Statist. Assoc.*, vol. 81, no. 395, pp. 625–634, 1986.
[13] M. Gibbs and D. J. C. MacKay. (1997). Efficient implementation of Gaussian processes. [Online]. Available: http://www.cs.toronto.edu/mackay/gpros.ps.gz
[14] D. J. C. Mackay, "Introduction to Gaussian processes," *Neural Netw. Mach. Learning*, vol. 168, pp. 133–165, 1998.
[15] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*. New York: Springer-Verlag, 2006.
[16] A. Krause, C. Guestrin, A. Gupta, and J. Kleinberg, "Near-optimal sensor placements: Maximizing information while minimizing communication cost," in *Proceedings of the 5th International Conference on Information Processing in Sensor Networks*. New York: ACM, 2006, pp. 2–10.
[17] J. Cortés, "Distributed kriged Kalman filter for spatial estimation," *IEEE Trans. Automat. Control*, vol. 54, no. 12, pp. 2816–2827, Dec. 2009.
[18] J. Choi, J. Lee, and S. Oh, "Biologically-inspired navigation strategies for swarm intelligence using spatial Gaussian processes," in *Proc. Int. Fed. Automat. Control World Congr.*, 2008, pp. 593–598.
[19] J. Choi, J. Lee, and S. Oh, "Swarm intelligence for achieving the global maximum using spatio-temporal Gaussian processes," in *Proc. Amer. Control Conf.*, 2008, pp. 135–140.
[20] K. V. Mardia, C. Goodall, E. J. Redfern, and F. J. Alonso, "The kriged Kalman filter," *Test*, vol. 7, no. 2, pp. 217–282, 1998.
[21] N. Cressie and C. K. Wikle, "Space-time Kalman filter," *Encyclopedia Environmetrics*, vol. 4, pp. 2045–2049, 2002.
[22] J. Cortés, "Discontinuous dynamical systems," *IEEE Control Syst. Mag.*, vol. 28, no. 3, pp. 36–73, Jun. 2008.
[23] Y. Xu and J. Choi, "Adaptive sampling for learning Gaussian processes using mobile sensor networks," *Sensors*, vol. 11, no. 3, pp. 3051–3066, 2011.
[24] A. J. Smola and P. L. Bartlett, "Sparse greedy Gaussian process regression," in *Advances in Neural Information Processing Systems 13*. Cambridge, MA: MIT Press, 2001, pp. 619–625.
[25] C. Williams and M. Seeger, "Using the Nystrom method to speed up kernel machines," in *Advances in Neural Information Processing Systems 13*. Cambridge, MA: MIT Press, 2001, pp. 682–688.
[26] N. Lawrence, M. Seeger, and R. Herbrich, "Fast sparse Gaussian process methods: The informative vector machine," in *Proc. Adv. Neural Inf. Process. Syst.*, 2003, pp. 625–632.
[27] M. Seeger, "Bayesian Gaussian process models: PAC-Bayesian generalisation error bounds and sparse approximations," Ph.D. dissertation, School of Informatics, Univ. Edinburgh, Edinburgh, U.K., 2003.
[28] V. Tresp, "A Bayesian committee machine," *Neural Comput.*, vol. 12, pp. 2719–2741, 2000.
[29] C. Williams and F. Vivarelli, "Upper and lower bounds on the learning curve for Gaussian processes," *Mach. Learning*, vol. 40, no. 1, pp. 77–102, 2000.
[30] P. Sollich and A. Halees, "Learning curves for Gaussian process regression: Approximations and bounds," *Neural Comput.*, vol. 14, no. 6, pp. 1393–1428, 2002.
[31] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. New York: Wiley-Interscience, 2006.
[32] D. P. Bertsekas, W. W. Hager, and O. L. Mangasarian, *Nonlinear Programming*. Belmont, MA: Athena Scientific, 1999.
[33] G. Mathews, H. Durrant-Whyte, and M. Prokopenko, "Decentralised decision making in heterogeneous teams using anonymous optimisation," *Robot. Auton. Syst.*, vol. 57, no. 3, pp. 310–320, 2009.
[34] W. Rudin, *Principles of Mathematical Analysis*. New York: McGraw-Hill, 1976.
[35] C. Dietrich and G. Newsam, "Fast and exact simulation of stationary Gaussian processes through circulant embedding of the covariance matrix," *SIAM J. Sci. Comput.*, vol. 18, no. 4, pp. 1088–1107, 1997.



Yunfei Xu (S'09–M'11) received the M.S. and B.S. degrees in automotive engineering from Tsinghua University, Beijing, China, in 2004 and 2007, respectively. Currently, he is working toward the Ph.D. degree with the Department of Mechanical Engineering, Michigan State University, East Lansing.

His current research interests include environmental adaptive sampling algorithms, Gaussian processes, and statistical learning algorithms with applications to robotics and mobile sensor networks.

Mr. Xu is a student member of the American Society of Mechanical Engineers.

Jongeun Choi (S'05–M'06) received the B.S. degree in mechanical design and production engineering from Yonsei University, Seoul, Korea, in 1998 and the Ph.D. and M.S. degrees in mechanical engineering from the University of California, Berkeley, in 2006 and 2002, respectively.

He is currently an Assistant Professor with the Department of Mechanical Engineering and the Department of Electrical and Computer Engineering, Michigan State University, East Lansing. His research interests include adaptive, distributed, and robust control and statistical learning algorithms, with applications to self-organizing systems, mobile robotic sensors, environmental adaptive sampling, and biomedical problems.

Dr. Choi was a recipient of the National Science Foundation CAREER Award in 2009. He is a member of the American Society of Mechanical Engineers.

Songhwai Oh (S'04–M'07) received the B.S. (Hons.), M.S., and Ph.D. degrees in electrical engineering and computer sciences (EECS) from the University of California, Berkeley, in 1995, 2003, and 2006, respectively.

He is an Assistant Professor with the School of Electrical Engineering and Computer Science, Seoul National University, Seoul, Korea. Before his Ph.D. studies, he was a Senior Software Engineer with Synopsys, Inc. and a Microprocessor Design Engineer with Intel Corporation. In 2007, he was a Postdoctoral Researcher with the Department of EECS, University of California. From 2007 to 2009, he was an Assistant Professor of EECS with the School of Engineering, University of California, Merced. His research interests include cyber-physical systems, wireless sensor networks, robotics, estimation and control of stochastic systems, multimodal sensor fusion, and machine learning.