

Article

Learning Environmental Field Exploration with Computationally Constrained Underwater Robots: Gaussian Processes Meet Stochastic Optimal Control

Daniel Andre Duecker ^{1,*},[†] , Andreas Rene Geist ^{1,2,†}, Edwin Kreuzer ¹ and Eugen Solowjow ³

¹ Institute of Mechanics and Ocean Engineering, Hamburg University of Technology, 21073 Hamburg, Germany; geist@is.mpg.de (A.R.G.); kreuzer@tuhh.de (E.K.)

² Max Planck Institute for Intelligent Systems, 70569 Stuttgart, Germany

³ Siemens Corporate Technology, Berkeley, CA 94704, USA; eugen.solowjow@siemens.com

* Correspondence: daniel.duecker@tuhh.de; Tel.: +49-404-2878-4896

† These authors contributed equally to this work.

Received: 4 April 2019; Accepted: 27 April 2019; Published: 6 May 2019



Abstract: Autonomous exploration of environmental fields is one of the most promising tasks to be performed by fleets of mobile underwater robots. The goal is to maximize the information gain during the exploration process by integrating an information-metric into the path-planning and control step. Therefore, the system maintains an internal belief representation of the environmental field which incorporates previously collected measurements from the real field. In contrast to surface robots, mobile underwater systems are forced to run all computations on-board due to the limited communication bandwidth in underwater domains. Thus, reducing the computational cost of field exploration algorithms constitutes a key challenge for in-field implementations on micro underwater robot teams. In this work, we present a computationally efficient exploration algorithm which utilizes field belief models based on Gaussian Processes, such as Gaussian Markov random fields or Kalman regression, to enable field estimation with constant computational cost over time. We extend the belief models by the use of weighted shape functions to directly incorporate spatially continuous field observations. The developed belief models function as information-theoretic value functions to enable path planning through stochastic optimal control with path integrals. We demonstrate the efficiency of our exploration algorithm in a series of simulations including the case of a stationary spatio-temporal field.

Keywords: autonomous exploration; environmental field monitoring; gaussian processes; gaussian markov random fields; kalman filtering; stochastic optimal control

1. Introduction

Autonomous underwater field exploration has been a fast-growing research area in the last decade. With continuous advances in small-scale computing technology, smart micro-robots are expected to play a prominent role in increasingly autonomous and interconnected exploration and monitoring systems [1]; examples of micro underwater robots include the Avexis [2] and the HippoCampus [3] micro autonomous underwater vehicle, see Figure 1. Autonomous multi-agent swarms can be deployed for maritime exploration tasks such as the monitoring of algae growth, oil spill, or underwater currents.

Hereby, the development of maritime exploration and monitoring systems profits from many synergies with research on similar onshore tasks such as the monitoring of urban environmental fields. As a result, micro-robots can monitor environmental processes such as particulate matter, acoustic pollution, or electromagnetic fields. Particularly in hazardous environments, autonomous micro-robots

inherit great potential to drastically increase safety and reduce costs compared to concepts involving the deployment of stationary sensors or humans taking measurements. Thereby, depending on the dynamics of the specific robotic platform sophisticated control algorithms such as [4,5] are required in order to allow a safe and robust deployment in the underwater domain. However, missions with mobile underwater robots inherit the challenge of limited communication bandwidth and range. This naturally enforces a high level of autonomy as all key components of the exploration algorithm have to run onboard.

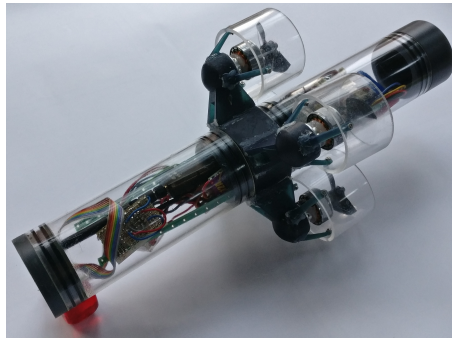


Figure 1. Micro underwater robot HippoCampus for environmental field exploration.

Systems for maritime exploration tasks consist of multiple building blocks as depicted in Figure 2. The state of the environment is represented in a so-called belief model which is updated through measurements of the environment. The observations are collected using mobile robots. Thus, a path-planning algorithm is required in order to maximize the expected information gain through future observations along the robots' paths. Since the robots cannot simply take measurements at arbitrary locations, a trade-off arises between the potential information gained and the effort to drive the robots to regions where information-rich observations can be collected.

1.1. Related Work

Methods for modeling a robot's environmental belief can be distinguished in physics-based models and non-physics based models. Physics-based models allow the extrapolation of the model to the vicinity of the known belief. Nonetheless, they require solving a partial differential equation (e.g., Navier–Stokes equation), which is computationally demanding. Moreover, such systems require information regarding the boundary conditions which is often not directly available to the system [6]. In recent years, probabilistic belief models have been developed as a promising alternative for efficient field estimation. These data-driven models may be more suitable for computationally constrained robot swarms since they avoid computational-costly solving the partial differential equation and additionally allow to model the uncertainty of the estimated physical process directly. Moreover, data-driven models inherit the potential to infer the process characteristics during operation.

A prominent inference method for learning environmental field beliefs is Gaussian process (GP) regression. Gaussian process regression, or in geo-statistics terminology 'Kriging', originated as a method for statistical inference of ore concentration fields [7]. Modeling environmental fields with GPs is attractive as their mean and covariance functions allow a spatial continuous field representation while additionally providing statistical uncertainty measures. Example applications include the estimation of a temperature field in lakes [8] and the reconstruction of spatio-temporal fields with mobile sensor nodes [9]. Moreover, Xu et al. [10] present a broad variety of different GP models for spatial field estimation.

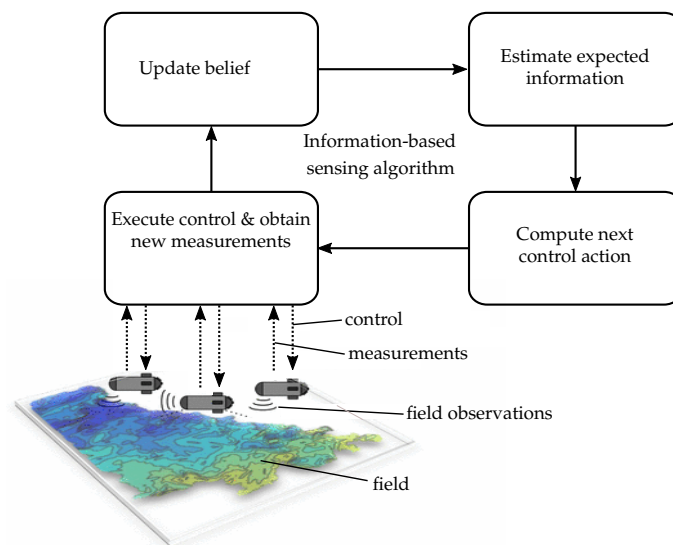


Figure 2. Components of a closed-loop, information-based sensing algorithm for field exploration with autonomous underwater robots.

While the continuous specification of a GP provides an intuitive interpretation of the underlying physical process, real-world applications of GPs are often hindered by the big n problem. The big n problem of GPs is particularly problematic when dealing with multidimensional dynamical processes. Although several methods have been proposed to overcome this constraint, e.g., by reducing the dimensions, simplifying the structure of the covariance matrix [11], or truncating the observations [9], these methods inevitably inherit the trade-off between approximation quality and computational cost. The computational burden of GPs becomes especially challenging and often intractable when resource-constrained underwater robots are involved. In contrast to surface scenarios, high-bandwidth communication links are often not available in the submerged domain. The low-bandwidth communication requires the mobile robots to maintain a decentralized field belief representation on-board. Recently, it has been shown that GPs can be sufficiently approximated through Gaussian Markov random fields (GMRFs). A GMRF approximates a GP on a predefined lattice of random variables (RV) by utilizing the Markov property [12]. Xu et al. discussed in a series of publications the suitability of GMRFs for mobile sensor networks [10,13]. In their recent work, Jдалиha et al. [14] present an extended GMRF framework for mobile robots which incorporates uncertainties in the observation location. In an alternative approach for circumventing the big n problem, Todescato et al. [15] combine Kalman filtering techniques with GPs to update a spatio-temporal field representation.

Regarding the task of efficient information gathering, planning algorithms can be grouped into four categories. *Myopic planning* algorithms are computationally efficient as they compute the next best action without a planning horizon. However, they suffer from the risk of getting stuck in local optima. *Sampling-based* strategies have gained increasing interest in recent years, but do not provide guarantees on global optimality; examples include the popular rapidly exploring random tree methods. *Dynamic programming* is the method of choice for informative path planning tasks which can be formulated as a (partially observable) Markov decision problem. However, they are often intractable if the exploration task shall be solved without a state and domain discretization. In their work [16] Hollinger and Sukhatme examine sampling-based information-gathering algorithms for continuous space. They include information quality metrics and motions constraints in their planning algorithms. Marchant and Ramos [17] present a double layered informative path planning concept using GPs and Bayesian optimization, whereby they use one GP to model the physical phenomenon and a second GP to model the quality of the selected paths. The resulting paths are described through cubic splines. In [18], the exploration problem is formulated as a POMDP. The authors use Monte-Carlo tree search and an upper confidence bound for trees together with sequential Bayesian optimization techniques.

They evaluate their method in a series of experiments to analyze the performance. Recently, Cui et al. proposed the combination of mutual information and rapidly-exploring random trees for underwater path planning in a scalar field sampling scenario [19].

In this context, many publications consider the scenario of exploring large-scale environments such as oceans. However, when reducing the scale to confined environments such as industrial tanks, the robot dynamics cannot be neglected anymore. This problem can be tackled using *receding horizon* schemes that optimize the full planning horizon while not providing guarantees beyond the horizon. However, such a planning algorithm require a continuous representation of the field belief. Kreuzer and Solowjow use linear functions to interpolate the field belief representation between the grid-points while Xu et al. [20] propose sinusoidal weighting functions which are an attractive nonlinear alternative for smooth field interpolation. The computational costs of these belief algorithms increase with the number of collected observations, rendering their application on real robots impractical. With a continuous belief representation, an exploratory path can be computed using the policy improvement with path integrals (PI²) algorithm proposed by [21]. The PI² algorithm origins from the work on solving a nonlinear stochastic optimal control via the use of path integrals [22]. In reinforcement learning terminology, our resulting exploration method could be seen as performing policy iteration. In which, first, the exploration policy is evaluated at each new location through an update of the field belief model. Subsequently, the belief is used as an information-theoretic value function for policy improvement as proposed by [23]. This approach enables rapid evaluation of the belief space for finding an optimal informative path.

1.2. Contributions

The contribution of this work is two-fold. First, we combine and extend approaches from previous works [13,23] on field exploration with GMRFs to meet the requirements of an application on a micro-robot platform and on the PI² stochastic path planning algorithm. These requirements are

- a constant computational complexity over time,
- a continuous spatial belief representation which allows efficient path planning.

Therefore, we first extend the PI-GMRF approach proposed in [23] with the concept of sequential Bayesian spatial prediction in [13] to guarantee a constant computational load while maintaining the ability to incorporate off-grid measurements. Second, we extend the recently presented spacetime Kalman filter (STKF) [15] by incorporating the concept of weighted shape functions to render the belief model compatible with PI². The combination of these two different belief models with PI² results in two novel algorithms for stochastic field exploration—namely, an improved PI-GMRF and the new PI-STKF.

We conduct two numerical experiments to compare these models regarding their ability to efficiently explore environmental fields and computational complexity. We assume that sufficient prior knowledge on the physical process is available and thus *no hyperparameter estimation* of the belief model is required. For the sake of brevity, we present the field estimation schemes for a *single robot*. Albeit, we derive the algorithms in a form that allows a direct extension to multiple agents sharing a centralized belief model.

1.3. Paper Structure

The paper is structured as follows. In Section 2, we briefly outline the problem of autonomous field exploration. In Section 3, the belief models, namely the fully Bayesian GMRF and STKF, are introduced and extended to incorporate off-grid measurements. In Section 4, we elaborate on the usage of PI² for path planning using information-theoretic value functions. In Section 5, the two belief models are compared with respect to their computational time and estimation results. In Section 6, we analyze the exploration performance of the final algorithm on an unknown spatio-temporal field. Finally, Section 7 concludes this work and highlights potential future research directions.

2. Problem Statement

We consider an autonomous underwater vehicle which explores an environmental field through point-wise observations in a confined environment. The goal is to minimize the error between the estimated and the true field within a minimal amount of time. Regarding the exploration algorithm, this task can be restated as maximizing the ratio of collected information per time step. To make this task tractable for mobile robot teams, the computational cost of the exploration algorithm has to be bounded over time.

2.1. Robot Model and Problem Formulation

A dynamical robot model allows sampling feasible exploratory trajectories which can be evaluated regarding their potential information gain on the environmental field belief. The motion of each robot is described by

$$\dot{x}_t = f_{\text{robot}}(x_t) + G(x_t)u_t, \quad (1)$$

where $x_t \in \mathbb{R}^n$ is the state of the robot. The passive dynamics f_{robot} define the state transition, which for path planning algorithms is commonly described through a simple particle model. The scaled control input $u_t \in \mathbb{R}^p$ is the computed state correction through the robot's actuators and $G(x_t) \in \mathbb{R}^{n \times p}$ the control matrix.

At each discrete time step, the robot collects measurements $y(x_t)$ to gather information about the environmental field $f(x_t)$ of interest. The sensor model of a single robot is described by

$$y(x_t) \sim h(f(x_t), v), \text{ with } v \sim \mathcal{N}(0, \Sigma_z), \quad (2)$$

where z_{real} depicts the real environmental field values, $h(\cdot)$ is the observation function, and Σ_y is the diagonal covariance matrix representing the measurement noise.

2.2. Field Belief Representation

In order to efficiently learn an environmental field estimate the robots have to gather information by taking measurements and infer knowledge from the collected data. The collected data is mapped to a belief of the underlying environmental process that enables the agents to plan actions. In the information-theoretic exploration algorithms, the second order moments of the field belief representation are used to evaluate the quality of the planned exploration paths. In this context, the concept of probabilistic kernel models is a natural choice for spatially correlated field values. The correlation of the field values is assumed to be known a priori. Thus, no hyperparameter estimation has to be performed during exploration.

The limited computational capacities of embedded systems such as mobile robots require the belief model to have a comparably small and constant computational cost. Therefore, we extend algorithms based on GMRFs and Kalman Filtering to limit their computational complexity and to enable their combination with a stochastic path planning controller for exploration tasks.

A GMRF defines a (finite-dimensional) random field that follows a multivariate Gaussian distribution while satisfying the Markov property. Due to the Markov assumption, the inverse of the covariance matrix Λ can be defined on a predefined lattice, while also being (desirably) sparsely populated. The GMRF is fully defined by a mean vector μ and a precision matrix Λ^{-1} as $\mathcal{N}(\mu, \Lambda^{-1})$. GMRFs are well suited for the approximation of conditional auto-regressive processes, but require the initialization of a fixed lattice of random variables (RVs). The initialization hinders the application of GMRFs to model temporal process correlations. Therefore, a Kalman regression algorithm, also referred to as spacetime Kalman Filtering (STKF), is utilized to express a belief of the spatio-temporal environmental field. However, both stochastic field belief models provide a measure of belief uncertainty in the form of the conditional variance.

2.3. Stochastic Optimal Control Problem

The stochastic optimal controller allows computing an optimal path with respect to the given value function. In this work, we use a field uncertainty based value function, whereby the field uncertainty is computed from the previously developed belief model. We use a stochastic optimal controller to plan a maximal informative path with respect to the belief model's predicted variance. As thoroughly discussed in [24], the predicted variance results in the conditional entropy, which is an indirect informative criterion. Hereby, the term indirect informative means that only the uncertainty of the next potential state is considered, while the information of surrounding field values is not considered. A critical insight for information theoretic path planning is that the posterior variance does not depend on the process values of the obtained measurements if the kernel function is known.

In order to sample potentially feasible paths, the following robot dynamics are considered

$$\dot{\mathbf{x}}_t = \mathbf{f}_{\text{robot}}(\mathbf{x}_t) + \mathbf{G}(\mathbf{x}_t)(\mathbf{u}_t + \boldsymbol{\varepsilon}_t), \quad (3)$$

where $\mathbf{x}_t \in \mathbb{R}^{n \times 1}$ denotes the system state, $\mathbf{f}_{\text{robot}}(\mathbf{x}_t) \in \mathbb{R}^{n \times 1}$ the passive dynamics and $\boldsymbol{\varepsilon}_t \in \mathbb{R}^{p \times 1}$ the additive Gaussian noise with variance $\boldsymbol{\Sigma}_{\boldsymbol{\varepsilon}}$. Moreover, let the index t denote any arbitrary time step, while we use the index t_i to emphasize a particular time. The final goal of a stochastic optimal controller is to compute the optimal controls \mathbf{u}_t with respect to the performance functional

$$V(\mathbf{x}_{t_i}) = V_{t_i} = \min_{\mathbf{u}_{i:(H-1)}} \mathbb{E}_{\boldsymbol{\tau}_{i:H}} (R(\boldsymbol{\tau}_{i:H})). \quad (4)$$

The expectation $\mathbb{E}_{\boldsymbol{\tau}_{i:H}}$ is taken over all trajectories starting at \mathbf{x}_{t_i} . Also $t_0 = 0$ s denotes the time at the current agent position, and t_H the last time step of the control horizon. We define the finite horizon cost function $R_{\boldsymbol{\tau}_{i:H}}$ for a trajectory piece $\boldsymbol{\tau}_{i:H}$ with start at time t_i and end at t_H as

$$R_{\boldsymbol{\tau}_{i:H}} = \boldsymbol{\phi}_{t_H} + \int_{t_i}^{t_H} r_t dt. \quad (5)$$

The term $\boldsymbol{\phi}_{t_H}$ denotes a terminal reward at time t_H . The immediate cost r_t at time t is chosen as

$$r_t = r(\mathbf{x}_t, \mathbf{u}_t, t) = q_t + \frac{1}{2} \mathbf{u}_t^\top \mathbf{R} \mathbf{u}_t, \quad (6)$$

with $q_t = q(\mathbf{x}_t, t)$ being a state-dependent cost function, and \mathbf{R} being a positive semi-definite weight matrix. Note that the control action \mathbf{u}_t is linear in (3) and quadratic in (6).

3. Probabilistic Belief Modeling for Field Exploration

In this section, we present extensions to two existing field belief concepts, namely the GMRF and the STKF approach, to allow information-based exploration control with constant computational cost over time. Both belief algorithm can be used to estimate a stochastic process on a predefined lattice of Gaussian random variables.

In order to enable the incorporation of the belief models into a stochastic optimal control exploration framework, we extend the belief algorithms analog to [23]. Therefore, we incorporate off-grid observations through an affine transformation of field measurements onto the belief grid using spatial shape functions. However, the original concept presented in [23] does not fulfill the requirement of constant computational cost over time.

The GMRF-based belief algorithm was originally proposed in [13] and enables efficient estimation of stationary spatial processes on a discrete grid of Gaussian random variables. The second belief algorithm, proposed in [15], combines GP regression of spatial process components with Kalman filtering of conditional-auto-regressive temporal process components.

Defining the field representation on a lattice raises the question on how to choose the grid discretization based on the fundamental trade-off between the accuracy of the field representation versus the available computational power. The latter aspect is of particular importance in the field of underwater robotics, as off-board computation of the field representation is not feasible due to the very limited communication bandwidth. Thus, the discretization step size has to be selected depending on the actual application scenario. Thereby, prior knowledge on the spatial scale of the physical process is a helpful and valid assumption. For instance, if the user aims to explore small scale processes in a local environment, e.g., an industrial tank, a dense grid is likely to be a better choice than a coarse grid which captures global physical phenomena with acceptable computational burden. Moreover, shape function approximation can be used to interpolate the field belief between the grid points. This allows to efficiently monitor large scale fields where the main interest lies in the exploration of global phenomena rather than local small scale processes.

3.1. Shape Functions

The introduced belief algorithms estimate the field on a discrete lattice $\{\mathcal{V}, \mathcal{E}\}$ with vertices $\mathcal{V} = \{1, \dots, n\}$ and edges \mathcal{E} . The set of continuous field locations is discretized into a finite subset of n spatial input locations $\mathcal{S} = \{s_1, \dots, s_n\}$, such that the vector $f(t) = [f(s_1, t), \dots, f(s_n, t)]^\top \triangleq [f_{1,t}, \dots, f_{n,t}]^\top$ is a discretization of $f(x, t)$. The lattice \mathcal{S} consists of a finite number of sub-domains $\mathcal{S}_{e,i}$, where each is enclosed by four vertices \bar{s}_i , with $i = 1, \dots, 4$. For the ease of illustration, \mathcal{S} is chosen as regular lattice with edges each of length a and b respectively, as depicted in Figure 3. As proposed by [23], the field value at position q can be approximated through a sum of weighted shape functions $\phi_i^e(q)$ and field values on the vertices $f(\bar{s}_i)$, such that

$$\mathcal{F}^e = \sum_{i=1}^4 \phi_i^e(q) f(\bar{s}_i) \tag{7}$$

Figure 3 illustrates shape function $\phi_{i=4}^e$ on an element domain \mathcal{F}_e . Each shape function is zero outside its corresponding element and equal to one at the associated vertex

$$\phi_i^e(\bar{s}_k) = \begin{cases} 1, & \text{if } i = k \\ 0, & \text{otherwise} \end{cases} \tag{8}$$

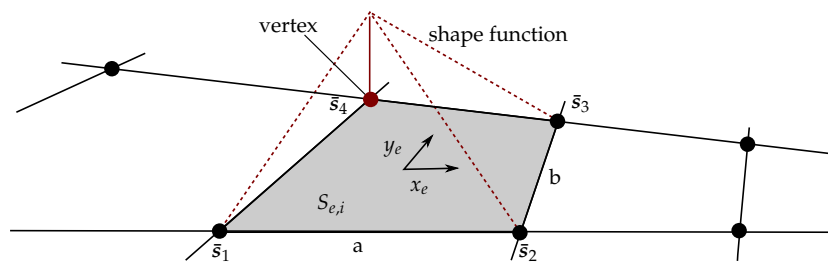


Figure 3. Shape function on a selected grid element.

A local coordinate system $\mathcal{K}_e(x_e, y_e)$ is defined on \mathcal{F}_e . The origin of $\mathcal{K}_e(x_e, y_e)$ lies in the center of the element. The corresponding coordinate axis are orthogonal to each other and parallel to the respective element edges. The shape functions are defined as

$$\begin{aligned} \phi_1^e &= \frac{1}{ab} \left(x_e - \frac{a}{2}\right) \left(y_e - \frac{b}{2}\right), & \phi_2^e &= -\frac{1}{ab} \left(x_e + \frac{a}{2}\right) \left(y_e - \frac{b}{2}\right), \\ \phi_3^e &= \frac{1}{ab} \left(x_e + \frac{a}{2}\right) \left(y_e + \frac{b}{2}\right), & \phi_4^e &= -\frac{1}{ab} \left(x_e - \frac{a}{2}\right) \left(y_e + \frac{b}{2}\right). \end{aligned} \tag{9}$$

Using the shape functions above, ϕ_j defines a mapping between the continuous field and four-element vertices through

$$\phi_j = [0 \dots \phi_1^e \phi_2^e \phi_3^e \phi_4^e \dots 0], \quad \phi_j \in [0, 1]^{1 \times n}. \tag{10}$$

In this manner, a measurement $y_j(t_k)$ can be expressed in terms of the element grid approximation, writing

$$y_j(\mathbf{q}_j) = \phi_j(\mathbf{q}_j)\mathbf{f} + v, \quad v \sim \mathcal{N}(0, \sigma_y^2). \tag{11}$$

Moreover, the mapping of N measurements at time t_k is obtained as $\Phi_k = [\phi_1(t_k)^\top, \dots, \phi_N(t_k)^\top]^\top$. Further we define the mapping $\Phi_{1:k} = [\Phi_1^\top, \dots, \Phi_k^\top]^\top$.

3.2. Gaussian Markov Random Field Regression

Gaussian Markov random fields define a conditional auto-regressive (CAR) process. A process is a CAR(j) process, if the expectation of a process value is fully defined through the next j adjacent graph vertices. The Markov assumption enables the direct construction of a sparse precision matrix. Given a labeled graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with vertices $\mathcal{V} = \{1, \dots, n\}$ and edges \mathcal{E} , a probabilistic graphical model η defines a GMRF, if the edges \mathcal{E} are chosen such that there is no edge between node i and j , if $\eta_i \perp \eta_j \mid \eta_{-ij}$, in which $-ij$ denotes the nodes adjacent to i and j , respectively [25]. The pairwise conditional independence properties of x on \mathcal{G} are inherent in the subdiagonal entries of the precision matrix Λ . We refer the reader to [25] for an in-depth discussion of GMRFs.

In order to construct the GMRF, the continuously indexed spatial field $\mathcal{F}^* \subset \mathbb{R}^d$ is discretized into a labeled undirected spatial graph with n^* vertex positions $S^* = x_1, \dots, x_{n^*}$, where x_i denotes the i -th field vertex position (Note that in this work the scalars x or y denote the spatial position coordinates of a two-dimensional spatial position vector x , while a bold y represents an environmental field observation vector.). The set of field locations S^* is extended to S with vertex positions $S = x_1, \dots, x_n$, as depicted in Figure 4, to compensate boundary effects occurring due to the GMRF approximation. Then on S , a GMRF η is constructed using the SPDE approach proposed by [12].

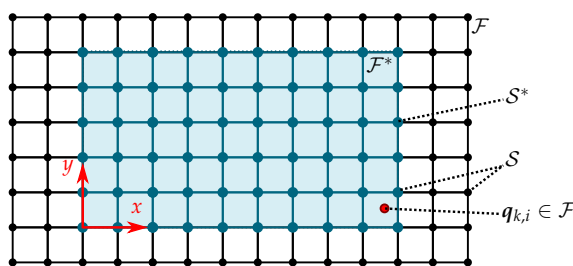


Figure 4. The agent’s position $q_{k,i}$ at the discrete time step k lies in a spatial field \mathcal{F}^* with coordinate values x and y . The field \mathcal{F}^* can be extended to \mathcal{F} and is then discretized into a regular grid S to enable the construction of a GMRF and to compensate for boundary effects.

Let $\eta \sim \mathcal{N}(0, \Sigma)$ be a GP on \mathbb{R}^2 defined by the Matérn covariance function defined as

$$k_{\text{Matérn}}(\mathbf{x}, \mathbf{x}') = \sigma_f^2 \frac{2^{1-\nu}}{\Gamma(\nu)} \left(\kappa \|\mathbf{x} - \mathbf{x}'\| \right)^\nu K_\nu \left(\kappa \|\mathbf{x} - \mathbf{x}'\| \right), \tag{12}$$

in which $\|\cdot\|$ denotes the Euclidean distance in \mathbb{R}^d and K_ν the modified Bessel function of the second kind. The GMRF $\boldsymbol{\eta} \sim \mathcal{N}(0, \boldsymbol{\Lambda}^{-1})$ defined on a regular two-dimensional lattice approximates a Matérn GP for $\nu = 0$ if the Gaussian full conditionals read

$$\mathbb{E}(\boldsymbol{\eta}|\boldsymbol{\eta}_{-ij}, \boldsymbol{\theta}) = \frac{1}{a} (\eta_{i-1,j} + \eta_{i+1,j} + \eta_{i,j-1} + \eta_{i,j+1}) = \frac{1}{a} \begin{pmatrix} \circ & \bullet & \circ \\ \bullet & \bullet & \bullet \\ \circ & \bullet & \circ \end{pmatrix}, \text{Pre}(\boldsymbol{\eta}|\boldsymbol{\eta}_{-ij}, \boldsymbol{\theta}) = a \tau. \quad (13)$$

For the case of $\nu = 1$, the Gaussian full conditionals read

$$\mathbb{E}(\boldsymbol{\eta}|\boldsymbol{\eta}_{-ij}, \boldsymbol{\theta}) = \frac{1}{4+a^2} \begin{pmatrix} \circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ \\ \circ & \circ & \bullet & \circ & \circ & \circ & \bullet & \circ & \circ & \circ & \circ & \circ \\ 2a & \circ & \bullet & \circ & \bullet & \circ & \circ & \circ & \circ & \circ & -1 & \bullet & \circ & \circ & \bullet \\ \circ & \circ & \bullet & \circ & \circ & \circ & \bullet & \circ & \bullet & \circ & \circ & \circ & \circ & \circ & \circ \\ \circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ & \bullet & \circ & \circ & \circ \end{pmatrix}, \text{Pre}(\boldsymbol{\eta}|\boldsymbol{\eta}_{-ij}, \boldsymbol{\theta}) = (4+a^2) \tau, \quad (14)$$

with $a = \kappa^2 + 4$ and $\boldsymbol{\theta} = [\tau, \kappa]^\top \in \mathbb{R}_{>0}^2$ being hyperparameters of the model. The additional hyperparameter τ adjusts the GMRF's signal variance independent of κ . The proof of Equations (13) and (14) for the general case of irregular grids is stated in [12]. Figure 5 illustrates the correspondence between the spatial lattice locations and the values in each column of $\boldsymbol{\Lambda}$ using the previously presented construction scheme.

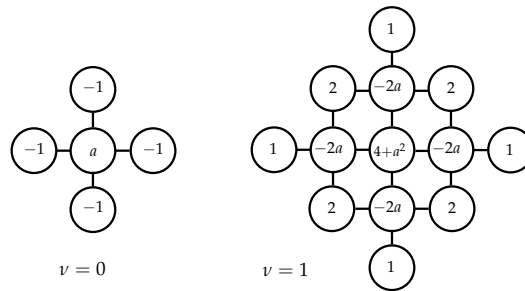


Figure 5. Non-zero elements of a column of the precision matrix $\boldsymbol{\Lambda}$ associated with one random variable and its neighbor vertices on a regular two-dimensional GMRF lattice.

When designing the GMRF precision matrix, the full conditionals for the border vertices of the GMRF grid affect the estimation result considerably. Three commonly used boundary conditions are the Dirichlet, Neumann, and torus boundary condition [25].

3.2.1. Sequential GMRF Regression

In this Subsection, the GMRF regression algorithm proposed in [13] is extended to enable spatial process estimation with continuous observations. The values of the field are represented by the latent variable $z_i = z(\mathbf{s}_i) \in \mathbb{R}$. The latent variables are expressed using a global linear model, such that

$$z_i = \mu(\mathbf{s}_i, \boldsymbol{\beta}) + \eta_i \quad \forall 1 \leq i \leq n, \quad (15)$$

$$\mu(\mathbf{s}_i, \boldsymbol{\beta}) = \mathbf{m}^\top \boldsymbol{\beta}. \quad (16)$$

Hereby, $\mathbf{m} = [m_1(\mathbf{s}_i), \dots, m_p(\mathbf{s}_i)]^\top \in \mathbb{R}^p$ denotes the regression function vector and the vector $\boldsymbol{\beta} = [\beta_1, \dots, \beta_p]^\top$ contains the unknown regression coefficients. The field belief on the lattice is denoted as $\mathbf{z} = [z_1, \dots, z_n]^\top$. The small-scale correlations of the field are modeled through the zero-mean GMRF $\boldsymbol{\eta} \sim \mathcal{N}(0, \boldsymbol{\Lambda}_\eta|\boldsymbol{\theta})$. We initialize the GMRF precision matrix $\boldsymbol{\Lambda}_\eta^{-1}$ with the full conditionals as defined in (13) and (14). A zero-mean Gaussian prior is assumed on the regression coefficients $\boldsymbol{\beta} \sim \mathcal{N}(0, \mathbf{T}^{-1})$ to estimate the regression coefficients as a function of \mathbf{z} and $\boldsymbol{\theta}$, where \mathbf{T}^{-1} is initialized

as a diagonal matrix with large diagonal elements. The probability distribution of the full latent field $\bar{z} = [z^\top, \beta^\top]^\top \in \mathbb{R}^{n+p}$ reads

$$\begin{aligned} p(\bar{z}, \theta) &= p(z|\beta, \theta) p(\beta), \\ &\propto \exp\left(-\frac{1}{2}(z - m\beta)^\top \Lambda_{\eta|\theta}(z - m\beta) - \frac{1}{2}\beta^\top T\beta\right), \\ &= \exp\left(-\frac{1}{2}\bar{z}^\top \Lambda_{\bar{z}|\theta}\bar{z}\right), \end{aligned} \quad (17)$$

with precision matrix $\Lambda_{\bar{z}|\theta} \in \mathbb{R}^{(n+p) \times (n+p)}$ being defined as

$$\Lambda_{\bar{z}|\theta} = \begin{bmatrix} \Lambda_{\eta|\theta} & -\Lambda_{\eta|\theta}m \\ -m^\top \Lambda_{\eta|\theta} & m^\top \Lambda_{\eta|\theta}m + T \end{bmatrix}. \quad (18)$$

By exploiting the GMRF's *canonical form*, only the current available measurements y_k are required to sequentially update the conditional precision matrix $\Lambda_{\bar{z}|\theta, y_{1:k}} \triangleq \Lambda_{k|\theta}$ and the *canonical mean* $b_k = \Lambda_{k|\theta}^{-1} \mu_{k|\theta}$. A sequential updating algorithm is obtained by inserting the canonical mean definition into the formula for conditioning of a multivariate Gaussian distribution, such that

$$p(\bar{z}|\theta, y_{1:k}) = \mathcal{N}(\Lambda_{k|\theta}^{-1} b_k, \Lambda_{k|\theta}^{-1}), \quad (19)$$

$$\Lambda_{k|\theta} = \Lambda_{\bar{z}|\theta} + \frac{1}{\sigma_y^2} \Phi_{1:k}^\top \Phi_{1:k} = \Lambda_{k-1|\theta} + \frac{1}{\sigma_y^2} \Phi_k^\top \Phi_k, \quad (20)$$

$$b_k = \frac{1}{\sigma_y^2} \Lambda_{k|\theta}^{-1} \Phi_{1:k}^\top y_{1:k} = b_{k-1} + \frac{1}{\sigma_y^2} \Phi_k^\top y_k, \quad (21)$$

with initial conditions $\Lambda_{0|\theta} = \Lambda_{\bar{z}|\theta}$ and $b_0 = \mathbf{0}$. Note that the shape function vectors are extended by a zero vector of length p such that $\Phi_k \in \mathbb{R}^{n+p}$. The final sequential GMRF regression algorithm is summarized in Algorithm 1. In order to obtain the variance vector $\text{diag}(\Lambda_{k|\theta}^{-1})$ of the full latent field, without calculating the inverse of the precision matrix, the Sherman–Morrison formula is used, Line 13. The complete derivation is outlined in Appendix A. For the sake of clarity, the notation for the conditioning of the GMRF matrices on the hyperparameters θ is omitted. It is worth mentioning that adding the product $\Phi_k^\top \Phi_k$ to $\Lambda_{0|\theta}$ does not significantly increase the density of the initial precision matrix. Thus, the algorithm has a computational complexity of $O(1)$ over time.

Algorithm 1 Sequential GMRF Regression

Require: Hyperparameter vector θ , Extended field grid \mathcal{S} , Regression function vector m

Measurement variance σ_y^2 , $\mathbf{b}_{0,0} = \mathbf{0}$, $\Lambda_{0,0} \triangleq \Lambda_{\bar{z}}$

- 1: compute $\text{diag}(\Sigma_0) = \text{diag}(\Lambda_{\bar{z}}^{-1})$
 - 2: **for** $k \in \mathbb{Z}_{>0}$ **do**
 - 3: **for** $1 \leq j \leq N$ **do**
 - 4: get j -th agent location $x_{k,j}$ and measurement $y_{k,j}$
 - 5: compute $\Phi_{k,j}(x_{k,j}, \mathcal{S})$
 - 6: $\mathbf{b}_{k-1,j} = \mathbf{b}_{k-1,j-1} + \frac{1}{\sigma_y^2} \Phi_{k,j}^\top \mathbf{y}_k$
 - 7: $\Lambda_{k-1,j} = \Lambda_{k-1} + \sum_{l=1}^N \frac{1}{\sigma_y^2} \Phi_{k,l}^\top \Phi_{k,l}$
 - 8: $\mathbf{h}_{k,j} = \Lambda_{k-1,j}^{-1} \Phi_{k,j}^\top$
 - 9: **end for**
 - 10: $\mathbf{b}_{k,0} = \mathbf{b}_{k-1,N}$
 - 11: $\Lambda_{k,0} = \Lambda_{k-1,N}$
 - 12: $\boldsymbol{\mu}_k = \Lambda_{k,0}^{-1} \mathbf{b}_{k,0}$
 - 13: $\text{diag}(\Sigma_k) = \text{diag}(\Sigma_{k-1}) - \sum_{l=1}^N \frac{\mathbf{h}_{k,l} \circ \mathbf{h}_{k,l}}{\sigma_y^2 + \Phi_{k,l} \mathbf{h}_{k,l}}$
 - 14: **end for**
-

3.2.2. Hyperparameter Estimation for Sequential GMRF Regression

A possible straightforward extension of the proposed model, is described in Xu et al. [13]. In this work, hyperparameter estimation is incorporated by defining the maximum a posteriori distribution $p(\theta|\mathbf{y}) \sim p(\mathbf{y}|\theta)p(\theta)$ with $p(\theta)$ being a uniform prior distribution over a discrete set of hyperparameter combinations. Approximating the integral by a discrete sum decreases the computational load compared to a numerical integration over $p(\theta|\mathbf{y})$. Furthermore, such an approach scales linearly with the number of hyperparameter combinations and can be extended to incorporation of continuous measurements. However, the method requires that the set of hyperparameters are chosen a priori.

3.3. Kalman Regression for Field Estimation

The Kalman regression model by [15], also referred to as spacetime Kalman filtering (STKF), incorporates off-grid measurements by adding new grid vertices to the belief lattice. In the following, we propose the concept of weighted shape functions to fuse new observations from continuous space into the already existing neighboring vertices of the discrete grid, see Figure 6. This allows us to keep the number of vertices and their spatial density constant. In order to make this article self-sufficient, we briefly summarize the derivation of the STKF.

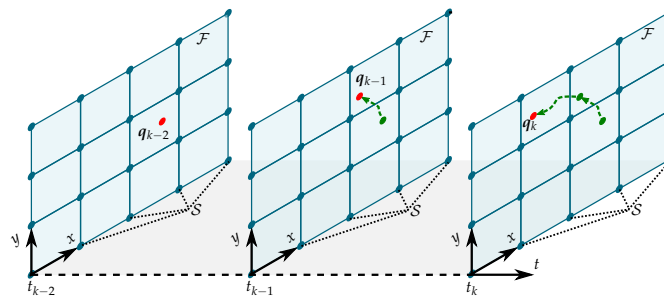


Figure 6. An agent takes measurements at the position q while maneuvering through a field \mathcal{F} . The environmental field is discretized into a regular lattice with the set of locations \mathcal{S} . The STKF random variables $\hat{f}(t)$ estimate the spatio-temporal process $f(t)$ on \mathcal{S} .

3.3.1. Process Model

Given the spatio-temporal physical process $f(x, t)$, its covariance function $K(\cdot)$ is assumed to be separable in time and space, as well as stationary in time $K(x, x', t, t') = K_s(x, x')K_t(t, t')$. Therefore, the power spectral density (PSD) $S_r(\omega) = W(i\omega)W(-i\omega)$ of the temporal covariance K_t can be approximated by a rational function of order $2r$. As rational functions are universal function approximators, arbitrary (non-stationary) temporal spectra can be approximated by increasing r .

The individual temporal process component $z_i(t)$ defined by K_t is represented by a continuous state space model S_i in companion form using the Wiener–Khinchin theorem and realization theory, such that

$$S_i : \begin{cases} \dot{s}_i(t) = F s_i(t) + G w_i(t), \\ z_i(t) = H s_i(t), \end{cases} \quad i \in \{1, \dots, n\}. \quad (22)$$

where $w(t) \sim \mathcal{N}(0, I)$ and the matrices F , G , and H are in companion form

$$F = \begin{bmatrix} 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ & & & \ddots & \\ 0 & 0 & 0 & \dots & 1 \\ -a_0 & -a_1 & -a_2 & \dots & -a_{r-1} \end{bmatrix}, \quad G = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix}, \quad H = [b_0 \quad b_1 \quad b_2 \quad \dots \quad b_{r-1}]. \quad (23)$$

The initial state yields $s(0) \sim \mathcal{N}(\mathbf{0}, \Sigma_0)$. The covariance matrix Σ_0 is obtained as the solution to the Lyapunov equation $F\Sigma_0 + \Sigma_0F^\top + GG^\top = \mathbf{0}$. Note that $s(t)$ does not provide a directly intuitive physical interpretation. The temporal process component is obtained as $z(t) = [z_1(t), \dots, z_n(t)]^\top \in \mathbb{R}^n$. The spatial covariance matrix \bar{K}_S is computed on S through the spatial covariance function K_s . Finally, the process on S is obtained by spatially correlating all $z_i(t)$ through the Cholesky factorization $\bar{K}_S^{1/2}$ of \bar{K}_S , reading

$$f(t) = \bar{K}_S^{1/2} z(t). \quad (24)$$

The spatio-temporal process model is depicted in Figure 7. With $s(t) = [s_1^\top(t), \dots, s_n^\top(t)]^\top(t) \in \mathbb{R}^{n \times r}$ and process noise $w(t) = [w_1(t), \dots, w_n(t)] \in \mathbb{R}^n$, Equation (22) is condensed to

$$S : \begin{cases} \dot{s}(t) = (I \otimes F)s(t) + (I \otimes G)w(t), \\ f(t) = \bar{K}_S^{1/2}(I \otimes H)s(t), \end{cases} \quad (25)$$

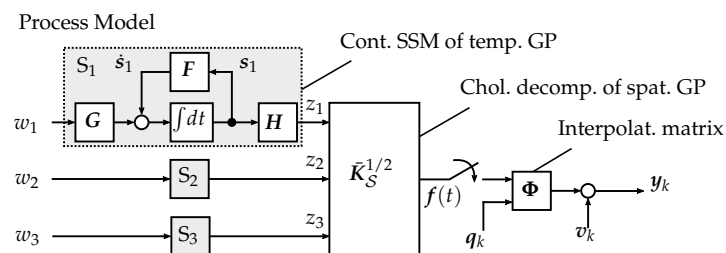


Figure 7. Block diagram of the process model on a field lattice with three vertices.

in which the Kronecker product is denoted by \otimes . The previous equations are discretized to enable numerical implementation. Thereby, the discrete time step t_k is abbreviated as k with a single time step being $T_k = t_k - t_{k-1}$. The discrete process model of (25) reads

$$\begin{cases} s_{k+1} = A s_k + w_k, \\ y_k = C_k s_k + v_k. \end{cases} \quad (26)$$

The discrete state transition matrix is obtained as

$$A = \exp(\mathbf{I} \otimes \mathbf{F})T_k \in \mathbb{R}^{rn \times rn}. \quad (27)$$

The zero mean Gaussian noise $\mathbf{w}_k \in \mathbb{R}^m$ is defined by the covariance matrix $\mathbf{Q}_k = \mathbf{I} \otimes \bar{\mathbf{Q}}_k$ with

$$\bar{\mathbf{Q}}_k = \int_0^{T_k} \left(\exp(\mathbf{F}\tau) \mathbf{G} \mathbf{G}^\top (\exp(\mathbf{F}\tau))^\top \right) d\tau. \quad (28)$$

The measurement noise vector $\mathbf{v}_k = [v_{k,1}, \dots, v_{k,N}]^\top$ is defined as $v_k \sim \mathcal{N}(\mathbf{0}, \Sigma_y)$ with $\Sigma_y = \sigma_y^2 \mathbf{I} \in \mathbb{R}^{N \times N}$.

The discrete observation matrix is obtained as

$$\mathbf{C}_{S,k} = \bar{\mathbf{K}}_S^{1/2} (\mathbf{I} \otimes \mathbf{H}) \in \mathbb{R}^{N \times rn}. \quad (29)$$

In contrast to [15], where new vertices are initialized to include off-grid information, we map the observation y_k collected at timestep k at position q_k to neighboring belief vertices through (11). Using the measurement interpolation matrix Φ_k , we obtain a transformation from the discrete belief lattice to a continuous field measurement as

$$\mathbf{C}_k = \Phi_k \bar{\mathbf{K}}_S^{1/2} (\mathbf{I} \otimes \mathbf{H}) = \Phi_k \mathbf{C}_{S,k} \in \mathbb{R}^{N \times rn}, \text{ with } \Phi = (\Phi_1^\top, \dots, \Phi_N^\top)^\top. \quad (30)$$

3.3.2. Kalman Regression

As the temporal process model in (26) is known and linear, a Kalman filter scheme can be used to estimate the evolution of the temporal process component \mathbf{s}_k by incorporating observation \mathbf{y}_{k+1} [26]. Furthermore, if the noise is assumed to be zero mean and Gaussian distributed, the Kalman filter estimates are optimal in a mean-square sense. The STKF belief model is summarized in Algorithm 2.

Given the state space model in (26), the temporal state component at time step $k + 1$ evolves in time according to $\mathbf{s}_{k+1} \sim \mathcal{N}(\mathbf{A}_k \mathbf{s}_k, \mathbf{Q}_k)$, where \mathbf{Q}_k is the corresponding process noise matrix. The measurement obtained from the real process is assumed to result from an affine transformation of the temporal state component, reading $\mathbf{y}_{k+1} \sim \mathcal{N}(\mathbf{C}_{k+1} \mathbf{s}_{k+1}, \Sigma_y)$.

In a first step, the Kalman filter *predicts* the temporal process component at the next time step $\hat{\mathbf{s}}_{k+1|k}$ based on the previous estimated temporal state component $\hat{\mathbf{s}}_{k|k}$, such that $\mathbf{s}_{k+1|k} \sim \mathcal{N}(\hat{\mathbf{s}}_{k+1|k}, \Sigma_{k+1|k})$. The state mean prediction $\hat{\mathbf{s}}_{k+1|k}$ and predicted state covariance matrix $\Sigma_{k+1|k}$ are depicted in lines 9 and 10 of Algorithm 2 respectively.

In the second step, the Kalman filter *updates* the temporal state component $\hat{\mathbf{s}}_{k+1|k+1}$ by conditioning the RV on \mathbf{y}_{k+1} , such that $\mathbf{s}_{k+1|k+1} \sim \mathcal{N}(\hat{\mathbf{s}}_{k+1|k+1}, \Sigma_{k+1|k+1})$. The equations for the updated state and covariance matrix are stated in lines 11 to 13 of Algorithm 2.

Figure 8 illustrates the STKF model. The process model consists of the state space models of the temporal process components $z_i(t)$, which are correlated using the product of the Cholesky decomposition of the spatial covariance matrix $\mathbf{C}_S = \bar{\mathbf{K}}_S^{1/2} (\mathbf{I} \otimes \mathbf{H})$. The Kalman filter predicts the next temporal process component, which is then updated using new measurements \mathbf{y}_k .

Algorithm 2 Kalman regression

Require: (F, G, H) state-space model of $S_r(\omega)$, measurement noise variance σ_y^2 , input location set \mathcal{S} , spatial, time kernels $K_s(\cdot, \cdot)$, and $h(\cdot)$

- 1: $\hat{s}(0|0) = 0$ and $\Sigma(0|0) = I \otimes \Sigma_0$.
- 2: Compute Σ_0 as solution of $F\Sigma_0 + \Sigma_0F + GG^\top = 0$
- 3: Compute $A_k, Q_k, \Sigma_{y,k}$ and $C_S = \bar{K}_S^{1/2}(I \otimes H)$
- 4: **for** $t \in \mathbb{R}_{>0}$ **do**
- 5: **if** $t \in]t_k, t_{k+1}[$ **then** {open-loop prediction}
- 6: $\hat{s}(t) = (\exp(I \otimes F)\tau)\Sigma_{k|k}(\exp(I \otimes F)\tau)^\top$
- 7: **else if** $t = t_{k+1}$ **then** {Kalman estimation}
- 8: Compute $\Phi(q_{k+1})$ and $C_{k+1} = \Phi(q_{k+1})C_S$
- 9: - Prediction step:
- 10: $\hat{s}_{k+1|k} = A_k\hat{s}_{k|k}$
- 11: $\Sigma_{k+1|k} = A_k\Sigma_{k|k}A_k^\top + Q_k$
- 12: - Update step:
- 13: $L_{k+1} = \Sigma_{k+1|k}C_{k+1}^\top(C_{k+1}\Sigma_{k+1|k}C_{k+1}^\top + \Sigma_{y,k+1})^{-1}$
- 14: $\hat{s}_{k+1|k+1} = \hat{s}_{k+1|k} + L_{k+1}(y_{k+1} - C_{k+1}\hat{s}_{k+1|k})$
- 15: $\Sigma_{k+1|k+1} = (I - L_{k+1}C_{k+1})\Sigma_{k+1|k}$
- 16: **end if**
- 17: - Process estimate:
- 18: $\hat{f}(t) = C_S\hat{s}(t)$
- 19: $\Sigma^f(t) = C_S\Sigma^s(t)C_S^\top$
- 20: **end for**

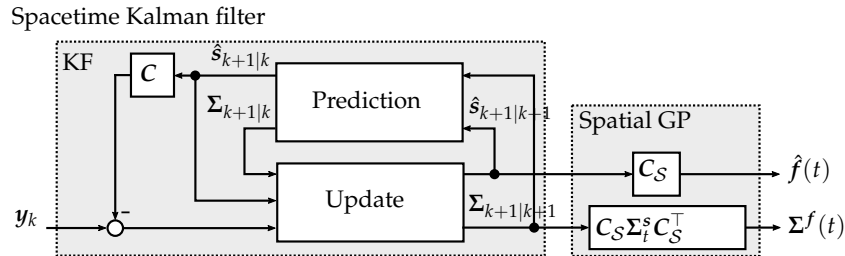


Figure 8. Block diagram of the spacetime Kalman filter.

The computational complexity of Algorithm 2 is dominated by the inverse computation of the Kalman gain in line 12. The computational complexity of the STKF algorithm is bounded by

$$O(r \cdot n \cdot N + N^3 + n \cdot P), \quad (31)$$

in which r is the order of a single state space model in (23), n is the number of vertices of \mathcal{S} , while P is the number of open-loop predictions performed, and N is the number of agents collecting measurements at each discrete time step [15]. In this work, we do not perform any open-loop predictions, thus P is zero. Note that all variables in (31) are assumed to be constant over time. If spatial hyperparameter estimation is performed, the Cholesky transformation of the new spatial covariance matrix needs to be computed. In this case, the computational load is dominated by the computation of the spatial Cholesky decomposition being $O(n^3)$.

3.3.3. Hyperparameter Estimation in Kalman Regression

In the STKF, the spatial Kernel hyperparameters can be estimated using a standard estimation method for GPs, such as maximum a posteriori estimation. However, GP hyperparameter estimation methods have the disadvantage of suffering from the big- n problem. In this sense, finding an optimization method that enables spatial hyperparameter estimation represents a challenge yet to be solved. As a state space model approximates the temporal process component, the temporal process hyperparameters can be estimated using methods developed for parameter estimation in finite state space models, as pointed out in [27]. Such methods inherit the advantage of having linear time computational complexity.

4. Path Integral Control for Exploratory Path Planning

The final path planning algorithm is summarized in Algorithm 3. In a discrete receding horizon formulation, the optimal control vector is recomputed at each sampling instance, while only the first control input is applied to the path planning model to generate the next way-point.

In the first initialization of the algorithm the initial control sequence $\mathbf{u}_{0:(H-1)}(k=0)$ is assumed to be the null vector. Afterwards, at each subsequent planning step, the initial control sequence is set to $\mathbf{u}_{0:(H-1)}(k) = [\mathbf{u}_{1:(H-2)}(k-1)^\top, 0]^\top$. With $\mathbf{u}_{0:(H-1)}(k)$ and the sampled exploration noise $\varepsilon_{0:(H-1),\ell}$ the ℓ -th path roll-out is computed in line 6 of Algorithm 3.

Exploration of underwater environmental fields is often conducted by underwater robots whose propulsion system produces mainly forward-directed thrust, which allows higher cruising speeds and, thus, faster exploration. These robots typically come with non-holonomic dynamics, e.g., the HippoCampus micro underwater robot [3]. Hence, analog to [23], we use an unicycle model to make use of the dynamic constraints and efficiently generate path roll-outs. The model reads

$$\mathbf{x}_{k+1} = \begin{bmatrix} x_{k+1} \\ y_{k+1} \\ \alpha_{k+1} \end{bmatrix} = \begin{bmatrix} v \cos(\alpha_k) \\ v \sin(\alpha_k) \\ f^{(c)} \end{bmatrix} \Delta t + \begin{bmatrix} 0 \\ 0 \\ g^{(c)\top} \end{bmatrix} (\mathbf{u}_k \Delta t + \varepsilon_k \sqrt{\Delta t}), \quad (32)$$

with the directly controllable system dynamics $f^{c}_{\text{robot}} = 0$, as well as the control transition matrix $g^{(c)\top} = 1$ being deterministic. Since $g^{(c)\top}$ is scalar, the weighted control projection matrix also reduces to a scalar $M_{t_j} = 1$.

The cost for the ℓ -th path segment $\tau_{i:H,\ell}$ is computed in line 7 of Algorithm 3. Note that if we average over $M_{t_i} \mathbf{u}_{0:(H-1)}$ the algorithm could become unstable. As [21] points out, the matrix M_{t_i} is a projection of $\mathbf{u}_{0:(H-1)}$ onto the column space of \mathbf{g}_{t_j} weighted by the metric \mathbf{R}^{-1} . A multiplication with M_{t_i} results in a decreasing $\mathbf{u}_{0:(H-1)}$. The state cost $q_{i,\ell}$ is set to be the predictive variance of the belief model at the associated state.

Afterwards, the probability of each path segment $P(\tau_{i:H,\ell})$ is obtained by normalizing each probability measure of $\tilde{S}(\tau_{i:H,\ell})$ through the sum of path segment probabilities of all roll-outs in line 10. In this line, λ is a sensitivity parameter that is eliminated by subtracting a constant term from $\tilde{S}(\tau_{i:H,\ell})$, writing

$$\exp\left(-\frac{1}{\lambda} \tilde{S}(\tau_{i:H,\ell})\right) = \exp\left(-c \frac{\tilde{S}(\tau_{i:H,\ell}) - \min \tilde{S}(\tau_{i:H,\ell})}{\max \tilde{S}(\tau_{i:H,\ell}) - \min \tilde{S}(\tau_{i:H,\ell})}\right), \quad (33)$$

with $c = 10$, as proposed by [21]. Figure 9 illustrates the computation of the path segment dependent control correction through line 13 and subsequently the computation of the averaged control vector through line 16. For all path segments of equal length, the exploration noise at each step of the path segment of equal length are weighted by $P(\tau_{i:H,\ell})$ and summed up. This results in the weighted path segment dependent control correction, line 13 of Algorithm 3. In this equation, the lower index defines the relative position in the path segment, while the higher index defines to which path segment the

vector entry belongs. Per definition, the index number of the last sample path location is equal to $H = t_H / \Delta t$. It follows that $\Delta \mathbf{u}_{0:(H-1)}$ is of size $H \times 1$ and $\Delta \mathbf{u}_{i:(H-1)}$ is of size $(H - i) \times 1$. Hence, line 13 computes an optimal control correction vector $\Delta \mathbf{u}_{i:(H-1)}$ for each path segment over all K roll-outs.

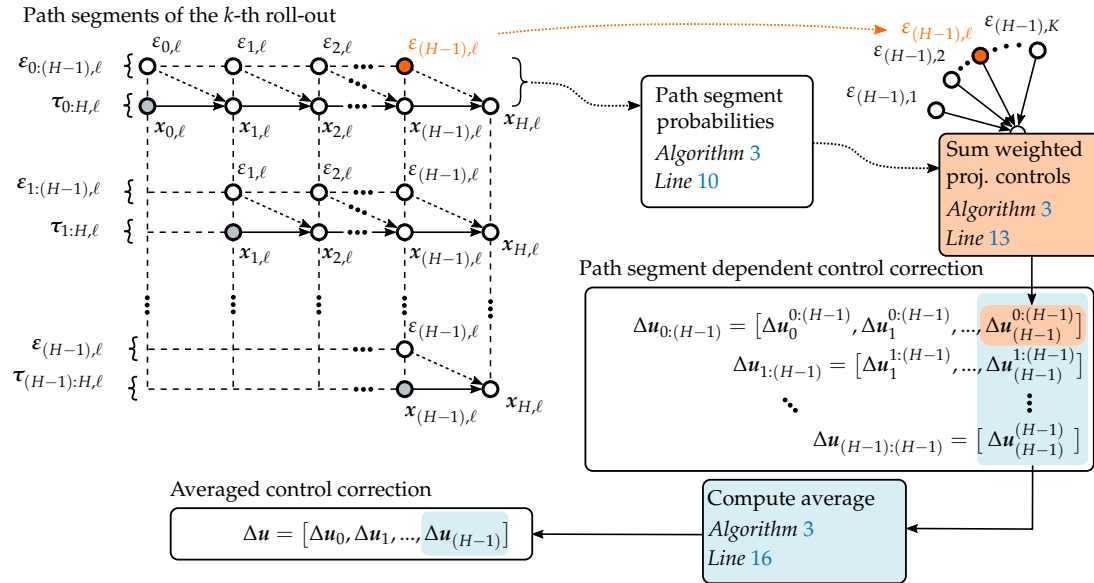


Figure 9. Schematic visualization of the computational steps involved in computing the averaged control correction from K sampled path roll-outs of length H using the PI^2 .

Algorithm 3 PI^2 for path planning

Require: Cost function $r_k = q_k + \mathbf{u}^\top \mathbf{R} \mathbf{u}$, unicycle exploration policy $\mathbf{x}_{k+1}(u_k, \mathbf{x}_k, \Delta t)$, exploration noise variance Σ_ϵ , sampling time Δt , initial optimal control sequence $\mathbf{u}_{0:(H-1)}(k=0)$, number of sampled paths K , control horizon steps H , control computation iterations n_{updated}

- 1: $\mathbf{u}_{0:(H-1)}^{(\text{start})}(k) = [\mathbf{u}_{1:(H-2)}(k-1)^\top, 0]^\top$
- 2: $\mathbf{M} = \frac{\mathbf{R}^{-1} \mathbf{g} \mathbf{g}^\top}{\mathbf{g}^\top \mathbf{R}^{-1} \mathbf{g}}$ (Weighted control projection matrix)
- 3: **for** $1 \leq (\dots) \leq n_{\text{updated}}$ **do**
- 4: **for** $1 \leq \ell \leq K$ **do**
- 5: - Sample exploration noise: $\epsilon_{0:(H-1),\ell} \sim \mathcal{N}(0, \Sigma_\epsilon)$
- 6: - Compute path roll-outs: $\tau_{0:H,\ell}(\mathbf{x}_i, \mathbf{u}_{0:(H-1)}, \epsilon_{0:(H-1),\ell}, \Delta t)$
- 7: - Compute cost of paths segments:

$$\tilde{S}(\tau_{i:H,\ell}) = \sum_{h=i}^H q_{h,\ell} + \frac{1}{2} \sum_{h=i-1}^{H-1} \frac{1}{2} (\mathbf{u}_h + \mathbf{M}_{h,\ell} \epsilon_{h,\ell})^\top \mathbf{R} (\mathbf{u}_h + \mathbf{M}_{h,\ell} \epsilon_{h,\ell})$$
- 8: **end for**
- 9: **for** $1 \leq \ell \leq K$ **do**
- 10: - Path segment probabilities: $P(\tau_{i:H,\ell}) = \frac{\exp(-\frac{1}{\lambda} \tilde{S}(\tau_{i:H,\ell}))}{\sum_{k=1}^K \exp(-\frac{1}{\lambda} \tilde{S}(\tau_{i:H,\ell}))}$
- 11: **end for**
- 12: **for** $0 \leq i \leq (H-1)$ **do**
- 13: - Path segment dependent control correction: $\Delta \mathbf{u}_{i:(H-1)} = \sum_{\ell=1}^K P(\tau_{i:H,\ell}) \mathbf{M} \epsilon_{i:(H-1),\ell}$
- 14: **end for**
- 15: **for** $0 \leq i \leq (H-1)$ **do**
- 16: - Averaged control correction: $[\Delta \mathbf{u}]_i = \frac{\sum_{h=0}^{H-1} ((H-h)[\Delta \mathbf{u}_{h:(H-1)}]_i)}{\sum_{h=0}^{H-1} (H-h)}$
- 17: **end for**
- 18: $\mathbf{u}_{0:(H-1)}^{(\text{new})} = \mathbf{u}_{0:(H-1)} + \Delta \mathbf{u}$
- 19: **end for**

Subsequently, the path segment dependent control vectors $\Delta \mathbf{u}_{i:(H-1)}$ are weighted by the trajectory length and averaged in line 16. In this manner, for the time step $(H - 1)$, K potential control corrections are computed and averaged to obtain a single control correction. The obtained control correction $\Delta \mathbf{u}$ is added to the previous control sequence resulting in the new control vector, line 18. The algorithm is repeated until the number of iterations reaches n_{updates} .

5. Field Belief Comparison

In this Section, the sequential GMRF regression model stated in Algorithm 1 and the STKF regression model stated in Algorithm 2 are compared for their computational performance time and prediction capabilities. Throughout the analysis, we use the empirical GMRF algorithm proposed by [23] as performance baseline.

5.1. Computational Complexity

In the following we analyze the computational complexity of our proposed field belief algorithms. The upper bounds of the belief algorithm's computational complexity are summarized in Table 1. Thereby, the original empirical GMRF algorithm as proposed in [23] already shows a drastic improvement with regard to processor and memory requirements compared to vanilla GP regression. Nonetheless, the empirical GMRF algorithm still suffers from a linearly increasing computational cost over the number of time steps k . The sequential GMRF regression algorithm utilizes the canonical form of the GP, which in combination with a predefined GMRF precision matrix enables a sequential update of the belief. Therefore, the sequential Bayesian GMRF algorithm has a constant computational time with upper bound $O(Nn^{3/2})$ for the two dimensional scenario. In general, the computational time of the GMRF increases with the number of dimensions as the bandwidth of the precision matrix increases [25]. The STKF's upper bound on the computational complexity is $O(rnN + N^3)$. For the case of spatial hyperparameter estimation, the STKF's computational complexity is limited by the computation burden of the spatial Cholesky factorization being $O(n^3)$ for dense matrices.

Table 1. Computational complexity of the developed belief algorithms for two and three dimensions of the field belief with number of agents N , number of discrete time steps k , number of field grid values n , and dimension of the state space model r .

Belief Algorithm	2d	3d
GP Regression	$O((Nk)^3)$	$O((Nk)^3)$
Empirical GMRF Regression	$O(n^{3/2}) + O(Nk)$	$O(n^2) + O(Nk)$
Bayesian GMRF Regression	$O(Nn^{3/2})$	$O(Nn^2)$
STKF	$O(rnN + N^3)$	$O(rnN + N^3)$

5.2. Environmental Field Estimation

The hyperparameter configurations of the individual algorithms and the corresponding acronyms are listed in Table 2. The hyperparameters of the belief model in Table 2 are tuned by hand in order to optimize the approximation result. Hereby, the size of the field lattices are tailored such that the computation time has the same value for all algorithms. Note that GMRF-2 and GMRF-3 use the same regression algorithm, but the used CAR process type, boundary condition, and lattice size differ. During the simulation, measurements are collected from the spatial field depicted in Figure 10. On each belief update step, the next measurement location is chosen to be the point with the highest predictive variance plus a Gaussian noise term with a variance of 0.5 m^2 .

In order to provide meaningful results the simulation setup is run over 50 individual cycles for each belief algorithm. Figure 11a shows the root mean square (RMS) of the predictive variance sum. We choose the predictive variance sum as belief convergence criteria as it—unlike the empirical

mean—is more sensitive to outliers. The convergence behavior of the predictive variance sum’s RMS is utilized as a measure for the exploration performance.

Table 2. Hyperparameter configurations of the individual algorithms and corresponding acronyms. The size of the belief field grid S is defined by the total number of vertices in x -direction being n_x . The second value inside the brackets depicts the total number of padding vertices in one dimension.

Acronym	Belief Algorithm	Process Type			Boundary Cond.
GMRF-1	Empirical GMRF	Matérn CAR(1)			Neumann
GMRF-2	Bayesian GMRF	Matérn CAR(1)			Neumann
GMRF-3	Bayesian GMRF	Matérn CAR(2)			Torus
STKF-1	STKF	Spat.: Matérn Cov. ($\nu = 1$)			-
		Temp.: Exp. Cov.			
Acronym	$n_x \times n_y$	κ^2	τ	σ_f	l
GMRF-1	$(80 + 10) \times (40 + 10)$	10^{-5}	1	-	-
GMRF-2	$(80 + 10) \times (40 + 10)$	10^{-4}	0.5	-	-
GMRF-3	$(40 + 20) \times (20 + 20)$	0.01	1	-	-
STKF-1	$(40 + 0) \times (20 + 0)$	-	-	Spat.: 1.8 Temp.: 1	Spat.: 3.2 Temp.: 10^7

Figure A1 in Appendix B illustrates the mean and variance prediction results for the different regression models. Due to the steep correlation structure between known and unknown field values induced by a CAR(1) model, comparatively many measurements are taken in the vicinity of the field boundary. If the padding around the true field GMRF grid is chosen relatively small, distortion effects of the boundary conditions affect the estimation result. The predictive variance of GMRF-1 and GMRF-2 after one measurement shows a non-circular shape, which is induced by the Neumann boundary condition, while the predictive variance of the GMRF-3 after one measurement increases on the edges of the field due to the Torus boundary condition. While designing a GMRF, the dependencies between the field approximation, the used GMRF model, and the chosen boundary condition must be taken into account.

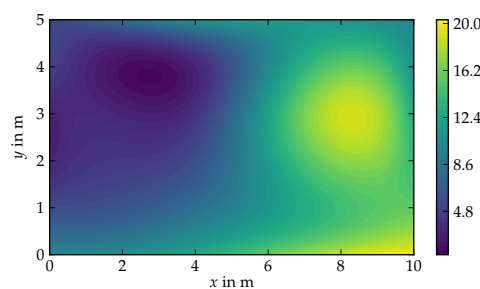


Figure 10. The concentration field used for generating measurements.

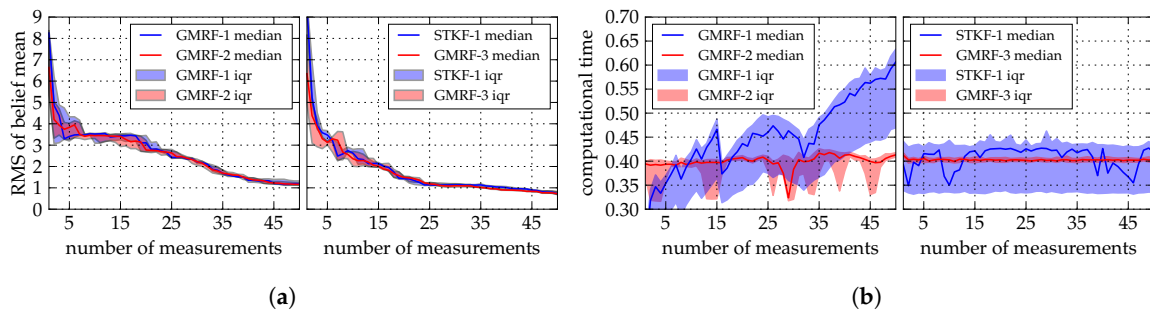


Figure 11. Belief model comparison. (a) Median and inter-quarter ranges of the root mean square of the belief algorithms' predictive variance sum. (b) Medians and inter-quarter ranges of the belief algorithms' computational time over fifty simulation runs.

Figure 11b illustrates the median and inter-quarter range (IQR) of the computational time of the different belief models after fifty simulation runs. As expected, the computational time of GMRF-1 increases practically linearly over time due to the increasing number of observations. GMRF-3 has approximately 4500 lattice vertices, while GMRF-2 has approximately 2400 lattice vertices. While GMRF-2 and GMRF-3 both utilize the same regression algorithm with different lattice sizes the computational time almost equals. The same computation time can be attributed to the CAR(2) model used for GMRF-3, which results in a less sparse precision matrix than compared to the CAR(1) model of GMRF-2.

6. Analysis of the Exploration Algorithm

In this Section, the STKF belief model is combined with the stochastic controller and the performance of the resulting exploration algorithm, abbreviated as PI-STKF, is analyzed. First, the PI-STKF algorithm is simulated for the scenario of a spatially stationary field as introduced in Section 5.2. Afterwards, the PI-STKF algorithm is simulated in a spatio-temporal exploration scenario, demonstrating the suitability of the developed exploration algorithm for long-term field monitoring tasks. The PI-STKF simulation parameters are listed in Table 3. In order to analyze the effect of different control horizons we consider three robot agents which we abbreviate as 'Agent-4', 'Agent-9', and 'Agent-14', corresponding to their control horizons of $t_H = 4$ s, $t_H = 9$ s, and $t_H = 14$ s respectively.

In order to analyze the performance of the algorithms completely isolated from external non-reproducible influences, simulations are conducted on a 2.4 GHz Dualcore-CPU 'i5-2430M' and 8 GB RAM computer where the algorithms are implemented in Python. The computer executes the exploration algorithm at a frequency of approximately 1.5–3 Hz and, thus, sufficiently fast to provide underlying low-level control schemes with the required data. Moreover, it is worth mentioning that the current Python-implementation is not yet speed-optimized such that a sufficiently fast execution time is expected on the micro robot's onboard hardware.

Table 3. PI-STKF simulation parameters.

PI-Control Parameters	Symbol	Value
Control Horizon	t_H	4 s 9 s 14 s
Agent velocity	v	0.5 $\frac{m}{s}$
Simulation time	-	150 s
Time step	Δt	1 s
Trajectory roll-outs	K	15
Control loop updates	$n_{updates}$	10
Measurement variance	σ_y^2	0.3
Exploration noise	ε	$\pi/10$
Control cost	R	1

6.1. Analytical Field Exploration

At every discrete time step, the agent receives a new measurement at its current location from the underlying simulated real field which is depicted in Figure 10. The measurement is perturbed by Gaussian measurement noise with variance σ_y^2 . The measurement is fed to the STKF belief model. The temporal length scale of the STKF belief model is set to 10^7 , such that the conditional variance of the belief model does not noticeably increase over time. The PI² algorithm utilizes the belief's conditional (predicted) variance as state cost. Figure 12 illustrates the roll-out sampling step for a control horizon length of $t_H = 4$ s (left) as well as $t_H = 14$ s (right). At $t = 40$ s the Agent-4 plans a optimal trajectory towards a local variance maximum. In contrast, Agent-14 takes a path towards the global variance maximum.

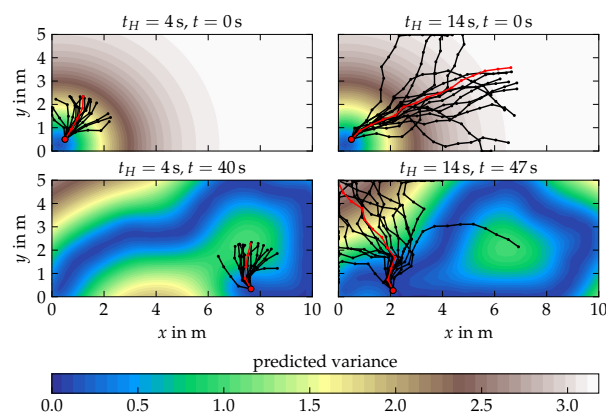


Figure 12. Illustration of the PI-STKF algorithm using different time horizon lengths t_H . The optimal path (1) starting at the current agent position (●) is computed by an iterative sampling of several potential paths (l) which are then weighted according to their probabilities which depend on the predicted variance.

Figure 13a,b illustrates the exploration result of the PI-STKF algorithm for a short a control horizon of 4 s and a longer horizon of 14 s. As depicted in Figure 12, Agent-4 samples trajectories which are not long enough to reach to the global maximum of the predicted variance field. As a result, Agent-4 follows a rather sub-optimal trajectory from $t_H = 9$ s until $t = 30$ s. Furthermore, between $t = 60$ s and 90 s Agent-4 moves again into the upper left corner, as the rather short sample roll-outs do not provide sufficient information regarding the location of the unknown field values. In contrast, Agent-14 shows a more exploratory behavior. On an intuitive level, one might wonder why Agent-14 first navigates to the upper field boundary, instead of directly maneuvering to the upper right corner of the field. However, even though the prediction horizon is longer than Agent-4's horizon, it is still possible that the relatively small number of sampled control roll-outs pushes the optimal path towards a temporary less informative path. When Agent-14 almost reaches the upper field boundary, symmetry breaking occurs. Symmetry breaking is a common phenomenon in stochastic optimal control that describes the sudden fixation of the controller towards one sample direction [22]. As the controller's state cost is the predicted field variance, the agents prioritize a path along the boundary of the field. After $t = 90$ s both agents obtained a good belief of the original field process. The effective computational time at each time instance sums up to 0.2 s when using a control horizon of $t_H = 4$ s and 0.3 s and 0.4 s for the control horizons of $t_H = 9$ s and $t_H = 14$ s respectively. Hereby, it is worth mentioning that the controller is implemented in Python using mainly a non-optimized for-loop structure.

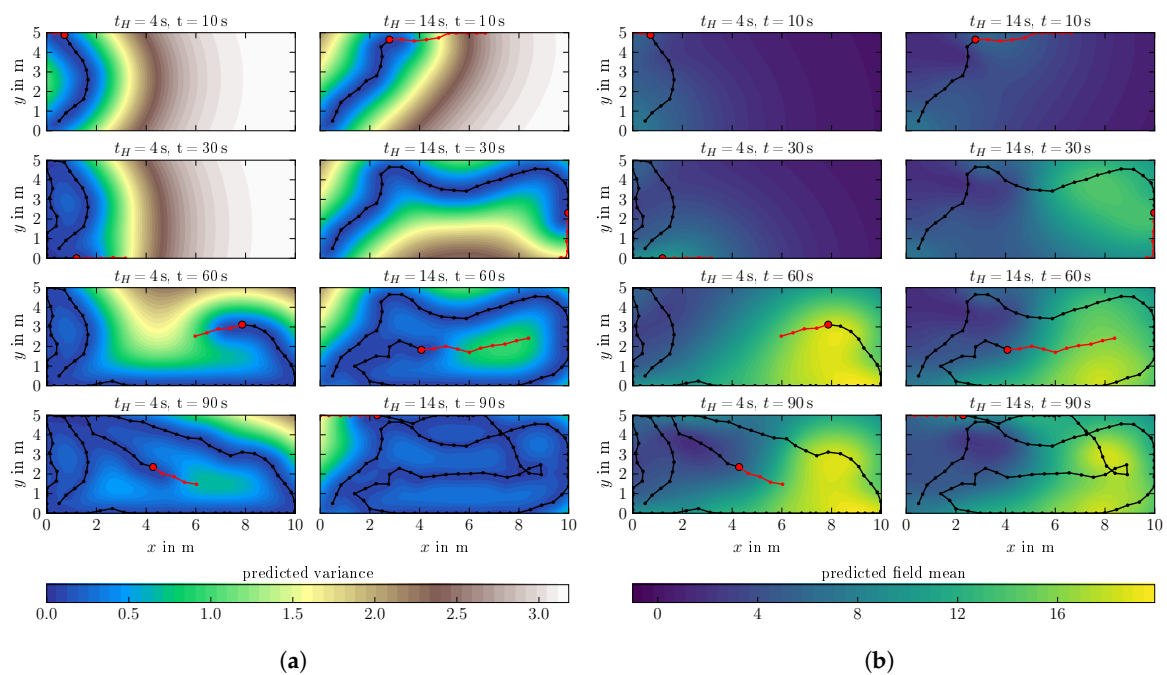


Figure 13. STKF algorithm exploration results for two time horizon lengths $t_H = 4$ s and $t_H = 14$ s and temporal length scale of 10^7 which results in a belief remaining almost constant with respect to time. The agent (●) moves along its trajectory (l) following an optimal path (l) by applying the first optimal control step. The optimal path is computed by PI² using the conditional variance of the STKF. (a) Predicted STKF field variance estimates. (b) Predicted STKF field mean estimates.

In order to measure the expected average exploration performance of the PI-STKF algorithm, the simulation with the stochastic field in Figure 10 is repeated 50 times for three different control horizons ($t_H = 4$ s, 9 s, 14 s) where each simulation episode has a length of 150 s. The agent initial position is picked uniformly random within the range of $x = 0.5$ m to $x = 9.5$ m for the x -coordinate while the y -coordinate is set to $y = 0.5$ m, and the robot's initial orientation is $\alpha = \pi/2$. The obtained results are compared to a random walk exploration strategy as a baseline. Figure 14a illustrates the median and inter-quarter range of the sum of the agent's conditional variance using STKF-1 as a belief model. Hereby, the stochastic controller consistently outperforms the random walk baseline strategy. In the first 15 s of each simulation the controllers with horizon $t_H = 4$ s and $t_H = 9$ s drive the agent towards the upper field boundary which results in a similar exploration performance. At approximately $t = 15$ s the exploration performance of Agent-4 decreases in comparison to the agents with longer control horizons. The information gain per time step in an unexplored field is almost independent from the control horizon when pursuing the the first time steps. However, agents with longer control horizons begin to profit from their far-sight controller as the field exploration mission continues. As a result, when compared to agents with longer horizon Agent-4 tends to maneuver itself on a less informative trajectory. In Figure 14a, the difference in exploration performance between the controller with $t_H = 9$ s and $t_H = 14$ s is small. Nonetheless, exploration missions in larger fields are likely to profit from longer control horizons.

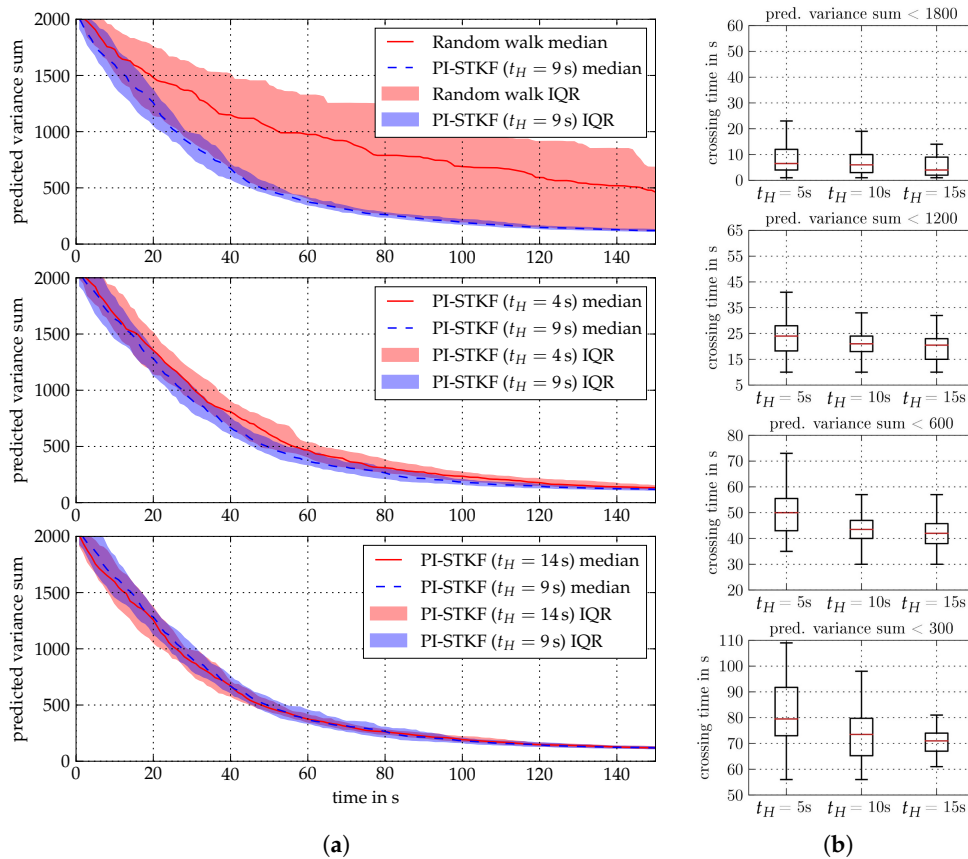


Figure 14. (a) Median and IQR of the conditional variance sum of the PI-STKF and a random walk exploration strategy after 50 simulation runs. (b) Box plot of the crossing times of a particular conditional variance sum for different control horizons t_H over 50 simulation runs.

In order to analyze the agent’s exploration efficiency we compare the time spans the agents require reach predefined exploration levels which we represent by the summed predictive variance. Hereby, a crossing time is defined as the time it takes for the agent to drive its field belief predictive variance sum beneath a predefined value. The corresponding crossing times are illustrated in Figure 14b. It can be seen that Agent-9 ($t_H = 9$ s) and Agent-14 ($t_H = 14$ s) outperform the controller of Agent-4 with a horizon of $t_H = 4$ s. Thereby, Agent-9 and Agent-4 show a similar exploration performance down to a predictive variance sum of 600. For lower predictive variances, the controller with $t_H = 14$ s provides a better and more predictable exploration performance. Note that although the difference of the crossing times lies within a range of 20 to 30 s and, thus, comparatively small the effect on the exploration performance will scale with the size of the field and the number of agents.

6.2. Spatio-temporal Field Exploration

In this section we examine the scenario of a spatio-temporal process. The field values are defined on the GMRF grid and interpolated between each other in order to obtain a continuous field. The temporal process at time instance $t = 10$ s, 30 s, 60 s, and 90 s is depicted in Figure 15(left column). We use the same parameters as in the PI-STKF simulation, see Table 3, except for the temporal length scale which we set to 15^5 . The reduction of temporal length scale let the agent’s belief variance increase over time if no measurement is obtained. Thus, the STKF is able to capture the evolution of the temporal process through the increase in uncertainty of the temporal process component. This is beneficial for coping with unknown stochastic components of the environmental process and hence potentially enables long-term autonomous monitoring scenarios.

Figure 15 illustrates the predicted field mean and variance of the STKF belief model, while the PI-STKF explores the spatio-temporal field. The spatio-temporal process at the agent's initial position starts with a concentration value of approximately -1.5 . After $t = 60$ s the agent has finished its initial clockwise exploration maneuver and passes the vicinity of its initial position again which concentration value has now changed to ca. 1.5 . As depicted in Figure 9, the optimal control is computed by sampling potential path roll-outs and weighting them through a path probability that results from the conditioned variance as well as the control effort. As the variance of the field belief has noticeably increased during this maneuver, the obtained measurements lead to a process estimate which fairly represents the underlying original concentration field. Thereby, the increasing conditional field variance describes the loss of information in field regions which have not been visited by the agent recently. The time instance $t = 90$ s is an illustrative example on the controller exploration strategy: the planned trajectory first heads towards a local variance maximum and then points to the agent's starting position at the beginning of the simulation. During the numerical experiment the computational time of the PI-STKF algorithm was on average 0.7 s and remained constant.

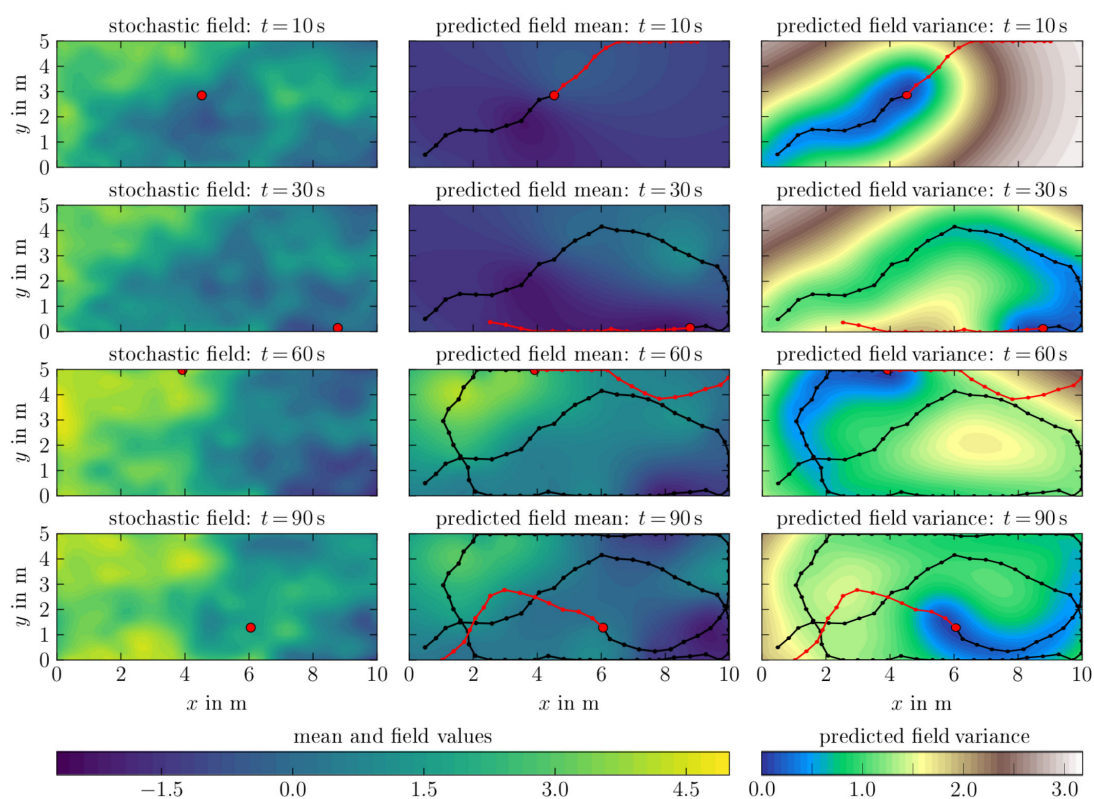


Figure 15. Agent-14 using the PI-STKF scheme with a control horizon of $t_H = 14$ s and temporal length scale of 15^5 resulting in a noticeable belief change over time, as represented by the increasing predicted field variance (left column). The agent collects measurements from an spatio-temporal concentration field to compute an optimal path (l) based on the predicted STKF field variance. The next position of the agent's trajectory (l) is obtained by applying the first value of the computed optimal control sequence. **Left:** Spatio-temporal concentration field **Center:** Predicted field mean **Right:** Predicted field variance.

7. Conclusions

In this work, we outline the combination of Kernel-based belief models with stochastic optimal control while ensuring an upper bound on the computational load of the algorithm. First, a sequential fully Bayesian GMRF algorithm was extended to incorporate off-grid observations through shape-functions. Second, we showed that a spacetime Kalman filter belief model combined with the PI^2 algorithm enables the exploration of spatio-temporal environmental fields. We demonstrated that

the STKF algorithm can be extended by using shape functions to handle off-grid observations while remaining constant computational complexity. Extensive simulations were carried out to systematically identify bottlenecks in the exploration algorithm's performance.

Future work will address the robustness of the PI-STKF algorithm while taking into account non-Gaussian measurement noise, uncertain localization, and uncertain hyperparameters. Furthermore, it is interesting to analyze the algorithms' performance in an underwater experimental setup by using micro underwater robots as mobile sensor nodes. However, the design of meaningful experimental studies is a challenging task itself, as it requires reproducible environmental fields to provide the same scenario setup to all algorithms. Such a benchmark field can be realized by using, for instance, look-up tables of the field onboard the robots or spatial depending illumination of the fluid volumes which is then measured by the robots. Moreover, future studies will examine the influence of different field geometries (e.g., irregular grid shapes) on the exploration performance. These will also include varying control horizons and exploration noise. Another aspect is the analysis of different information theoretic criteria to redefine the state cost of the optimal control problem. Such future studies could evolve around the approximation of common information theoretic criteria for path planning such as mutual information [24] to increase exploration performance. In this context, further analysis of the belief models' computational properties should be conducted. A combination of a GMRF which approximates the spatial process component while e.g., a Kalman Filter captures the temporal process component could result in a computationally efficient spatio-temporal GP model. Furthermore, the presented algorithms could be extended to a multi-agent fleet which uses a decentralized belief representation rather than sharing one central belief model in order to reduce the dependence on restrictive underwater communication systems.

Finally, the proposed approach for synthesizing an autonomous field exploration algorithm shares similarities to the problem of safely learning dynamic models, such as in [28,29]. In the aforementioned works, the authors leverage GP models to safely explore the state-space of an unknown dynamic system, while the state can only be changed continuously. We believe that the findings in one of those fields could be very fruitful for the other.

Author Contributions: D.A.D., E.K., and E.S. initiated and coordinated the project. D.A.D. and A.R.G. designed the numerical experiments. D.A.D. and A.R.G. performed the numerical experiments. D.A.D., A.R.G., E.S. analyzed the data. D.A.D., A.R.G., E.K., and E.S. wrote and revised the manuscript.

Funding: This research funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under grants Kr752/33-1 and Kr752/36-1. The publication of this article is funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) - Grant 392323616 and the Hamburg University of Technology (TUHH) in the funding programme *Open Access Publishing*. The support is greatly acknowledged.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

GP	Gaussian process
GMRF	Gaussian Markov random field
STKF	Spacetime Kalman filter
SSM	State Space Model
CAR	Conditional auto-regressive
PI ²	Policy improvement with path integrals
PI-GMRF	Combination of the GMRF belief model with the PI ² path planning algorithm
PI-STKF	Combination of the STKF belief model with the PI ² path planning algorithm
RMS	Root mean square
RL	Reinforcement learning
IQR	Inter-quarter range

Appendix A. Sequential Update Rule for Continuous GMRF Algorithm

First, (20) is rewritten into a sum of vector products with $\Phi_{k,j}$ denoting the interpolation matrix of the j -th agent at discrete time step k , such that

$$\begin{aligned} \Lambda_{k|\theta} &= \Lambda_{k-1|\theta} + \frac{1}{\sigma_y^2} \Phi_k^\top \Phi_k = \Lambda_{k-1|\theta} + \sum_{j=1}^N \frac{1}{\sigma_y^2} \Phi_{k,j}^\top \Phi_{k,j}, \\ &= \underbrace{\Lambda_{k-1|\theta} + \frac{1}{\sigma_y^2} \Phi_{k,1}^\top \Phi_{k,1}}_{\Lambda_{k-1,1|\theta}} + \frac{1}{\sigma_y^2} \Phi_{k,2}^\top \Phi_{k,2} + \dots + \frac{1}{\sigma_y^2} \Phi_{k,N}^\top \Phi_{k,N}. \end{aligned} \tag{A1}$$

$\underbrace{\hspace{15em}}_{\Lambda_{k-1,N|\theta} = \Lambda_{k|\theta}}$

By analyzes of the structure of (A1) an update rule is obtained that allows the sequential incorporation of new measurements as

$$\Sigma_{k-1,j|\theta} = \Lambda_{k-1,j|\theta}^{-1} = \left(\Lambda_{k-1,j-1|\theta} + \frac{1}{\sigma_y^2} \Phi_{k,j}^\top \Phi_{k,j} \right)^{-1}. \tag{A2}$$

Applying the Sherman-Morrison formula on (A2) results in

$$\Sigma_{k-1,j|\theta} = \Sigma_{k-1,j-1|\theta} - \frac{\Sigma_{k-1,j-1|\theta} \Phi_{k,j}^\top \Phi_{k,j} \Sigma_{k-1,j-1|\theta}}{\sigma_y^2 + \Phi_{k,j} \Sigma_{k-1,j-1|\theta} \Phi_{k,j}^\top}. \tag{A3}$$

With (A1) and (A3), the conditional covariance matrix is obtained as

$$\Sigma_{k|\theta} = \Sigma_{k-1|\theta} - \sum_{j=1}^N \frac{\Sigma_{k-1,j-1|\theta} \Phi_{k,j}^\top \Phi_{k,j} \Sigma_{k-1,j-1|\theta}}{\sigma_y^2 + \Phi_{k,j} \Sigma_{k-1,j-1|\theta} \Phi_{k,j}^\top} = \Sigma_{k-1|\theta} - \sum_{j=1}^N \frac{h_{k,j} h_{k,j}^\top}{\sigma_y^2 + \Phi_{k,j} h_{k,j}}. \tag{A4}$$

Thus, the sequential update rule for the conditional variance can be written as

$$\text{diag}(\Sigma_{k|\theta}) = \text{diag}(\Sigma_{k-1|\theta}) - \sum_{j=1}^N \frac{h_{k,j} \circ h_{k,j}}{\sigma_y^2 + \Phi_{k,j} h_{k,j}}, \text{ with} \tag{A5}$$

$$h_{k,j} = \Sigma_{k-1,j|\theta} \Phi_{k,j}^\top = \Lambda_{k-1,j|\theta}^{-1} \Phi_{k,j}^\top, \tag{A6}$$

$$\Lambda_{k-1,j|\theta} = \Lambda_{k-1|\theta} + \sum_{j=1}^N \frac{1}{\sigma_y^2} \Phi_{k,j}^\top \Phi_{k,j}. \tag{A7}$$

Appendix B. Estimation Examples of the Different Belief Models

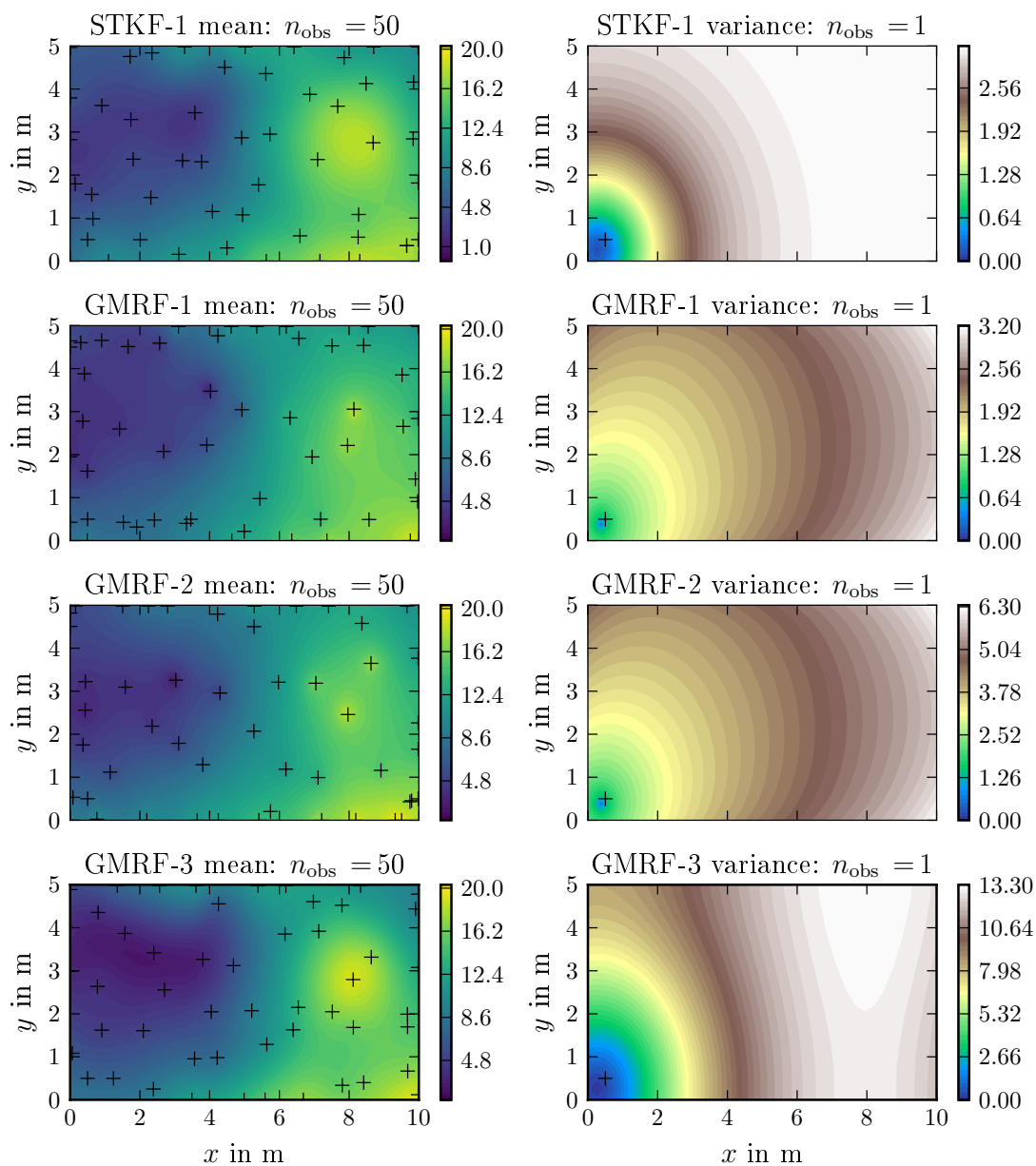


Figure A1. Estimation results of the Gaussian belief models. The parameters of the regression algorithms are denoted in Table 2. **Left:** Predicted field mean of different regression models after fifty observations. In each prediction step the next observation was chosen to be close to the maximum predicted variance. **Right:** Predicted field variance of different belief models after one observation.

References

- Gifford, C.M.; Webb, R.; Bley, J.; Leung, D.; Calnon, M.; Makarewicz, J.; Banz, B.; Agah, A. A novel low-cost, limited-resource approach to autonomous multi-robot exploration and mapping. *Robot. Autonomous Syst.* **2010**, *58*, 186–202. [[CrossRef](#)]
- Griffiths, A.; Dikarev, A.; Green, P.R.; Lennox, B.; Poteau, X.; Watson, S. AVEXIS—Aqua vehicle explorer for in-situ sensing. *IEEE Robot. Autom. Lett.* **2016**, *1*, 282–287. [[CrossRef](#)]

3. Duecker, D.A.; Hackbarth, A.; Johannink, T.; Kreuzer, E.; Solowjow, E. Micro Underwater Vehicle Hydrobatatics: A Submerged Furuta Pendulum. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), Brisbane, QLD, Australia, 21–25 May 2018; pp. 7498–7503.
4. Wang, H.; Liu, P.X.; Xie, X.; Liu, X.; Hayat, T.; Alsaadi, F.E. Adaptive fuzzy asymptotical tracking control of nonlinear systems with unmodeled dynamics and quantized actuator. *Inf. Sci.* **2018**, in press. [[CrossRef](#)]
5. Zhao, X.; Shi, P.; Zheng, X. Fuzzy Adaptive Control Design and Discretization for a Class of Nonlinear Uncertain Systems. *IEEE Trans. Cybern.* **2016**, *46*, 1476–1483. [[CrossRef](#)] [[PubMed](#)]
6. Hackbarth, A.; Kreuzer, E.; Gray, A. Multi-Agent Motion Control of Autonomous Vehicles in 3D Flow Fields. *PAMM* **2012**, *12*, 733–734. [[CrossRef](#)]
7. Krige, D.G. A Statistical Approach to some Basic Mine Valuation Problems on the Witwatersrand. *J. S. Afr. Inst. Mining Metall.* **1951**, *52*, 119–139.
8. Zhang, B.; Sukhatme, G.S. Adaptive Sampling for Estimating a Scalar Field using a Robotic Boat and a Sensor Network. In Proceedings of the 2007 IEEE International Conference on Robotics and Automation, Roma, Italy, 10–14 April 2007; pp. 3673–3680.
9. Xu, Y.; Choi, J.; Oh, S. Mobile Sensor Network Navigation using Gaussian Processes with Truncated Observations. *IEEE Trans. Robot.* **2011**, *27*, 1118–1131. [[CrossRef](#)]
10. Xu, Y.; Choi, J.; Dass, S.; Maiti, T. *Bayesian Prediction and Adaptive Sampling Algorithms for Mobile Sensor Networks: Online Environmental Field Reconstruction in Space and Time*; Springer: Heidelberg, Germany, 2015.
11. Cameletti, M.; Lindgren, F.; Simpson, D.; Rue, H. Spatio-temporal Modeling of Particulate Matter Concentration through the SPDE Approach. *AStA Adv. Stat. Anal.* **2013**, *97*, 109–131. [[CrossRef](#)]
12. Lindgren, F.; Rue, H.; Lindström, J. An Explicit Link between Gaussian Fields and Gaussian Markov Random Fields: the Stochastic Partial Differential Equation Approach. *J. R. Stat. Soc. Ser. B (Stat. Methodol.)* **2011**, *73*, 423–498. [[CrossRef](#)]
13. Xu, Y.; Choi, J.; Dass, S.; Maiti, T. Efficient Bayesian Spatial Prediction with Mobile Sensor Networks using Gaussian Markov Random Fields. *Automatica* **2013**, *49*, 3520–3530. [[CrossRef](#)]
14. Jadhaliha, M.; Jeong, J.; Xu, Y.; Choi, J.; Kim, J. Fully Bayesian Prediction Algorithms for Mobile Robotic Sensors under Uncertain Localization Using Gaussian Markov Random Fields. *Sensors* **2018**, *18*, 2866. [[CrossRef](#)] [[PubMed](#)]
15. Todescato, M.; Carron, A.; Carli, R.; Pillonetto, G.; Schenato, L. Efficient Spatio-Temporal Gaussian Regression via Kalman Filtering. *arXiv* **2017**, arXiv:1705.01485.
16. Hollinger, G.A.; Sukhatme, G.S. Sampling-based robotic information gathering algorithms. *Int. J. Robot. Res.* **2014**, *33*, 1271–1287. [[CrossRef](#)]
17. Marchant, R.; Ramos, F. Bayesian Optimisation for informative continuous path planning. In Proceedings of the 2014 IEEE International Conference on Robotics and Automation (ICRA), Hong Kong, China, 31 May–7 June 2014; pp. 6136–6143. [[CrossRef](#)]
18. Marchant, R.; Ramos, F.; Sanner, S. Sequential Bayesian Optimisation for Spatial-Temporal Monitoring. In Proceedings of the Conference on Uncertainty in Artificial Intelligence (UAI), Quebec, QC, Canada, 23–27 July 2014; pp. 553–562.
19. Cui, R.; Li, Y.; Yan, W. Mutual Information-Based Multi-AUV Path Planning for Scalar Field Sampling Using Multidimensional RRT-Star. *IEEE Trans. Syst. Man. Cybern. Syst.* **2016**, *46*, 993–1004. [[CrossRef](#)]
20. Xu, Y.; Choi, J. Spatial prediction with mobile sensor networks using Gaussian processes with built-in Gaussian Markov random fields. *Automatica* **2012**, *49*, 1735–1740. [[CrossRef](#)]
21. Theodorou, E.; Buchli, J.; Schaal, S. A Generalized Path Integral Control Approach to Reinforcement Learning. *J. Mach. Learn. Res.* **2010**, *11*, 3137–3181.
22. Kappen, H.J. An Introduction to Stochastic Control Theory, Path Integrals and Reinforcement Learning. *AIP Conf. Proc.* **2007**, *887*, 149–181.
23. Kreuzer, E.; Solowjow, E. Learning Environmental Fields with Micro Underwater Vehicles: A Path Integral—Gaussian Markov Random Field Approach. *Auton. Robots* **2018**, *42*, 761–780. [[CrossRef](#)]
24. Krause, A.; Singh, A.; Guestrin, C. Near-Optimal Sensor Placements in Gaussian Processes: Theory, Efficient Algorithms and Empirical Studies. *J. Mach. Learn. Res.* **2008**, *9*, 235–284.
25. Rue, H.; Held, L. *Gaussian Markov Random Fields: Theory and Applications*; CRC Press: Boca Raton, FL, USA, 2005.

26. Kalman, R.E. A New Approach to Linear Filtering and Prediction Problems. *J. Basic Eng.* **1960**, *82*, 35–45. [[CrossRef](#)]
27. Sarkka, S.; Solin, A.; Hartikainen, J. Spatiotemporal Learning via Infinite-dimensional Bayesian Filtering and Smoothing: A look at Gaussian Process Regression through Kalman Filtering. *IEEE Signal Process. Mag.* **2013**, *30*, 51–61. [[CrossRef](#)]
28. Turchetta, M.; Berkenkamp, F.; Krause, A. Safe exploration in finite markov decision processes with gaussian processes. In *Advances in Neural Information Processing Systems*; Curran Associates Inc.: Barcelona, Spain, 2016; pp. 4312–4320.
29. Zimmer, C.; Meister, M.; Nguyen-Tuong, D. Safe Active Learning for Time-Series Modeling with Gaussian Processes. In *Advances in Neural Information Processing Systems*; Curran Associates Inc.: Montreal, QC, Canada, 2018; pp. 2730–2739.



© 2019 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).