Jongeun Choi[1]
Associate Professor
Mem. ASME
Department of Mechanical Engineering,
Department of Electrical and
Computer Engineering,
Michigan State University,
East Lansing, MI 48864
e-mail: jchoi@egr.msu.edu

Dejan Milutinović
Associate Professor
Mem. ASME
Computer Engineering Department,
University of California,
Santa Cruz,
Santa Cruz, CA 95064
e-mail: dejan@soe.ucsc.edu

Tips on Stochastic Optimal Feedback Control and Bayesian Spatiotemporal Models: Applications to Robotics

This tutorial paper presents the expositions of stochastic optimal feedback control theory and Bayesian spatiotemporal models in the context of robotics applications. The presented material is self-contained so that readers can grasp the most important concepts and acquire knowledge needed to jump-start their research. To facilitate this, we provide a series of educational examples from robotics and mobile sensor networks.
[DOI: 10.1115/1.4028642]

1 Introduction

Since the very beginning, optimal control theory has been developed around concepts such as dynamic programming [1], the Hamilton–Jacobi–Bellman (HJB) equation [2], and Pontryagin's minimum principle [3]. These concepts continue to be of fundamental importance for stochastic optimal control [4–6] as well. For continuous-time systems, the concept of the HJB equation for deterministic systems naturally extends to stochastic differential equations (SDEs) [7], or piecewise deterministic systems [8], as well as stochastic hybrid systems [9]. Solutions of the HJB equation are generally nonsmooth, and the theory of viscosity solutions [10] has to be used. However, the dynamic programming, which is directly applicable only to control problems involving discrete decision steps and discrete state space, can also be used for the numerical evaluation of the HJB equation [5]. When there is an optimal feedback control, it depends on the value function defined over the state space. The evaluation of the value function can be formulated as the estimation problem of a spatiotemporal scalar field over the state space, which can be solved by using the Gaussian process approximation of the value function [11–13].

However, the estimation of a spatiotemporal field has been a topic of interest in its own right, especially with an increasing exploitation of mobile robot sensor networks interacting with uncertain environments [14–21]. To tackle a variety of tasks such as exploration, estimation, prediction, and maximum seeking of a scalar field, it is essential to deal with spatial models of various physical fields. Computationally demanding, physics-based field models (e.g., atmospheric modeling [22]) have been developed. Recently, phenomenological and statistical modeling techniques such as kriging, kernel regression, Gaussian process regression, and Gaussian Markov random fields (GMRFs) have gained much attention for resource-constrained mobile robots [14–18,23–29].

The outline of the paper is given as follows: First, we give technical preliminaries on SDEs in Sec. 2. In Sec. 3, we present two stochastic optimal control problems for robotic vehicles. With each problem, we provide the process involved in computing the optimal control and its results. In Sec. 4, we give an exposition of some recent developments on spatiotemporal models for sensor network applications ranging from a simple parametric model to a fully Bayesian approach [16–18,26–28].

Note that this paper is based on the previous works by the authors and is not meant to be a comprehensive review of the topics of interest. The aim of the paper is to provide a self-contained material on the application of stochastic processes in robotics so that readers can grasp the most important concepts and acquire the knowledge necessary to jump-start their research.

2 Langevin Equation and Îto Integrals

SDEs are those in which derivatives describing state evolutions depend on stochastic processes. In the most general form, we can write them as $\dot{x} = f(x, \xi(t), t)$, where $x \in \mathbb{R}^m$ is a state, $\xi(t) \in \mathbb{R}^m$ is a stochastic process, and $t \in \mathbb{R}^1$ denotes the time. Among all variants of SDEs, the type that is commonly used in the Kalman filter theory to describe the dynamics is

$$\frac{dx}{dt} = a(x,t) + b(x,t)\xi(t) \tag{1}$$

where $a(x, t)$ and $b(x, t)$ are nonlinear functions, i.e., mappings of appropriate dimensions. This type of SDEs is also called the Langevin equation. In this equation, $\xi(t)$ is the so-called process noise and is considered to be the zero-mean, unit intensity white noise, i.e.,

$$\mathbb{E}\{\xi(t)\} = 0 \quad \text{and} \quad \mathbb{E}\{\xi(t_i)\xi(t_j)\} = I_{m\times m}\delta(t_i - t_j) \tag{2}$$

where $\mathbb{E}$ denotes the expected value, $I_{m\times m}$ is the unity matrix of dimension $m\times m$, t_i and t_j denote two arbitrary time points, and the function $\delta(t)$ is the Dirac delta function, which means that the process $\xi(t)$ has a covariance matrix $\mathbb{E}\{\xi(t)^2\} = I_{m\times m}\delta(0)$ that is not bounded.

To deal with the white noise, we multiply the Langevin equation (1) by dt and rewrite it in a more convenient form as

$$dx = a(x,t)dt + b(x,t)dw \tag{3}$$

where $dw(t) = \xi(t)dt$ is an increment of the Wiener process $w(t)$ at time point t, i.e., $dw(t) := w(t + dt) - w(t)$, but the symbol t is usually omitted. Increments of the Wiener process are independent, with the zero mean and the variance

$$\mathbb{E}\left\{dw(t)^2\right\} = I_{m\times m}dt \tag{4}$$

[1]Corresponding author.

Contributed by the Dynamic Systems Division of ASME for publication in the Journal of Dynamic Systems, Measurement, and Control. Manuscript received February 14, 2014; final manuscript received September 22, 2014; published online October 21, 2014. Assoc. Editor: J. Karl Hedrick.

The solution of Eq. (3) can be expressed as a sum of two integral terms

$$x(t) = x(t_0) + \int_{t_0}^{t} a(x,\tau)d\tau + \int_{t_0}^{t} b(x,\tau)dw \tag{5}$$

where τ is the time variable used in the integration and $d\tau$ is its increment. Note that the solution $x(t)$ is a stochastic process and that the first integral includes only $a(x, t)$, which is stochastic, while in the second integral, both $b(x, \tau)$ and dw are stochastic. The solution $x(t)$ can be approximated as

$$x(t) \approx x(t_0) + \sum_{k=0}^{N-1} a(x,\tau_k)\Delta t + \sum_{k=0}^{N-1} b(x,\tau_k)\Delta w_k \tag{6}$$

with $\tau_{k+1} - \tau_k = \Delta t, \tau_k \in [t_k, t_{k+1}], \Delta w_k = w(t_{k+1}) - w(t_k)$ and

$$t_k = t_0 + k\Delta t \quad \text{and} \quad \Delta t = \frac{t - t_0}{N} \tag{7}$$

The sums defining the solution depend on τ_i, which is anywhere in the interval $[t_k, t_{k+1}]$. If the sampling points are chosen to be $\tau_k = t_k$, where $k = 0, 1, \ldots, N-1$, we deal with Îto stochastic integrals and the corresponding Îto calculus. Equation (6) can be rewritten in an iterative form as

$$x(t_k{+}1) \approx x(t_k) + a(x,t_k)\Delta t + b(x,t_k)\Delta w_k \tag{8}$$

which approximates the solution of Eq. (3) under the assumptions of Îto calculus and where $\Delta w \sim \mathcal{N}(0, \Delta t I_{m\times m})$ is an m-dimensional zero mean value Gaussian random vector with a covariance matrix $\Delta t\, I_{m\times m}$. The latter can be illustrated with the following scalar case ($m = 1$) of the SDE in Eq. (3).

When $m = 1$, $a(x, t) = 0$, and $b(x, t) = 1$, the SDE in Eq. (3) is $dx = dw$, or, in other words, the solution $x(t)$ is the one-dimensional Wiener process, i.e., $x(t) = w(t)$. Thus, the solution can be expressed by a simple stochastic integral and its approximation as

$$w(t) = \int_{t_0}^{t} dw \approx w_N(t) = w_0 + \sum_{k=0}^{N-1} \Delta w_k \tag{9}$$

where $w_0 = w(t_0)$ is the value of $w(t)$ at the time t_0 and the usual assumption is that $w(t_0) = 0$. Since $w(t)$ is a stochastic process, its value, as well as its approximation at any time point $w_N(t)$ is random, but we can find its statistical properties. The expected value $\overline{w}_N$ and the variance $\sigma^2_{w_N}$ of w_N are given, respectively, by

$$\begin{aligned} \overline{w}_N &= \mathbb{E}\{w_N(t)\} = w_0 \\ \sigma^2_{w_N} &= \mathbb{E}\left\{(w_N(t) - w_0)^2\right\} = N\Delta t = t - t_0 \end{aligned} \tag{10}$$

In the limit of $\Delta t \to 0$, we have in Eq. (9) an infinite sum; therefore, based on the central limit theorem, we can say that the distribution of $w(t)$ is Gaussian, i.e.,

$$w(t) \sim \mathcal{N}(w_0, t - t_0) = \frac{\exp\left(-\frac{1}{2}\frac{(w(t) - w_0)^2}{(t - t_0)}\right)}{\sqrt{2\pi(t - t_0)}} \tag{11}$$

Since the process $w(t)$, which is the solution of the stochastic integral $\int_0^t dw$, is random, this Gaussian distribution is a complete description of the solution $w(t)$, i.e., the Wiener process. One way of dealing with the solution of integrals, including random increments dw is to represent the solution in the form of differential equations, where the Îto calculus chain rule is very useful.

The Îto calculus chain rule can be briefly derived based on the following rules. First, we have to use the second-order Taylor expansion of $f(x)$. Second, if we have terms $(dw)^2$, we have to substitute them with dt. Finally, every term of the form dt^p with $p > 1$ should be ignored, i.e., considered to be zero. In the case of $f(x)$: $\mathbb{R} \to \mathbb{R}$ and dx defined by Eq. (3), and a, b: $\mathbb{R} \times \mathbb{R} \to \mathbb{R}$, the first rule yields

$$df(x) = \frac{\partial f(x)}{\partial x}dx + \frac{1}{2}\frac{\partial^2 f(x)}{\partial x^2}dx^2 \tag{12}$$

After the substitutions of dx from Eq. (3) and $(dw)^2 = dt$, we have

$$df(x) = \left(\frac{\partial f(x)}{\partial x}a + \frac{1}{2}\frac{\partial^2 f(x)}{\partial x^2}b^2\right)dt + \frac{\partial f(x)}{\partial x}b\,dw \tag{13}$$

The probability density function of the multidimensional Wiener process $w(t) \in \mathbb{R}^m$ is

$$w(t) \propto \frac{\exp\left(-\frac{1}{2}(w(t) - w_0)^T [I_{m\times m}(t - t_0)]^{-1}(w(t) - w_0)\right)}{\sqrt{2\pi(t - t_0)^N}} \tag{14}$$

where the multivariate Gaussian distribution on the right side can be also written as $\mathcal{N}(w_0, I_{m\times m}(t - t_0))$. The multidimensional Wiener process and its increments are associated with the multidimensional SDE in Eq. (3) and, in that case, the Îto chain rule is

$$df(x) = \left(\frac{\partial f^T}{\partial x}a(x(t),(t)) + \frac{1}{2}\mathrm{tr}\left\{\frac{\partial^2 f}{\partial x^2}bb^T\right\}\right)dt + \frac{\partial f}{\partial x}b(x(t),t)dw$$

where $\mathrm{tr}\{A\}$ is the trace of A. With this, we have covered all technical preliminaries that are necessary for a basic understanding of SDE models and for considering the optimal control of SDEs.

3 Stochastic Approach to Robot Control

In this section, we describe elements of stochastic optimal control with applications to robotics. To keep examples nonlinear and realistic, yet simple enough to be educational, we use two-wheeled robot control problems similar to Refs. [30] and [31] that can be easily replicated in laboratories. We model the robot as a fixed velocity Dubins vehicle, and the only control variable is the limited robot turning rate. This type of vehicles is nonlinear, nonholonomic, and underactuated. It is also a good model for autopilot-controlled unmanned aerial vehicles [32].

The control problems we consider are illustrated in Fig. 1. In the first one (P1), the robot has to navigate to a point in the space in a minimum expected time. In the second problem (P2), the robot vehicle has to keep a constant distance from a moving target, while the target trajectory is unknown. These two problems are different not only in the cost functions, but in the way that random processes enter the models.

The stochastic component in (P1) is representative of the anticipated uncertainty of our control; for example, it results from the noise in measurements that propagates to the control variable. In (P2), the stochastic component results from the anticipation of unknown target trajectory. None of these two problems includes "real" disturbances, for example, stochastic drifts [33]. By this, we underline that instead of using stochastic processes exclusively for modeling of real, physical disturbances, we can use them to formulate probability priors for the future evolution of dynamical system states.

We present here the problems (P1) and (P2) because they can be formulated as stochastic optimal control problems for which there exists stochastic optimal control feedback control [1]. The emphasis on feedback control, i.e., closed-loop optimal control, is because it naturally leads to robot reactive behaviors. Note that

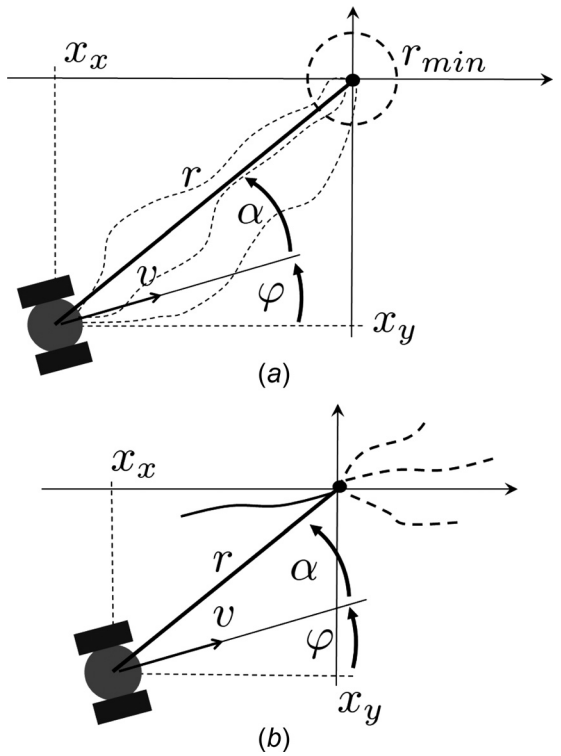

Fig. 1 Fixed velocity two-wheel robot control problems: (*a*) Minimum expected time control (P1); (*b*) Distance keeping control (P2); (x_x,x_y)—the robot coordinates relative to the target which is at the origin, *r*—distance between the robot and the target, φ—robot heading angle, α—bearing angle and *v*—velocity

the solutions of the problems we present are based on value functions in the state space, which is in the core of machine learning control methods known as reinforcement learning [34], or more recently, as approximate dynamic programming [35]. In this sense, we would like to point that the computation of optimal control can be linked to the problems of spatial or spatiotemporal modeling of the scalar function, the techniques which are covered in Sec. 4.

3.1 Minimum Expected Time Control (P1). In Fig. 1(*a*), the robot navigates based on the feedback controlled turning rate u with a constant velocity v to reach the target point seven in a minimum time. Even though the robot motion is precisely executed, the increments of the heading angle $d\varphi$ may be noisy, therefore, stochastic. To anticipate that uncertainty, we use the following robot model:

$$\begin{aligned} dx_x &= \nu\cos\theta dt \\ dx_y &= \nu\sin\theta dt \\ d\varphi &= udt + \sigma_r dw \end{aligned} \tag{15}$$

in which x_x and x_y are the robot coordinates (see Fig. 1) and the last equation defines a probabilistic prior for the evolution of the heading angle φ. The prior can be computed [36], but this is not a necessary step for the computation of optimal feedback control.

Although u is a well-defined, measurement dependent, feedback control variable, in our model we anticipate that the robot trajectory will be always different even for the same initial position. Since the SDE model in Eq. (15) defines a family of trajectories, we can only look for the control u that minimizes the expected time to the target, i.e., the cost function is defined as

$$J(u) = \mathbb{E}\left\{\int_0^\tau 1dt\right\} \tag{16}$$

where τ is the time point in which the robot reaches a target set. Since we would like to compute control that depends on the relative position, we use the relative coordinates $r = \sqrt{x_x^2 + x_y^2}$ and α (see Fig. 1), and Îto calculus to express the kinematics in Eq. (15) as

$$\begin{aligned} dr &= -\nu\cos\alpha dt \\ d\alpha &= \left(\frac{\nu}{r}\sin\alpha - u\right)dt + \sigma_r dw \end{aligned} \tag{17}$$

Because of the second equation, which is not bounded for $r \to 0$, we define the target $\mathcal{T}$ not as a point, but as a set $\mathcal{T} := \{(r,\alpha) \mid r \in [0, r_{\min}],\ \alpha \in [-\alpha_{\max}, \alpha_{\max}]\},\ \alpha_{\max} \in (0, \pi/2)$. The target set $\mathcal{T}$ is defined as a set of robot positions within the circle of radius $r_{\min}$ and a range of bearing angle α. The bearing angle range is not required, but it is introduced to limit the angle under which the circular set is entered.

We can define the cost-to-go function $V(r,\alpha)$, whose value is the expected cost in Eq. (16) from the point in space (r,α) under the assumption that the optimal control policy is applied. The cost-to-go function can be found as the solution of the HJB equation

$$0 = \min_u\left\{b_1\frac{\partial V}{\partial r} + b_2(u)\frac{\partial V}{\partial \alpha} + \frac{\sigma_r^2}{2}\frac{\partial^2 V}{\partial \alpha^2} + 1\right\} \tag{18}$$

where $b_1 = -\nu\cos\alpha$ and $b_2(u) = (\nu/r)\sin\alpha - u$ match the dt multiplying expressions in Eq. (17), respectively. In this case, the HJB is the second-order partial differential equation (PDE) due to the stochastic component in Eq. (17) and includes the minimization with respect to the control u. Instead of solving the HJB, we can discretize the control problem defined by Eqs. (16) and (17) using the locally consistent Markov-chain approximation [5] and solve the problem computationally as a discrete time Markov-chain control problem.

To discretize the control problem, we first define the discrete steps Δr and $\Delta\alpha$ for r and α, respectively. Then we substitute in Eq. (18), $b_1(\partial V/\partial r) = (V(r+\Delta r,\alpha) - V(r,\alpha))/\Delta r\, b_1^+ - (V(r,\alpha) - V(r-\Delta r,\alpha))/\Delta r\, b_1^-$, which is the derivative's upwind approximation, $b_1^+ = \max[0, b_1]$, $b_1^- = \max[0, -b_1]$. We use the same definition for b_2^+ and b_2^- and substitute the derivative with respect to α as $b_2(\partial V/\partial\alpha) = (V(r,\alpha+\Delta\alpha) - V(r,\alpha))/\Delta\alpha\, b_2^+ - (V(r,\alpha) - V(r,\alpha-\Delta\alpha))/\Delta\alpha\, b_2^-$. Finally, we substitute the second derivative in Eq. (18) as $(\partial^2 V/\partial\alpha^2) = (V(r,\alpha+\Delta\alpha) + V(r,\alpha-\Delta\alpha) - 2V(r,\alpha))/(\Delta\alpha)^2$, and if we move all the terms that include $V(r,\alpha)$ to the left side of Eq. (18), define $|b_1| = b_1^+ + b_1^-$, $|b_2| = b_2^+ + b_2^-$ and $\Delta t = \left((|b_1|/\Delta r) + (|b_2|/\Delta\alpha) + (\sigma_r^2/(\Delta\alpha)^2)\right)^{-1}$ we obtain

$$\begin{aligned} V(r,\alpha) = \min_u \{&p_{\Delta r^+}V(r+\Delta r,\alpha) + p_{\Delta r^-}V(r-\Delta r,\alpha) \\ &+ p_{\Delta\alpha^+}V(r,\alpha+\Delta\alpha) + p_{\Delta\alpha^-}V(r,\alpha-\Delta\alpha) + \Delta t\} \end{aligned} \tag{19}$$

with $p_{\Delta r^\pm} = \Delta t(b_1^\pm/\Delta r)$ and $p_{\Delta\alpha^\pm} = \Delta t\left(b_2^\pm/\Delta\alpha + \sigma_r^2/(2\Delta\alpha^2)\right)$ that can be interpreted as the discrete Markov-chain transition probabilities from the point (r,α) to the points that are denoted in the brackets of V. Note that Δt is the time interpolation interval defined by other problem parameters; therefore, this type of discretization is called time implicit discretization [5].

Expression (19) is the discrete variant of Eq. (18), and it can be solved numerically using the so-called value iteration [37] starting from an initial guess for $V(r,\alpha)$ values. However, as in the case of PDEs, for the bounded domain in which we compute the solution, we have to specify boundary conditions. The computational domain is defined as $C = \{[r_{\min} r_{\max}] \times [-\pi, \pi - \Delta\alpha)\}$, which is the

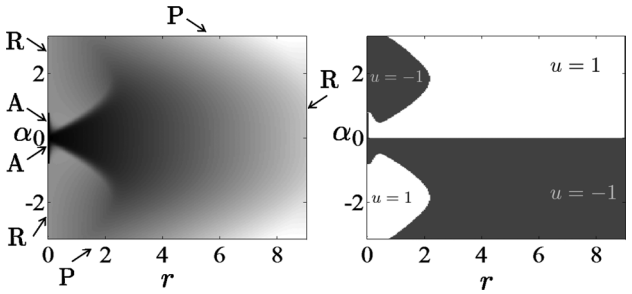

Fig. 2 Solution of the minimum expected time problem (P1): (left panel) gray colored map of the value function $V(r, \alpha)$; black color at the absorbing boundary (A) indicates $V(r, \alpha) = 0$ and the lighter shades depict longer expected times. The type of the boundary conditions is labeled by P—periodic, R—reflective, A—absorbing; (right panel) optimal feedback control; white $u = 1$ and gray $u = -1$.

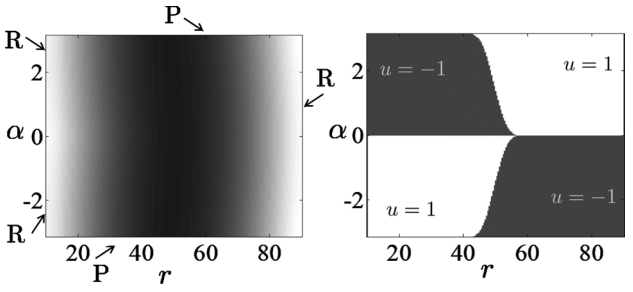

Fig. 3 Solution of the distance keeping control problem (P2): (left panel) gray colored map of the value function $V(r, \alpha)$; darker shades correspond to smaller values of $V(r, \alpha)$ and lighter shades depict its larger values. The type of the boundary conditions are labeled by P—periodic, R—reflective; (right panel) optimal feedback control, white $u = 1$ and gray $u = -1$.

set bounded by the minimal $r_{\min}$ and maximal distance $r_{\max}$. Since the angle a has a full range, $\alpha \in [-\pi, \pi - \Delta\alpha]$, we have the periodic boundary condition $V(r, \pi - \Delta\alpha) = V(r, -\pi + \Delta\alpha)$. A part of the boundary $r_{\min}$ with $\alpha \in [-\alpha_{\max}, \alpha_{\max}]$ belongs to our target set; therefore, we specify that the expected time to reach the set of these points is zero, i.e., $V(r_{\min}, \alpha) = 0$ for $\alpha \in [-\alpha_{\max}, \alpha_{\max}]$. This is the so-called absorbing boundary condition. For all other segments of the boundaries defined by $r_{\min}$ and $r_{\max}$, we use the reflective boundary conditions $V(r_{\min} + \Delta r, \alpha) = V(r_{\min}, \alpha)$ and $V(r_{\max} + \Delta r, \alpha) = V(r_{\max}, \alpha)$. This way we avoid specifying the value function at these boundaries, but we actually incorporate in our solution that the stochastic process does not cross the boundaries. This is true for the segments defined by $r = r_{\min}$ and $|\alpha| > \alpha_{\max}$, but it is an approximation for the boundary $r = r_{\max}$; therefore, we should use a large enough $r_{\max}$.

For the parameters of the problem $\sigma = 0.1$, $\alpha_{\max} = \pi/4$ and the discretization $\Delta r = 0.045$ and $\Delta\alpha = 2\pi/361$, $r_{\min} = 0.01$ and $r_{\max} = 9.0$, we obtain the results presented in Fig. 2. The figure also identifies the boundaries and their nature. Note that the computed control is "bang-bang," which is a consequence of the fact that we do not penalize the control. Therefore, the shape of control policy is completely determined by the problem's nonlinearity and the presence of noise.

3.2 Distance Keeping Control (P2). The control in which we navigate the robot to keep a constant distance from a moving target is depicted in Fig. 1(*b*). In this problem, the motion of the robot is modeled by Eq. (15) with $\sigma_r = 0$, which is the deterministic model, while the target motion is modeled as $dx_T = \sigma_T dw$, $dy_T = \sigma_T dw$. Now, we use the distance between the robot and the target $r = \sqrt{(x_x - x_T)^2 + (x_y - y_T)^2}$ and the bearing angle α (see Fig. 1(*b*)) to define the cost function

$$J(u) = \mathbb{E}\left\{\int_0^\infty e^{-\beta t}(r-d)^2 dt\right\} \tag{20}$$

which is defined on the infinite time horizon and includes the discounting factor $e^{-\beta t}$. To keep a constant distance, i.e., $r = d$, we compute an optimal control that minimizes the cost function (20). All details on how to compute this control are provided in Ref. [38], and here, we provide only major points. First, we apply the Ito chain rule to derive the SDE describing evolution r and α, which is

$$dr = \left(-\nu\cos\alpha + \frac{\sigma_T^2}{2r}\right)dt + \sigma_T dw_r$$
$$d\alpha = \left(\frac{\nu}{r}\sin\alpha - u\right)dt + \frac{\sigma_T}{r}dw_n$$

where dw_r and dw_n are the Wiener process increments along the robot-target direction and normal to it, respectively. The HJB equation in this case is

$$0 = \min_u\left\{b_1\frac{\partial V}{\partial r} + b_2(u)\frac{\partial V}{\partial \alpha} + \frac{\sigma_T^2}{2}\frac{\partial^2 V}{\partial r^2} + \frac{\sigma_T^2}{4r^2}\frac{\partial^2 V}{\partial \alpha^2} - \beta V(r,\alpha)\right\}$$

with $b_1 = -\nu\cos\alpha + \sigma_T^2/(4r^2)$ and $b_2(u) = (\nu/r)\sin\alpha - u$, where the $\beta V(r, \alpha)$ term is the consequence of the discounting factor $e^{-\beta t}$. The discretization procedure and the value iteration step are the same as in the previous example, except that in this case the boundaries for $r_{\min}$ and $r_{\max}$ are both reflective.

For the problem parameters $\beta = 1$, $d = 50$ $\sigma_T = 5$ and the discretization parameters $r_{\min} = 10$, $r_{\max} = 90$, $\Delta r = 0.4$, and $\Delta\alpha = 2\pi/361$, we obtain the results presented in Fig. 3. Our computations again result in a control u that is bang-bang. It is worth mentioning, based on the results presented in Ref. [38], that the slope of the curved line separating different values of the control depends on the intensity of the noise scaling parameter σ_T.

4 Spatiotemporal Models

In this section, we discuss a series of spatial or spatiotemporal models, which can be used to model uncertain environments for mobile sensor networks as well as approximate the value function for optimal control. To this end, we introduce parametric, nonparametric, empirical Bayes, and fully Bayeisan approaches [16–18,26–28].

4.1 Parametric Approach. Among phenomenological spatial models, the simplistic model is a static spatial field that could be modeled by a network of basis functions. A measurable environmental variable μ at $x \in$ X location could be modeled by

$$\mu(x) := \sum_{j=1}^{m} f_j(x)\beta_j = f^T(x)\beta \tag{21}$$

where $f^T(x)$ and β are defined, respectively, by

$$f(x) := [f_1(x)\cdots f_m(x)]^T \in \mathbb{R}^m, \quad \beta := [\beta_1 \cdots \beta_m]^T \in \mathbb{R}^m$$

For example, the basis function could take a Gaussian function $f_j(x) := (1/Z)\exp(-\|x - c_j\|^2/\sigma_j^2)$, where c_j and σ_j are the center location and the width of the basis function, respectively.

Adaptive control of multiple robotic sensors based on the parametric model in Eq. (21) can be designed as a sum of swarming and gradient ascent efforts as follows:

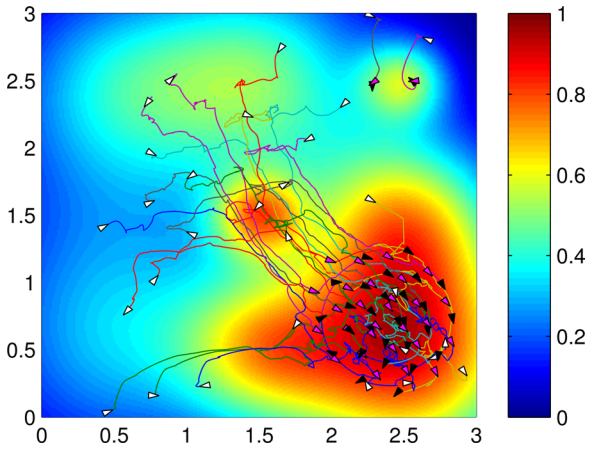

Fig. 4 Each agent is driven by swarming and gradient ascent efforts based on its own recursively estimated field (with a model in Eq. (21)) via locally collected measurements by itself and its neighboring agents. The multiagent system locates peaks of an uncertain static field in a distributed and scalable manner [39].

$$\begin{aligned}\frac{dq^{(i)}}{dt} &= p^{(i)} \\ \frac{dp^{(i)}}{dt} &= u_s\left(q^{(i)}, p^{(i)}\right) - \varepsilon_1 p^{(i)} + \varepsilon_2 f'^T\left(q^{(i)}\right)\hat{\beta}_i(t)\end{aligned} \tag{22}$$

where $q^{(i)}$ and $p^{(i)}$ denote the position and velocity of agent i. $\varepsilon_i > 0$ are gain factors. $u_s(\cdot,\cdot)$ denotes the swarming effort [39] and $f'(x^*)$ is the partial derivative of $f(x)$ with respect to x^*. Note that $\hat{\beta}_i(t)$ is updated based on the recursive least squares estimator using locally collected measurements [16,39]. The multiagent system under distributed control in Eq. (22) locates peaks of an uncertain static field as shown in Fig. 4. However, the control in Eq. (22) needs a persistent excitation condition for convergence of regression coefficients in β [16,39] (Fig. 4), while control strategies based on Bayesian spatial models do not require such conditions, (e.g., by utilizing priori distributions as in Kalman filtering [15] or Gaussian process regression [17]). Hence, control engineers become more aware of the usefulness of nonparametric Bayesian approaches such as Gaussian processes [40,41] to statistically model physical phenomena for the navigation of mobile sensor networks [14,17,23–27,42,43]. A Gaussian process is an elegant way of defining prior distributions for data-driven and flexible regression. We review models based on Gaussian process regression in Sec. 4.2.

4.2 Nonparametric Approach: Gaussian Processes. A Gaussian process is a collection of random variables, any finite number of which have a joint Gaussian distribution [41,44]. A Gaussian process at a point $x \in \mathrm{X}$ can be defined by a mean function $\mu(x) := \mathbb{E}[z(x)]$ and a covariance function $C(x,x') := \mathbb{E}[(z(x)-\mu(x))(z(x')-\mu(x'))]$, where $x, x' \in \mathrm{X}$, which is denoted by

$$z(x) \sim \mathcal{GP}(\mu(x), C(x,x')) \tag{23}$$

For example, a point x in (d-dimensional) space and time takes $x := (s,t) \in \mathbb{R}^{d+1}$, where $s \in \mathbb{R}^d$ and $t > 0$. The Gaussian process in Eq. (23) is a prior over the function $z(x)$ and will be used for Bayesian regression of $z(x)$ for the given measured data. Before the data are collected, the Gaussian process prior provides the uncertainty about the unobserved function $z(x)$. After observing the data, our belief on $z(x)$ shall be updated as a conditional probability distribution of $z(x)$ for the given data. The conditional probability distribution of z on x_*, which is not measured, is called a posterior predictive distribution.

A mean function may take a form of Eq. (21), and a covariance function could take a form of the squared exponential covariance function

$$C(x,x') = \sigma_f^2 \exp\left(-\frac{1}{2}\|x-x'\|^2_{\Sigma_b^{-1}}\right) \tag{24}$$

$\|x-x'\|^2_{\Sigma_b^{-1}}$ is the weighted vector norm of $x-x'$ with a weight (of bandwidths) Σ_b^{-1} such that

$$\|x=x'\|^2_{\Sigma_b^{-1}} := (x-x')^T\Sigma_b^{-1}(x-x')$$

When $d=2$, Σ_b can be parameterized by a vector θ.

$$\Sigma_b(\theta) = \begin{bmatrix}\sigma_1^2 & 0\\ 0 & \sigma_2^2\end{bmatrix}, \quad \theta := \begin{bmatrix}\sigma_1^2 & \sigma_2^2\end{bmatrix} \tag{25}$$

A covariance function for a spatiotemporal Gaussian process can take the following form:

$$C(x,x') = \sigma_f^2 \mathrm{Corr}(x,x') = \sigma_f^2 \mathrm{Corr}_s\left(\frac{\|s-s'\|}{\sigma_s}\right)\mathrm{Corr}_t\left(\frac{|t-t'|}{\sigma_t}\right)$$

where $\mathrm{Corr}_s(\cdot)$ and $\mathrm{Corr}_t(\cdot)$ are the spatial and time correlation functions, respectively. Here, we have $\theta := (\sigma_s, \sigma_t)^T \in \mathbb{R}^2$.

If a covariance function is valid, it must result in a positive definite covariance matrix for any set of sampling points. This can be shown by considering an arbitrary collection $\{z(x^{(1)}), \ldots, z(x^{(n)})\}$ of random variables from a Gaussian process with covariance function $C(x,x')$. For arbitrary real numbers $\{c_1, \ldots, c_n\}$ we then have

$$\mathrm{Var}\left[c_1 z(x^{(1)}) + \cdots + c_n z(x^{(n)})\right] = \sum_{i=1}^n \sum_{j=1}^n c_i c_j C(x^{(i)}, x^{(j)}) > 0$$

If the covariance function is invariant to translations in the input space, i.e., $C(x, x') = C(x-x')$ we call it stationary. Furthermore, if the covariance function is a function of only the distance between the inputs, i.e., $C(x,x') = C(\|x-x'\|)$ then it is called isotropic. For a review of covariance functions, the reader is referred to Refs. [41] and [45].

Suppose we have n sampling positions and the associated noise-corrupted observations $\mathcal{D} := \{(x^{(i)}, y^{(i)}) | i = 1, \ldots, n\}$, and the measurement noise is given such that $y|z \sim \mathcal{N}(z, \sigma_w^2 I)$. Then the collection of observations $y = [y^{(1)} \cdots y^{(n)}]^T \in \mathbb{R}^n$ has the Gaussian distribution

$$y \sim \mathcal{N}(\mu, C_y)$$

where $\mu_i := \mu(x^{(i)})$, and $(C_y)_{ij} := C(x^{(i)}, x^{(j)}) + \sigma_w^2 I$.

The objective of Gaussian process regression is to compute the posterior predictive distribution of the function values $z_* := z(x_*)$ at a target point x_* for given observations $\mathcal{D}$. This posterior predictive distribution is a Gaussian distribution as follows:

$$z_* \Big| \mathcal{D} \sim \mathcal{N}\left(\mu_{z_*|\mathcal{D}}, \sigma^2_{z_*|\mathcal{D}}\right) \tag{26}$$

where the prediction of z_* is given by

$$\mu_{z_*|\mathcal{D}} = \mu(x_*) + k^T C_y^{-1}(y-\mu) \tag{27}$$

here k is the covariance between z and z_* obtained by $(k)_i = C(x^{(i)}, x_*)$. The prediction error variance of z_* is given as

$$\sigma^2_{z_*|\mathcal{D}} = C(x_*, x_*) - k^T C_y^{-1} k = \sigma_f^2 - k^T C_y^{-1} k \tag{28}$$

Note that both Eqs. (27) and (28) have complexity $O(n^3)$ due to C_y^{-1}. Thus, it is difficult for resource-constrained robots to use Gaussian process regression in Eq. (27) since it is not scalable as the number of observations n increases. For various topics on flexible and efficient Gaussian process regression, see Ref. [46] and references therein. Distributed coordination of mobile sensor networks under the limited communication range using truncated observations is proposed in Ref. [17]. Mobile sensing agents with the distributed navigation strategy produce an emergent, swarminglike, collective behavior for communication connectivity and are coordinated to improve the quality of the collective prediction capability.

The quality of the prediction over a collection of target points T can be measured by the averaged prediction error variance over T and agents which is a function of sampling positions $q = \mathrm{col}(q^{(1)}, \ldots, q^{(n)})$.

$$J(q) := \frac{1}{n} \sum_i \frac{1}{|T|} \sum_{x \in T} \sigma^{2(i)}_{z(x)|\mathcal{D}^{(i)}}\left(q^{(i)}\right) \tag{29}$$

where $|T|$ is the cardinality of T. Superscript i is given to local prediction error variance based on local data from agent i. Note that J in Eq. (29) is a function of q not of observations y. This implies that we can minimize the cost function J over possible sampling positions q of robotic sensors without taking any actual measurements when the mean and covariance functions of the Gaussian process in Eq. (23) are perfectly known.

4.3 Selection of Gaussian Process Prior. In this subsection, we illustrate the importance of selecting a Gaussian process prior via hyperparameters when we make an inference from the experimental data. To have illustrative cases, we consider the experimental data collected by the robotic boat (see Fig. 5(*a*)) that was deployed in a private pond in Lansing, MI. Similar robotic boats were used to sample the environmental variables [20,47,48]. A set of water depth values and sampling locations were collected from on-board sensors in the robotic boat. Fig. 5(*b*) shows the location site with sampling locations shown in colored dots. Without loss of generality, the global positioning system (GPS) data, in particular the longitude and latitude, are normalized to [0, 1]. Let us assume that the process has a known constant mean so that we only care to select the covariance function. With the squared exponential covariance function in Eqs. (24) and (25), i.e.,

$$C(x, x'; \theta) = \sigma_f^2 \exp\left\{ -\sum_{\ell=1}^{2} \frac{1}{2} \frac{(x_\ell - x'_\ell)^2}{\sigma_\ell^2} \right\},$$

the estimated depth field and the prediction error variance with different hyperparameters are shown in Figs. 6 and 7, respectively. With different bandwidths (or length scales) σ_1 and σ_2 while keeping other parameters fixed on the same values, the prediction and its prediction error variance on the same experimental data are significantly different. With smaller bandwidths, the predicted depth field is much more wiggly and wavy than its counterpart (Figs. 6(*a*) and 7(*a*)). Very large values of the bandwidths imply that the depth value is expected to be constant with respect to different locations. As shown in Figs. 6 and 7, the Gaussian process with smaller bandwidths (and weaker spatial correlations) allows much more complicated spatial details and needs to be densely sampled. On the other hand, the Gaussian process with larger bandwidths (and stronger spatial correlations) tends to be smooth and does not need dense sampling for a decent level of prediction quality as shown in Fig. 7. From these findings, we recognize the importance of the choice of the covariance function to make precise prediction as well as schedule the sampling in an optimal way. More discussions on the selection of covariance functions from a Bayesian perspective can be found in Ref. [44].

(*a*)

(*b*)

Fig. 5 (*a*) Remotely controlled boat equipped with depth sensor and GPS and (*b*) experiment site with the sampling locations (shown as colored dots)

4.4 Empirical Bayes Approach. When mean and covariance functions are not known, they need to be estimated from the experimental observations. Consider a case where the mean function is known and the covariance function needs to be estimated. For given collected data, hyperparameters ϕ of the covariance function in Eq. (24), i.e., $\phi := \{\sigma_f^2, \theta\}$, can be estimated by an empirical Bayes method. The estimated parameters can be considered as true ones and we can then proceed to make inferences using Eq. (26). The point estimator of ϕ can be obtain by maximizing the likelihood function, i.e., $L(\phi|y) = \pi(y|\phi)$. The maximum likelihood (ML) estimate ϕ_{ML} is then defined as

$$\hat{\phi}_{ML} = \arg\max_{\phi \in \Phi} L(\phi|y) = \arg\max_{\phi \in \Phi} \log L(\phi|y)$$

Note that it is convenient to use the log likelihood function

$$\log L(\phi|y) = -\frac{1}{2}(y-\mu)^T C_y^{-1}(y-\mu) - \frac{1}{2}\log\det(C_y) - \frac{n}{2}\log 2\pi$$

If the prior distribution $\pi(\phi)$ is available, we could seek for the maximum a posteriori (MAP) estimate $\hat{\phi}_{\mathrm{MAP}}$.

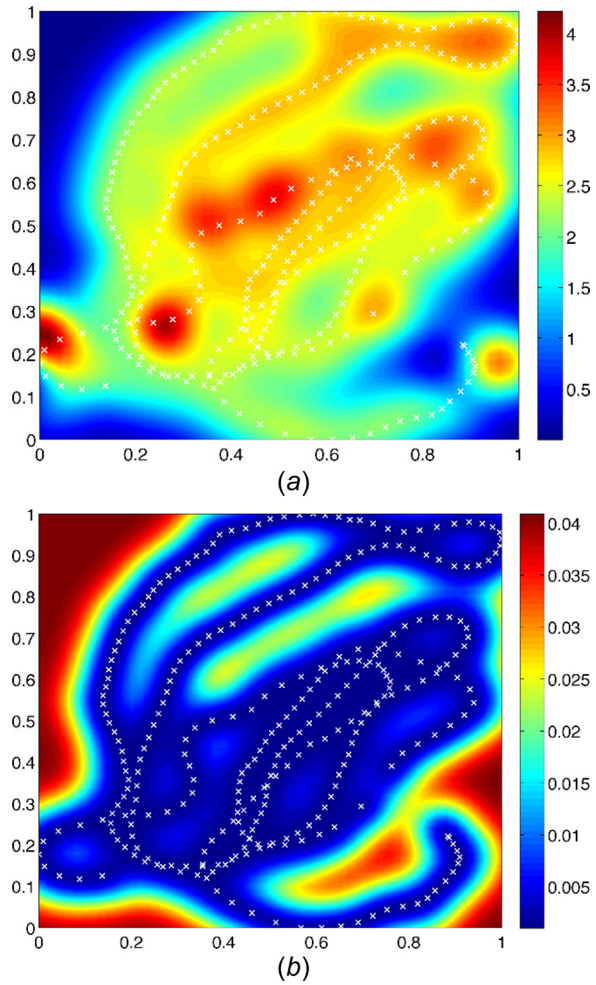

Fig. 6 Prediction with hyperparameters $\sigma_f = 0.2$, $\sigma_1 = 0.07$, $\sigma_2 = 0.07$, and $\sigma_w = 0.03$. (*a*) Estimated depth and (*b*) prediction error variance, with sampling positions shown as white crosses.

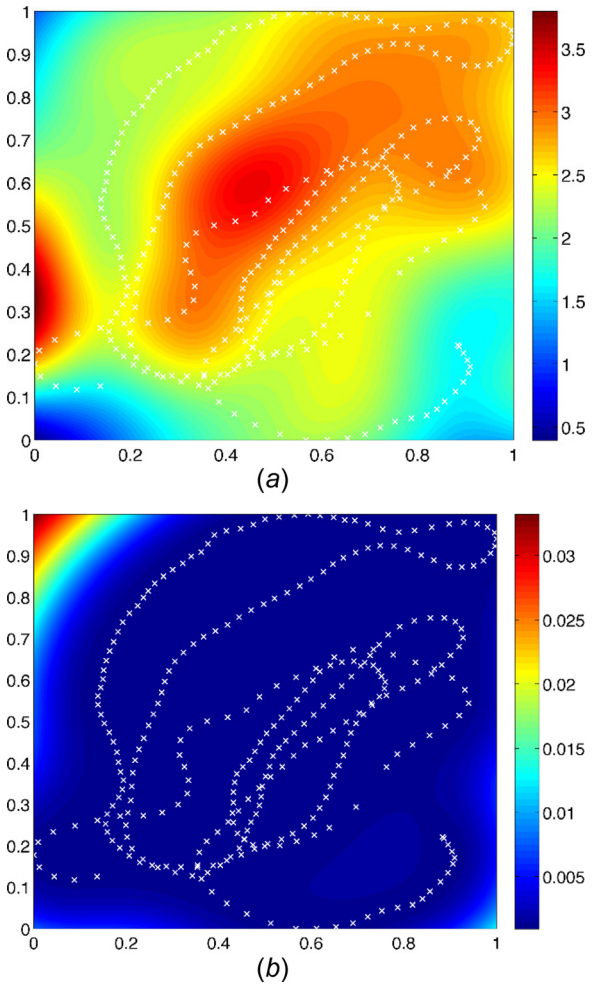

Fig. 7 Prediction with hyperparameters $\sigma_f = 0.2$, $\sigma_1 = 0.2$, $\sigma_2 = 0.2$, and $\sigma_w = 0.03$. (*a*) Estimated depth and (*b*) prediction error variance, with sampling positions shown as white crosses.

$$\hat{\phi}_{MAP} = \arg\max_{\phi\in\Phi} \log[\pi(y|\phi)\pi(\phi)]$$
$$= \arg\max_{\phi\in\Phi} [\log L(\phi|y) + \log \pi(\phi)]$$

Gradient-based optimization techniques (e.g., the conjugate gradient method) could find the point estimators using the partial derivative of $L(\phi|y)$ with respect to ϕ_i.

$$\frac{\partial \log L(\phi|y)}{\partial \phi_i} = \frac{1}{2}(y-\mu)^T C_y^{-1} \frac{\partial C_y}{\partial \phi_i} C_y^{-1}(y-\mu) - \frac{1}{2}\mathrm{tr}\left(C_y^{-1}\frac{\partial C_y}{\partial \phi_i}\right)$$

The aforementioned methods can be easily extended for a case where both the mean and covariance functions need to be estimated. Mobile sensor agents can be moved in order to maximize the information contained in the observations for estimating hyperparameters [49]. The cost function for mobility could be selected according to D-optimality: $J(q) := \log\det(M^{-1}(q;\phi))$ or A-optimality: $J(q) := \mathrm{tr}(M^{-1}(q;\phi))$ where M is the Fisher information matrix.

$$[M(q;\phi)]_{ij} = -\mathbb{E}\left(\frac{\partial^2 L(\phi|y)}{\partial\phi_i \partial\phi_j}\right)$$

Such criteria are popular choices for optimal experimental design since the Cramer–Rao lower bound (CRLB) theorem states $\mathbb{E}[(\hat{\phi}-\phi)(\hat{\phi}-\phi)^T] \succeq M^{-1}$. In fact, M is not only a function of sampling points q but also that of the true parameters ϕ. Therefore, a good initial guess needs to be plugged in for optimization.

4.5 Fully Bayesian Approach. While this empirical Bayes method is quite practical to use, the point estimate (ML or MAP estimate) itself needs to be identified a priori and it does not fully incorporate the uncertainty in the estimated hyperparameters into the prediction in a fully Bayesian perspective. The advantage of a fully Bayesian approach is that the uncertainty in the model parameters are incorporated in the prediction [50]. The main difference is that in a fully Bayesian approach, unknown hyperparameters are considered as random variables while they are considered as unknown constants for a frequenist point of view.

In this subsection, to present a fully Bayesian approach, we treat unknown β, σ_f^2, and θ as random variables. Therefore, we define the prior distribution reflecting the a priori belief of uncertainty for them.

$$\pi\left(\beta, \sigma_f^2, \theta\right) = \pi\left(\beta\middle|\sigma_f^2\right)\pi\left(\sigma_f^2\right)\pi(\theta) \tag{30}$$

where $\beta|\sigma_f^2 \sim \mathcal{N}(\beta_0, \sigma_f^2 T)$ and $\sigma_f^2 \sim IG(a,b)$ which is taken to be the inverse gamma distribution, chosen to guarantee positiveness of σ_f^2 and a closed-form expression for the posterior distribution of σ_f^2 for computational ease of the method. In case there is no prior information about β, noninformative prior can be used in a

Table 1 A Gibbs sampler

Input: Initial samples $\beta^{(1)}$, $\sigma_f^{2(1)}$, and $\theta^{(1)}$

Output: Samples $\left\{\beta^{(i)}, \sigma_f^{2(i)}, \theta^{(i)}\right\}_{i=1}^{m}$ from joint distribution $\pi\left(\beta, \sigma_f^2, \theta|y\right)$

1: Initialize $\beta^{(1)}$, $\sigma_f^{2(1)}$, $\theta^{(1)}$
2: **for** $i=1$ to m **do**
3: sample $\beta^{(i+1)}$ from $\pi(\beta|\sigma_f^{2(i)}, \theta^{(i)}, y)$
4: sample $\sigma_f^{2(i+1)}$ from $\pi(\sigma_f^2|\beta^{(i+1)}, \theta^{(i)}, y)$
5: sample $\theta^{(i+1)}$ from $\pi(\theta|\beta^{(i+1)}, \sigma_f^{2(i+1)}, y)$
6: **end for**

Bayesian perspective. This case, let $\beta_0=0$, and $T=\alpha I$, subsequently, let $\alpha \to \infty$.

The posterior distribution due to the Bayes rule is then given by

$$\pi\left(\beta, \sigma_f^2, \theta|y\right) = \frac{\pi\left(y\middle|\beta, \sigma_f^2, \theta\right)\pi\left(\beta, \sigma_f^2, \theta\right)}{\iiint \pi\left(\beta, \sigma_f^2, \theta\right)\pi\left(\beta, \sigma_f^2, \theta\right)d\beta d\sigma_f^2 d\theta} \tag{31}$$

Since there is no analytical solution to compute (31), a Markov-chain Monte Carlo (MCMC)-based approach [51] can be used to make inference. For example, a Gibbs sampler can be use to compute Eq. (31). The inference on β, σ_f^2, and θ can be carried out by sampling from the posterior distribution in Eq. (31) via the Gibbs sampler as shown in Table 1. The derivation of the conditional distributions in Table 1 can be found in Ref. [52].

For this MCMC-based approach, the posterior predictive distribution can be written by

$$\pi(z_*|y) = \iiint \pi\left(z_*|y, \beta, \sigma_f^2, \theta\right)\pi\left(\beta, \sigma_f^2, \theta|y\right)d\beta d\sigma_f^2 d\theta$$

Note that the predictive distribution for given $\beta, \sigma_f^2, \theta$, and y is given by

$$z_*|\beta, \sigma_f^2, \theta, y \sim \mathcal{N}\left(\mu_{z_*|\beta,\sigma_f^2,\theta,y}, \sigma^2_{z_*|\beta,\sigma_f^2,\theta,y}\right), \text{with}$$
$$\mu_{z_*|\beta,\sigma_f^2,\theta,y} = \mathbb{E}\left(z_*|\beta, \sigma_f^2, \theta, y\right) = f(x_*)^T\beta + k^T C_y^{-1}(y - F\beta),$$
$$\sigma^2_{z_*|\beta,\sigma_f^2,\theta,y} = \text{Var}\left(z_*|\beta, \sigma_f^2, \theta, y\right) = \sigma_f^2 - k^T C_y^{-1} k$$

F is defined as $F := [f(x^{(1)}) \cdots f(x^{(n)})^T] \in \mathbb{R}^{n\times m}$.

Now approximated posterior predictive distribution can be computed using samples drawn from the Gibbs sampler

$$\pi(z_*|y) \approx \frac{1}{m}\sum_{i=1}^{m} \pi\left(z_*|y, \beta^{(i)}, \sigma_f^{2(i)}, \theta^{(i)}\right)$$

where $\beta^{(i)}$, $\sigma_f^{2(i)}$, and $\theta^{(i)}$ are drawn from the Gibbs sampler given in Table 1. Posterior predictive mean and variance can then be obtained as follows:

$$\mu_{z_*|y} = \mathbb{E}(z_*|y) \approx \frac{1}{m}\sum_{i=1}^{m} \mu_{z_*|\beta^{(i)},\sigma_f^{2(i)},\theta^{(i)},y}.$$
$$\sigma^2_{z_*|y} = \text{Var}(z_*|y) \approx \frac{1}{m}\sum_{i=1}^{m} \sigma^2_{z_*|\beta^{(i)},\sigma_f^{2(i)},\theta^{(i)},y} + \frac{1}{m}\sum_{i=1}^{m}\left(\mu_{z_*|\beta^{(i)},\sigma_f^{2(i)},\theta^{(i)},y} - \mu_{z_*|y}\right)^2$$

4.5.1 The MCMC-Based Solution to a 1D Example. Let us consider a scenario in which five robots sample the spatiotemporal Gaussian process in a 1D space and the central station performs Bayesian prediction. The signal-to-noise ratio is set to be 26 dB, which corresponds to $\sigma_w = 0.05$. The true values for the parameters used in simulating the Gaussian process are given by $[\beta \ \sigma_f^2 \ \sigma_s \ \sigma_t]^T = [0 \ 1 \ 2 \ 8]^T$ for a squared exponential function $\text{Corr}_s(h) = -\frac{1}{2}h^2$ and a compactly supported correlation function for time $\text{Corr}_t(h)$ used in Ref. [26]. Notice that the dimension of the regression coefficient β is assumed to be one, i.e., the mean function is assumed to be an unknown random variable. We assume that $\beta|\sigma_f^2$ has the uninformative prior and $\sigma_f^2 \sim IG(3, 20)$. The Gibbs sampler in Table 1 was used to generate samples from the posterior distribution of the parameters. A random sampling strategy was used in which robots make observations at random locations at each time step.

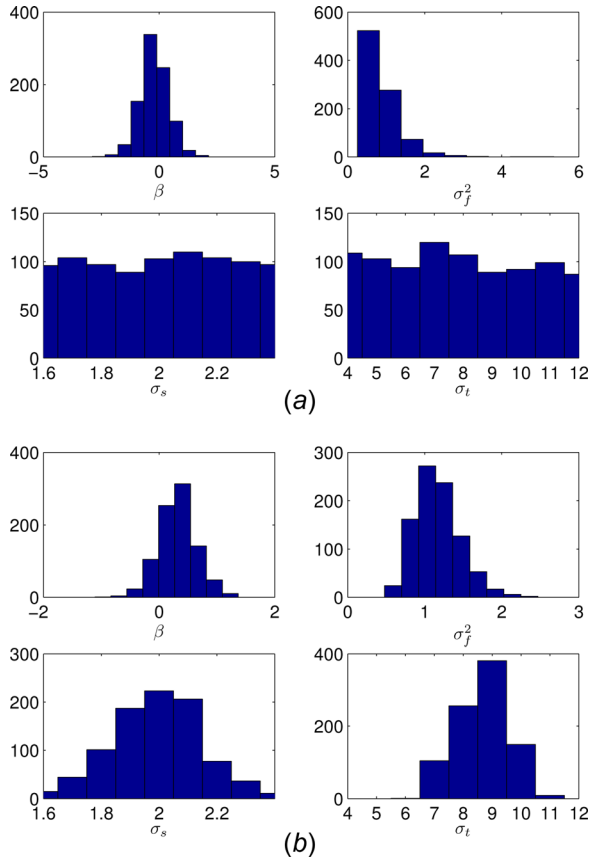

Fig. 8 The posterior distribution of β, σ_f^2, σ_s, and σ_t at (*a*) t_1 and (*b*) t_{20}

The histograms of the samples at time t_1 and t_{10} are shown in Figs. 8(*a*) and 8(*b*), respectively. It is clear that the distributions of the parameters are centered around the true values with 100 observations at time t_{20}. The prediction results at time t_1 and t_{20} are shown in Figs. 9(*a*) and 9(*b*), respectively. However, with only 100 observations, the running time using the full Bayesian approach is about several minutes, which will soon become intractable as the number of observations increases.

4.6 Efficient Fully Bayesian Methods. While this fully Bayesian approach is flexible and truly adaptive to collected data, MCMC methods are not suitable for resource-constrained mobile robots due to the high computational complexity. Some efforts made to derive methods without resorting to MCMC methods are discussed in Ref. [26]. In Ref. [24], an iterative prediction algorithm without resorting to MCMC methods was developed based on analytical closed-form solutions from Ref. [53], by assuming that the bandwidths in θ in the covariance function of the

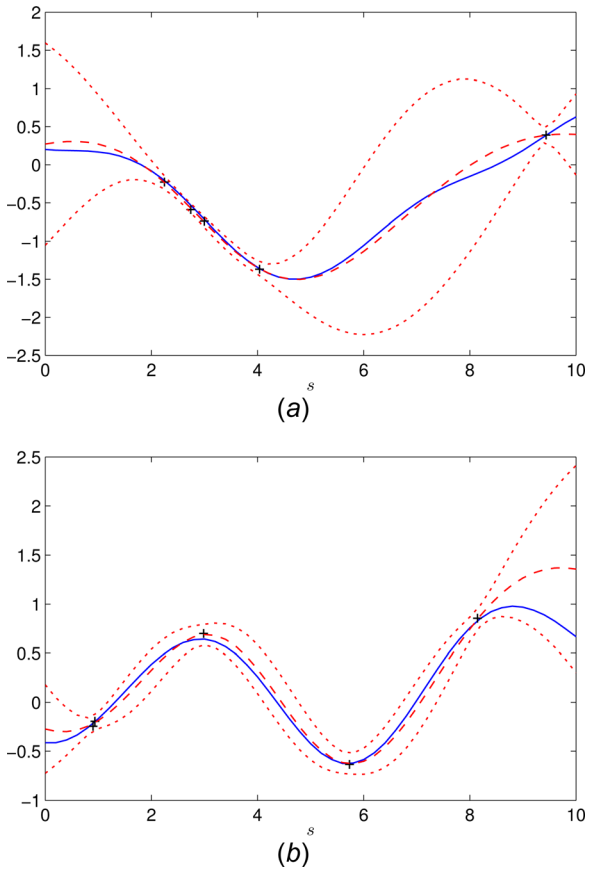

Fig. 9 The prediction at (*a*) t_1 and (*b*) t_{20} using the MCMC-based approach. The true fields are plotted in blue solid lines. The predicted fields are plotted in red dashed lines. The area between red dotted lines indicates the 95% confidence interval.

Gaussian random field are known a priori. In Refs. [26] and [51], a sequential Bayesian prediction algorithm was developed to deal with unknown bandwidths by using a compactly supported kernel and selecting a subset of collected measurements. For a special case, a distributed implementation of sequential Bayesian prediction algorithms was proposed for mobile sensor networks. An adaptive sampling strategy for mobile sensors, using MAP estimation, was proposed to minimize the prediction error variances [26].

Other more data-driven approaches have also developed (without a statistical structure used in Gaussian processes) such as using kernel regression [28] and in reproducing kernel Hilbert spaces [29].

Recently, there have been efforts to fit a computationally efficient GMRF on a discrete lattice to a Gaussian random field on a continuum space [54–56]. It has been demonstrated that GMRFs with small neighborhoods can approximate Gaussian fields surprisingly well [54]. This approximated GMRF and its regression are very attractive for the resource-constrained mobile sensor networks due to its computational efficiency and scalability [57] as compared to the standard Gaussian process and its regression. Fast kriging of large data sets by using a GMRF as an approximation of a Gaussian field has been proposed in Ref. [56]. In Ref. [27], the authors provided a new class of Gaussian processes that builds on a GMRF and derived the formulas for predictive statistics. However, they both assume the precision matrix (i.e., the inverse of the covariance matrix) is given or estimated a priori. In a recent work of Ref. [18], a discretized spatial random field was modeled by a GMRF with uncertain hyperparameters. From a fully Bayesian perspective, a sequential prediction algorithm was designed to exactly compute the predictive inference of the random field. Additionally, the proposed algorithm in Ref. [18] has the computational efficiency due to the sparse structure of the precision matrix, and the scalability as the number of measurements increases.

4.7 Gaussian Processes for Machine Learning (GPML) Toolbox. To jump-start spatial modeling, the readers are encouraged to use the GPML toolbox [58], which has originally grown from the book on Gaussian processes [41]. The GPML toolbox can be downloaded.[2] The GPML toolbox provides a large scope of useful MATLAB functions, for example, to build complex mean and covariance functions from simple ones for Gaussian process inference. It also provides various inference methods including exact and variational inference, expectation propagation [59], Laplace's method [60], and other useful techniques to deal with large scale data.

Acknowledgment

Dr. Milutinović thanks Dr. Ross Anderson for his MATLAB routines used to generate the optimal control solutions. Dr. Choi thanks Dr. Yunfei Xu for his help in MATLAB-based numerical analysis used in this paper. Dr. Choi also thanks NSF RET teacher Mr. Alexander Robinson and Mr. Huan N. Do for collecting the experimental data using the robotic boat. Dr. Choi's work has been supported by the National Science Foundation through CAREER Award CMMI-0846547. This support was gratefully acknowledged.

References

[1] Bertsekas, D. P., 1995, *Dynamic Programming and Optimal Control*, Vol. 1, Athena Scientific, Belmont, MA.
[2] Bryson, A. E., and Ho, Y.-C., 1975, *Applied Optimal Control: Optimization, Estimation, and Control*, Taylor & Francis, New York.
[3] Pontryagin, L. S., 1962, *The Mathematical Theory of Optimal Processes*, Vol. 4, Interscience Publishers, New York.
[4] Young, J., and Zhou, X. Y., 1999, *Stochastic Controls: Hamiltonian Systems and HJB Equations*, Vol. 43, Springer, New York.
[5] Kushner, H. J., and Dupuis, P., 2001, *Numerical Methods for Stochastic Control Problems in Continuous Time*, Vol. 24, Springer, New York.
[6] Stengel, R. F., 1986, *Optimal Control and Estimation*, Dover, New York.
[7] Oksendal, B., 2003, *Stochastic Differential Equations: An Introduction With Applications*, Springer Verlag, New York.
[8] Davis, M. H., 1993, *Markov Models and Optimization*, Vol. 49, Chapman & Hall/CRC, Boca Raton, FL.
[9] Koutsoukos, X. D., 2004, "Optimal Control of Stochastic Hybrid Systems Based on Locally Consistent Markov Decision Processes," Proceedings of the 2005 IEEE International Symposium on Intelligent Control, Limassol, Cyprus, pp. 435–440.
[10] Fleming, W. H., and Soner, H. M., 2006, *Controlled Markov Processes and Viscosity Solutions*, Vol. 25, Springer, New York.
[11] Engel, Y., Mannor, S., and Meir, R., 2005, "Reinforcement Learning With Gaussian Processes," Proceedings of the 22nd International Conference on Machine Learning, University of Bonn, Germany, ACM, pp. 201–208.
[12] Engel, Y., Mannor, S., and Meir, R., 2003, "Bayes Meets Bellman: The Gaussian Process Approach to Temporal Difference Learning," Proceedings of the 20th International Conference on Machine Learning (ICML-2003), Washington, DC, pp. 154–161.
[13] Deisenroth, M. P., Rasmussen, C. E., and Peters, J., 2009, "Gaussian Process Dynamic Programming," Neurocomputing, **72**(7), pp. 1508–1524.
[14] Leonard, N. E., Paley, D. A., Lekien, F., Sepulchre, R., Fratantoni, D. M., and Davis, R., 2007, "Collective Motion, Sensor Networks, and Ocean Sampling," Proc. IEEE, **95**(1), pp. 48–74.
[15] Lynch, K. M., Schwartz, I. B., Yang, P., and Freeman, R. A., 2008, "Decentralized Environmental Modeling by Mobile Sensor Networks," IEEE Trans. Rob., **24**(3), pp. 710–724.
[16] Choi, J., Oh, S., and Horowitz, R., 2009, "Distributed Learning and Cooperative Control for Multi-Agent Systems," Automatica, **45**(12), pp. 2802–2814.
[17] Xu, Y., Choi, J., and Oh, S., 2011, "Mobile Sensor Network Navigation Using Gaussian Processes With Truncated Observations," IEEE Trans. Rob., **27**(6), pp. 1118–1131.
[18] Xu, Y., Choi, J., Dass, S., and Maiti, T., 2013, "Efficient Bayesian Spatial Prediction With Mobile Sensor Networks Using Gaussian Markov Random Fields," Automatica, **49**(12), pp. 3520–3530.
[19] Le Ny, J., and Pappas, G., 2013, "Adaptive Deployment of Mobile Robotic Networks," IEEE Trans. Autom. Control, **58**(3), pp. 654–666.

[2] http://gaussianprocess.org/gpml/code/matlab/doc/index.html

[20] Jadaliha, M., and Choi, J., 2013, "Environmental Monitoring Using Autonomous Aquatic Robots: Sampling Algorithms and Experiments," IEEE Trans. Control Syst. Technol., **21**(3), pp. 899–905.
[21] Cao, Y., Yu, W., Ren, W., and Chen, G., 2013, "An Overview of Recent Progress in the Study of Distributed Multi-Agent Coordination," IEEE Trans. Ind. Inf., **9**(1), pp. 427–438.
[22] Kalnay, E., 2003, *Atmospheric Modeling, Data Assimilation, and Predictability*, Cambridge University, New York.
[23] Cortés, J., 2009, "Distributed Kriged Kalman Filter for Spatial Estimation," IEEE Trans. Autom. Control, **54**(12), pp. 2816–2827.
[24] Graham, R., and Cortés, J., 2009, "Cooperative Adaptive Sampling of Random Fields With Partially Known Co-Variance," Int. J. Rob. Nonlinear Control, **22**(5), pp. 504–534.
[25] Graham, R., and Cortés, J., 2012, "Adaptive Information Collection by Robotic Sensor Networks for Spatial Estimation," IEEE Trans. Autom. Control, **57**(6), pp. 1404–1419.
[26] Xu, Y., Choi, J., Dass, S., and Maiti, T., 2012, "Sequential Bayesian Prediction and Adaptive Sampling Algorithms for Mobile Sensor Networks," IEEE Trans. Autom. Control, **57**(8), pp. 2078–2084.
[27] Xu, Y., and Choi, J., 2012, "Spatial Prediction With Mobile Sensor Networks Using Gaussian Processes With Built-In Gaussian Markov Random Fields," Automatica, **48**(8), pp. 1735–1740.
[28] Xu, Y., and Choi, J., 2012, "Stochastic Adaptive Sampling for Mobile Sensor Networks Using Kernel Regression," Int. J. Control Autom. Syst., **10**(4), pp. 778–786.
[29] Varagnolo, D., Pillonetto, G., and Schenato, L., 2012, "Distributed Parametric and Nonparametric Regression With On-Line Performance Bounds Computation," Automatica, **48**(10), pp. 2468–2481.
[30] Samson, C., and Ait-Abderrahim, K., 1990, "Mobile Robot Control, Part 1: Feedback Control of a Nonholo-Nomic Wheeled Cart in Cartesian Space," Institut National de Recherche en Informatique et en Automatique, Report 1288, Le Chesnay, France.
[31] Aicardi, M., Casalino, G., Bicchi, A., and Balestrino, A., 1995, "Closed Loop Steering of Unicycle Like Vehicles via Lyapunov Techniques," IEEE Rob. Autom. Mag., **2**(1), pp. 27–35.
[32] Ren, W., and Beard, R., 2004, "Trajectory Tracking for Unmanned Air Vehicles With Velocity and Heading Rate Constraints," IEEE Trans. Control Syst. Technol., **12**(5), pp. 706–716.
[33] Anderson, R., Bakolas, E., Milutinović, D., and Tsio-tras, P., 2013, "Optimal Feedback Guidance of a Small Aerial Vehicle in a Stochastic Wind," J. Guidance Control Dyn., **36**(4), pp. 975–985.
[34] Sutton, R. S., and Barto, A. G., 1998, *Reinforcement Learning: An Introduction*, Vol. 1, The MIT Press, Cambridge, MA.
[35] Powell, W. B., 2007, *Approximate Dynamic Programming: Solving the Curses of Dimensionality*, Vol. 703, John Wiley & Sons, Hoboken, NJ.
[36] Long, A. W., Wolfe, K. C., Mashner, M. J., and Chirikjian, G. S., 2012, "The Banana Distribution is Gaussian: A Localization Study With Exponential Coordinates," Proceedings of Robotics: Science and Systems, Syndey, Australia, pp. 265–272.
[37] Thrun, S., Burgard, W., and Fox, D., 2005, *Probabilistic Robotics*, The MIT Press, Cambridge, MA.
[38] Anderson, R. P., and Milutinović, D., 2011, "A Stochastic Approach to Dubins Feedback Control for Target Tracking," Proceedings of the 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), San Francisco, CA, pp. 3917–3922.
[39] Jadaliha, M., Lee, J., and Choi, J., 2012, "Adaptive Control of Multiagent Systems for Finding Peaks of Uncertain Static Fields," J. Dyn. Syst. Meas. Contr., **134**(5), p. 051007.
[40] Cressie, N., 1986, "Kriging Nonstationary Data," J. Am. Stat. Assoc., **81**(395), pp. 625–634.
[41] Rasmussen, C. E., and Williams, C. K. I., 2006, *Gaussian Processes for Machine Learning*, The MIT Press, Cambridge, MA.
[42] Choi, J., Lee, J., and Oh, S., 2008, "Biologically-Inspired Navigation Strategies for Swarm Intelligence Using Spatial Gaussian Processes," Proceedings of the 17th International Federation of Automatic Control (IFAC) World Congress, Seoul, Korea.
[43] Choi, J., Lee, J., and Oh, S., 2008, "Swarm Intelligence for Achieving the Global Maximum Using Spatio-Temporal Gaussian Processes," Proceedings of the 27th American Control Conference (ACC), Seattle, WA, pp. 135–140.
[44] Shi, J. Q., and Choi, T., 2011, *Gaussian Process Regression Analysis for Functional Data*, CRC, Boca Raton, FL.
[45] Abrahamsen, P., 1997, *A Review of Gaussian Random Fields and Correlation Functions*, Norsk Regnesentral/Norwegian Computing Center, Oslo, Norway.
[46] Snelson, E. L., 2007, "Flexible and Efficient Gaussian Process Models for Machine Learning," Ph.D. thesis, University College, London, UK.
[47] Zhang, B., and Sukhatme, G., 2007, "Adaptive Sampling for Estimating a Scalar Field Using a Robotic Boat and a Sensor Network," 2007 IEEE International Conference on Robotics and Automation, Rome, Italy, Apr. 10–14, pp. 3673–3680.
[48] Laut, J., Henry, E., Nov, O., and Porfiri, M., 2014, "Development of a Mechatronics-Based Citizen Science Platform for Aquatic Environmental Monitoring," IEEE/ASME Trans. Mechatron., **19**(5), pp. 1541–1551.
[49] Xu, Y., and Choi, J., 2011, "Adaptive Sampling for Learning Gaussian Processes Using Mobile Sensor Networks," Sensors, **11**(3), pp. 3051–3066.
[50] Bishop, C. M., 2006, *Pattern Recognition and Machine Learning*, Springer, New York.
[51] Xu, Y., Choi, J., Dass, S., and Maiti, T., 2011, "Bayesian Prediction and Adaptive Sampling Algorithms for Mobile Sensor Networks," Proceedings of the 2011 American Control Conference (ACC), San Francisco, CA, pp. 4095–4200.
[52] Xu, Y., 2011, "Environmental Adaptive Sampling for Mobile Sensor networks using Gaussian processes," Ph.D. thesis, Michigan State University, East Lansing, MI.
[53] Gaudard, M., Karson, M., Linder, E., and Sinha, D., 1999, "Bayesian Spatial Prediction," Environ. Ecol. Stat., **6**(2), pp. 147–171.
[54] Rue, H., and Tjelmeland, H., 2002, "Fitting Gaussian Markov Random Fields to Gaussian Fields," Scand. J. Stat., **29**(1), pp. 31–49.
[55] Cressie, N., and Verzelen, N., 2008, "Conditional-Mean Least-Squares Fitting of Gaussian Markov Random Fields to Gaussian Fields," Comput. Stat. Data Anal., **52**(5), pp. 2794–2807.
[56] Hartman, L., and Hössjer, O., 2008, "Fast Kriging of Large Data Sets With Gaussian Markov Random Fields," Comput. Stat. Data Anal., **52**(5), pp. 2331–2349.
[57] Le Ny, J., and Pappas, G. J., 2009, "On Trajectory Optimization for Active Sensing in Gaussian Process Models," Proceedings of the 48th IEEE Conference on Decision and Control, Shanghai, China, pp. 6286–6292.
[58] Rasmussen, C. E., and Nickisch, H., 2010, "Gaussian Processes for Machine Learning (GPML) Toolbox," J. Mach. Learn. Res., **11**, pp. 3011–3015.
[59] Minka, T. P., 2001, "Expectation Propagation for Approximate Bayesian Inference,". Proceedings of the Seventeenth Conference on Uncertainty in Artificial Intelligence, Morgan Kaufmann, pp. 362–369.
[60] Williams, C. K., and Barber, D., 1998, "Bayesian Classification With Gaussian Processes," IEEE Trans. Pattern Anal. Mach. Intell., **20**(12), pp. 1342–1351.