IEEE Robotics and Automation Letters (RAL) paper presented at
2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 14-18, 2024. Abu Dhabi, UAE

Distributed Coverage Control for Spatial Processes Estimation with Noisy Observations

Mattia Mantovani *Student Member, IEEE*, Federico Pratissoli *Member, IEEE*, and Lorenzo Sabattini *Senior Member, IEEE*

***Abstract*—The present study addresses the challenge of effectively deploying a multi-robot team to optimally cover a domain with unknown density distribution. Specifically, we propose a distribute coverage-based control algorithm that enables a group of autonomous robots to simultaneously learn and estimate a spatial field over the domain. Additionally, we consider a scenario where the robots are deployed in a noisy environment or equipped with noisy sensors. To accomplish this, the control strategy utilizes Gaussian Process Regression (GPR) to construct a model of the monitored spatial process in the environment. Our strategy tackles the computational limits of Gaussian processes (GPs) when dealing with large data sets. The control algorithm filters the set of samples, limiting the GP training data to those that are relevant to improving the process estimate, avoiding excessive computational complexity and managing the noise in the observations. To evaluate the effectiveness of our proposed algorithm, we conducted several simulations and real platform experiments.**

***Index Terms*—Multi-Robot Systems; Distributed Robot Systems**

I. Introduction

MULTI-VEHICLE systems have various applications, two of which are coverage control and autonomous exploration. In these applications, a group of networked robots work together to explore an unknown environment and maximize its coverage. This is achieved by coordinating the robots to concentrate in areas of higher interest, resulting in a more effective and efficient exploration. The focus of this paper is on addressing the spatial coverage problem [1] in an unknown environment with a limited sensing range multi-robot systems. This approach, which is based on Voronoi partitioning of the environment, ensures that networked robots converge to a configuration that maximizes the coverage of the environment's most important areas. These areas are defined by a probability density function that is specified over the environment. In recent years, various coverage control strategies have been developed that consider heterogeneous and limited robot capabilities [2], [3], as well as different scenarios associated with different density functions [4], [5]. In most cases, the density distribution is used to describe a spatial process or phenomenon that a group of robots is responsible for monitoring. However, it is commonly assumed that the density function is known in advance to each of the robots involved. This assumption cannot be applied when the multi-robot system is deployed in an environment that is either partially known or completely unknown. This work aims to address the problem of efficiently sampling data and collecting information in order to obtain an accurate reconstruction of a physical process. These types of algorithms have been extensively studied in the field of informative sampling and information gathering [6], [7]. The GP is a commonly used tool for reconstructing a signal from a limited number of samples and making predictions about the signal across its domain [8]. It provides a statistical model that characterizes the distribution over functions and enables uncertainty quantification in the prediction process.

Control strategies to generate the trajectory for a single robot with the aim to optimize the information gathering in an unknown environment have been widely studied in the last years [9]–[11]. Only a few works can be found in this field that exploit a team of robots. A multi-robot approach has been presented by [12], which is based on a discretization of the environment exploiting Voronoi partitioning. Alternatively, the work presented recently by [13] coordinates a group of robots scheduling meeting points to exchange data.

Existing literature in this research field with multi-robot approaches typically neglects the optimal deployment of robots based on the estimated process in the environment. Instead, robots are usually directed towards areas that require sampling to improve the accuracy of the estimation. This paper proposes the use of a coverage-based control algorithm to optimize the deployment of robots in the environment according to the density distribution. This distribution is learned and estimated by the robots through an exploration of the environment. GPs are well-suited for this task as they provide both the estimation of the spatial signal and the associated uncertainties across the domain of estimation.

Some works have been recently dedicated to the problem of multi-robot optimal coverage with the aim of cooperative exploration and estimation of unknown processes [14]. In [15] the authors focus on a methodology for regulating the sampling process. The approach does not consider the exploration of unknown areas in the environment, nor does it address the trade-off between exploration and exploitation. Instead, the robots start from a spread configuration and then follow the estimated density function once it has been estimated.

Manuscript received: October, 31, 2023; Revised January, 25, 2024; Accepted March, 04, 2024.

This paper was recommended for publication by Editor M. Ani Hsieh upon evaluation of the Associate Editor and Reviewers' comments. This work was supported by the AI-DROW Project through the Italian Ministry for University and Research under the PRIN 2022 program, funded by the European Union - Next Generation EU

Authors are with Department of Sciences and Methods for Engineering (DISMI), University of Modena and Reggio Emilia, Italy {name.surname}@unimore.it

Digital Object Identifier (DOI): see top of this page.

In [16], [17] the authors propose two control strategies that enable the multi-robot team to optimally learn the spatial field and optimally cover it. On the other hand, [18] primarily focuses on the decentralized aspect of such control strategies. Differently, in [19] the robots, starting in random positions, are coordinated to optimally cover the spatial field, which is estimated from the data sampled along the way. Moreover, no filtering strategy on the acquired samples has been provided. Some works are more focused on a real world implementation without considering an optimal strategy to explore and cover the spatial process with a multi-robot system [20], [21]. Other approaches divide the control strategy in two different phases, an initial optimal exploration phase to estimate the density function and a final exploitation phase where the robots use the density function to optimally cover the environment [22], [23]. It is noteworthy that the issue of optimally balancing the coordination of a group of robots for estimating a spatial field and their coordination for optimal coverage of the same field remains a current topic, which has been addressed by only a few studies [16]–[18]. However, these studies have not taken into account the computational complexity introduced by the GP regression, nor have they considered a sampling strategy, which is essential for obtaining an accurate prediction of the spatial field and preventing excessive computational complexity.

Moreover, an issue that is generally ignored or only marginally considered is the presence of noise in the environment or on the sensors of the robots. Noise can have a significant impact on the accuracy of the spatial process estimation and can lead to sub-optimal robot deployment strategies. Therefore, it is important to take into account the presence of noise and develop appropriate filtering and control strategies to mitigate its effects. Kalman filtering techniques can be exploited to deal with noise-corrupted observations. However, these approaches assume that the state-transition matrices in the estimation systems are known a priori, which is usually not the case in practice [24]. In summary the contributions of this work are as follows:

- A novel control strategy that optimizes the exploration and exploitation trade-off to learn, estimate and cover a spatial process with a team of robots, even with noisy observations.
- A novel approach to filter the collected data efficiently, addressing the GP computational complexity.
- An efficient sharing of sampled data by the robot, making it feasible to deploy a distributed approach.

II. Notation and Definitions

We denote by $\mathbb{N}$, $\mathbb{R}$, $\mathbb{R}_{\geq 0}$, and $\mathbb{R}_{>0}$ the set of natural, real, real non-negative, and real positive numbers. Given $x \in \mathbb{R}^n$, let $\|x\|$ be the Euclidean norm. Let $\mathbb{F}(\mathbb{R}^2)$ be the collection of finite point sets in $\mathbb{R}^2$. We can denote an element of $\mathbb{F}(\mathbb{R}^2)$ as $\mathcal{P} = \{p_1, \dots, p_n\} \subset \mathbb{R}^2$, where $\{p_1, \dots, p_n\}$ are points in $\mathbb{R}^2$. With $\overline{B}(p_i, R) = \{x \in \mathbb{R}^2 | \|x - p_i\| \leq R\}$, we denote the closed ball in $\mathbb{R}^2$ centered at p_i with radius R, for $p_i \in \mathbb{R}^2$ and $R \in \mathbb{R}_{>0}$.

The *limited Voronoi partitioning* is defined on a polygonal environment in $\mathbb{R}^2$ following the idea presented in [2]. In the rest of the paper, we will use $Q \subset \mathbb{R}^2$ to denote the polygonal environment to be covered by the robots. An arbitrary point in Q is denoted by $x \in Q$. Let $N_{(R)}(p_i, \mathcal{P})$ be the set of neighbors locations of the agent i in the sensing range with radius R. Then $\mathcal{P}$ is a set of n points $\{p_1, \dots, p_n\}$ in Q and $P_i = \{p_i \cup N_{(R)}(p_i, \mathcal{P})\}$ the set of points including the robot position p_i and its neighbors locations. The limited Voronoi partitioning generated by $\mathcal{P}$ consists of the set $\mathcal{V}^r(\mathcal{P}) = \{V_1^r(P_1), \dots, V_n^r(P_n)\}$, where:

$$V_i^r(P_i) = \{x \in \overline{B}_{\cap Q}(p_i, r) \mid \|x - p_i\| \leq \|x - p_j\|, \forall p_j \in N_{(R)}(p_i, \mathcal{P})\}, \quad (1)$$

where $\overline{B}_{\cap Q}(p_i, r) = \{Q \cap \overline{B}(p_i, r)\}$ is the intersection between the environment and the ball of radius r for robot i and r consists in half of the sensing radius R: $r = R/2$. Two agents are said to be *Voronoi neighbours* if $V_i^r(P_i) \cap V_j^r(P_j) \neq \emptyset$. We refer to [25] for a discussion about the Voronoi diagrams and to [2] for information about the limited Voronoi partitioning.

III. Background

A. Coverage Control

We will now briefly summarize the solution to the coverage problem for the multi-robot system control previously introduced in [2], upon which we build.

The control strategy is based on the definition of a performance function that has to be maximized in order to obtain the optimal coverage of the group of robots over Q.

Moreover, an integrable probability density function $\phi : Q \to \mathbb{R}_{\geq 0}$ is defined to encode which are the areas of the environment Q with highest relevance. We have a limited Voronoi partitioning $\mathcal{V}^r(\mathcal{P})$ of the environment Q into, so-called, n Voronoi cells $\{V_1^r, \dots, V_n^r\}$, as depicted in [2].

The *optimization* function $H : Q \to \mathbb{R}$ is defined as:

$$\mathcal{H}_{\mathcal{V}}^r(\mathcal{P}) = \sum_{i=1}^{n} \int_{V_i^r} f(\|x - p_i\|)\phi(x)dx, \quad (2)$$

where each agent covers its designated environment portion. The performance function $f(x)$ is chosen as in [2, Eq. (12)], for a limited Voronoi partitioning, namely:

$$f^r(x) = -min(x^2, r^2) \quad (3)$$

We compute the gradient of the optimization function to solve the optimization problem, obtaining:

$$\frac{\partial \mathcal{H}_{\mathcal{V}}^r(\mathcal{P})}{\partial p_i} = 2M_{V_i^r}(C_{V_i^r} - p_i), \quad (4)$$

as demonstrated by Theorem 1 in [2]. $M_{V_i^r}$ and $C_{V_i^r}$ are respectively the mass and the centroid of the *i*-th limited Voronoi cell V_i^r. These elements are weighted by the Voronoi cell density function ϕ of robot i at p_i in $V_i^r \subset Q$.

Thus, the mass and centroid can be computed as follows:

$$M_{V_i^r} = \int_{V_i^r} \phi(q)dq, \quad (5)$$

$$C_{V_i^r} = \frac{1}{M_{V_i^r}} \int_{V_i^r} q\phi(q)dq. \quad (6)$$

The solution to the coverage problem is achieved when each agent is located at the centroid of its Voronoi cell, such that $p_i = C_{V_i^r}$, $\forall i$. In particular, this is used to design the control input for each robot that drives the multi-robot system to a configuration that optimizes the coverage of the environment according to the density function ϕ:

$$u_i = k(C_{V_i^r} - p_i), \tag{7}$$

where $k \in \mathbb{R}_{>0}$ is a proportional gain. The limited Voronoi partitioning $\mathcal{V}^r(\mathcal{P})$ of the region Q is continuously updated with the control input. More details on coverage control and the used methodology can be found in [1], [2], [26].

B. Gaussian Process Regression

GPs are a powerful tool that can be used to model an unknown function starting from a few samples and have the ability to provide the uncertainty over the predicted values of the function [27]. In particular, a GP defines a prior distribution over the space of functions such that, together with the observed data, it can lead to a posterior multivariate Gaussian distribution able to predict the function behaviour. A GP is completely specified by its mean function $\mu(x)$ and covariance function $k(x, x')$. Let $\phi(x)$ be the environmental spatial process that the GP has to model, then, $\mu(x)$ represents the expected value of ϕ at input x, and $k(x, x')$ represents the correlation between two variables x and x'. We assume the robots' sensory observations of the spatial process are white Gaussian noisy measurements:

$$y = \phi(x) + \mathcal{N}(0, \sigma_r^2). \tag{8}$$

Thus, we define the set of noise values for every robot sensor as $\nu = \{\sigma_{r,1}, \sigma_{r,2}, \dots, \sigma_{r,n}\}$.

In the GPR, we have that $\phi(x)$ can be modeled as a GP:

$$\phi(x) \sim \mathcal{GP}(\mu(x), k(x, x')). \tag{9}$$

Without loss of generality, to simplify the notation and to reduce the computational complexity, we consider a zero-mean GP. This assumption can be done since the zero-mean assumption is on the prior distribution, the posterior will have an updated non-zero mean according to the observations. The mean function allows to incorporate prior knowledge or beliefs about the underlying data before observing any actual data points. A GP with a mean $\mu(x) \neq 0$ can be treated by a change of variables. The learning and the estimation of the process is defined by the so called *kernel* $k(x, x')$. The kernel has to be chosen according to the type of the function we want to estimate, e.g. periodic, linear, quadratic, etc. Since we are assuming the spatial process has a smooth behaviour, we consider a squared exponential kernel:

$$k(x, x') = \sigma_f^2 \exp\left(-\frac{\|x - x'\|^2}{2\lambda^2}\right), \tag{10}$$

where λ and σ_f are hyper-parameters that can be estimated from the sampled data by maximizing the likelihood function. We have the observations $\mathcal{D}_t = \{X_t, y_t\}$ collected by the robots until time t, with $X_t = [x_1, \dots, x_n]$ the set of the visited locations. The objective is to predict spatial process values in unvisited locations $X^* = [x_1^*, \dots, x_n^*]$. Hence, we can define the covariance function $k(X^*, X^*)$ and the predicted values $\phi^* = [\phi(x_1^*), \dots, \phi(x_n^*)]$. Applying Bayesian theorem, as in [27], reveals ϕ^*'s conditional distribution:

$$(\phi^* | X_t, y_t, X^*) \sim \mathcal{N}(\mu(X^* | D_t), \Sigma(X^* | D_t)), \tag{11}$$

which is a multivariate normal distribution with mean:

$$\mu(X^*) = k(X^*, X_t)^T [k(X_t, X_t) + \sigma_\nu^2 I]^{-1} y_t \tag{12}$$

and covariance matrix:

$$\Sigma(X^*) = k(X^*, X^*) - k(X^*, X_t)^T [k(X_t, X_t) + \sigma_\nu^2 I]^{-1} k(X^*, X_t) \tag{13}$$

where $k(X^*, X_t)$ is a covariance vector. The spatial process estimation is associated with the mean of the distribution, while the uncertainty of this estimation is described by the covariance matrix. The process predicted values are strongly influenced by how similar the observed data X_t are to the points X^* that we want to predict. This correlation is described by the kernel $k(X^*, X_t)$.

The kernel hyper-parameters $\lambda, \sigma_f, \sigma_\nu$ are unknown and need to be inferred from the sampled data. The estimation of the hyper-parameters can be optimized by maximising the marginal (log) likelihood function. Given the observations $\mathcal{D}_t$ and the hyper-parameters $\theta = (\lambda, \sigma_f^2, \sigma_\nu^2)$, the log marginal likelihood function can be expressed in closed form:

$$\log p(y | X, \theta) = -\frac{1}{2} y^T K_y^{-1} y - \frac{1}{2} \log |K_y| - \frac{n}{2} \log 2\pi \tag{14}$$

where $K_y = K(X_t, X_t) + \sigma_\nu^2 I$. For more details the reader is referred to [27], [28].

It should be emphasized that the computational complexity of GPs can become a significant issue when dealing with large datasets [29], [30], especially if the control algorithm is executed on board the robots. In particular, the algorithm has a time complexity of $\mathcal{O}(N^3)$ and a memory complexity of $\mathcal{O}(N^2)$, where N represents the total number of sampled data. The presence of noise in the observations directly impacts not only the estimate but also the associated uncertainty and computational complexity. This is due to the fact that an increased number of samples is needed to enhance the accuracy of the process estimate. As a result, the algorithm can be hardly executed onboard the platforms in the presence of high noise and large datasets.

IV. Problem Statement

In this work we get rid of the assumption that the density function $\phi(x)$ in (2) is known beforehand. For this reason, we need to define a strategy that allows the robots to learn and estimate online the spatial process from the data sampled during the motion. We consider a multi-robot system constituted by n robots that move in a 2-dimensional space. We assume each robot to be modeled as a single integrator system,[1] whose position $p_i \in \mathbb{R}^2$ evolves according to

$$\dot{p}_i = u_i, \tag{15}$$

where $u_i \in \mathbb{R}^2$ is the control input, $\forall i = 1, \dots, n$. The set of robots is represented by $\mathcal{P} = \{p_1, \dots, p_n\}$. We consider the following setting:

1) *Convex unknown environment*: the multi-robot system has to maximize coverage of an unknown spatial field distributed in the environment, a convex polytope Q.
2) *Time invariant signal*: the signal does not evolve over time and is therefore considered time invariant.

We also consider the following assumptions:

1) *Limited sensing capabilities*: each robot is able to measure the position of neighboring robots and objects, and to detect the boundaries of the environment Q within its limited sensing range. This allows the computation of the limited Voronoi partitioning of the environment as defined in (1).
2) *Communication capabilities*: The team of robots possesses restricted communication capabilities, enabling the exchange of information and data only when the robots are within their respective sensing ranges.
3) *Spatial process behaviour*: The behavior of the spatial process in the environment exhibits a smooth characteristic, making it suitable for modeling through the use of a GP with a Squared Exponential Kernel.
4) *Noisy observations*: Sensor sampling is affected by noise that varies depending on the quality of the sensor.

The problem in this paper is formalized as follows:

Problem *Define a distributed control strategy that enables a multi-robot system to optimally explore and estimate an unknown spatial field while concurrently covering it in the environment Q, in the presence of noisy observations.*

V. Exploration-Exploitation Problem

The primary objective of this study is to perform efficient exploration of the unknown spatial field and utilize GPs to learn and optimally cover the environment. By leveraging the information obtained from the GP, the robot can concentrate their coverage efforts on the areas of higher interest. Equation (11) provides the distribution of the density function: the mean function corresponds to the estimation, while the covariance function represents the uncertainty associated with the estimation. Bayesian Optimization relies on an essential component known as the *acquisition function*, which plays a crucial role in balancing the trade-off between exploration and exploitation [27]. This function combines both the mean and variance to formulate a criterion that maximizes utility, facilitating the process of learning about the function or utilizing its information effectively. In coverage-based control, the multi-robot system employs a substitute for the density function that coincides with an acquisition function, in our study:

$$\phi'_t(x) = \sigma_{t-1}(x) + W_t \mu_{t-1}(x), \tag{16}$$

where $\sigma_{t-1}(x) = \sqrt{\Sigma_{t-1}(x, x)}$ is the standard deviation. The weight W_t is a term introduced in order to balance the exploration and exploitation process. We propose a novel approach for defining the weight W_t, which is selected according to the preference over the trade off between exploration and exploitation and is defined by:

$$W_t = C_1 \arctan(C_2 t). \tag{17}$$

The idea behind this definition is that, during the initial phase, the environment is inherently unknown. Consequently, we prefer a strategy that prioritizes exploration by the robots. As time progresses, the variable W_t gradually converges to a value that strikes a balance between the exploration and exploitation behaviors of the robots. The constants C_1 and C_2 are chosen by the user to tune the exploration-exploitation trade off according to the monitoring scenario. High values of W_t mean that the exploitation is preferred over the exploration, and viceversa. These constants can be learnt similarly to the GP hyper-parameters, but this procedure is left as a topic for future works. The substitute density function $\phi'_t(x)$ aims to adapt the coverage control strategy to push the robots to the high utility areas. The utility distribution in the environment is more weighted towards the exploration if the density function is completely unknown ($\phi'_t(x) \sim \sigma_{t-1}(x)$) and is more weighted towards the exploitation if the density function has been sufficiently explored already ($\phi'_t(x) \sim \mu_{t-1}(x)$).

VI. Noisy Data Acquisition

Given the decentralized architecture of our system, each robot autonomously computes its individual GP. This locally computed GP, denoted as $\mathcal{GP}_i$, encompasses unique kernel hyper-parameters $\lambda_i, \sigma_{f,i}, \sigma_{\nu,i}$ and manages an exclusive dataset $\mathcal{D}_{t,i}$. Stable interconnections among the robots allow for data sharing, within their visual range, promoting the exchange of collected sample datasets with neighboring robots. This collaborative data sharing enables the robots to improve their process estimates. This entails that every local $\mathcal{GP}_i$ will eventually converge to a unique and less uncertain GP estimation. The present study is focused on tackling the challenge of managing noisy observations. We remind that the observations of each robot are influenced by a certain amount of noise as detailed in (8). The hyper-parameter $\sigma_{\nu,i}$ represents the noise estimated in the GP computation by maximizing (14). Although each robot is affected by a unique value of noise correlated to the sensor quality, this value will eventually converge to a hyper-parameter equal for all robots, due to the collaborative data sharing. The noise directly influences the mean function and the variance function (respectively in (12) and (13)). The dataset $\mathcal{D}_{t,i}$ is expected to not only contain samples collected by the i-th robot but also to include data from the shared datasets of its neighboring robots, as previously mentioned.

[1] We would like to remark that, even though the single integrator is a very simplified model, it can still effectively be exploited to control real mobile robots: using a sufficiently good trajectory tracking controller, the single integrator model can be used to generate velocity references for widely used mobile robotic platforms, such as wheeled mobile robots [31], and unmanned aerial vehicles [32].

In this work, we propose an algorithm to efficiently filter the data collected by robots during the exploration. The proposed policy filters and selects only the samples that provide a significant contribution to the estimation and prediction of the spatial process. Specifically, a new sample is added into the dataset during robot movement only if it can considerably improve the model and reduce the uncertainty. GPs entail two types of uncertainty: *aleatoric* and *epistemic*. Aleatoric uncertainty stems from inherent data randomness, like noise in observations, and it cannot be reduced. Epistemic uncertainty, on the other hand, refers to lack of knowledge of the model, often due to insufficient training data. Consequently, regions with high epistemic uncertainty indicate areas requiring further exploration by robots. This can be clearly seen in Fig. 1. The goal is to minimize the epistemic uncertainty, which reflects the lack of knowledge in our model. By minimizing epistemic uncertainty the model becomes more confident in its predictions and attains a better understanding of the explored domain. We address the uncertainty by employing a confidence interval centered around the mean (Fig. 1). The z-score, denoted as z, quantifies the deviation of a value from the mean in terms of standard deviations. Following the *empirical rule*, we establish a 95% confidence interval around the mean, resulting in a z-score of 1.96. As mentioned before, in the presented strategy, the robot selectively filters observations and incorporates additional samples into the dataset only if they contribute to an improved understanding of the underlying process. To achieve this, we introduce a threshold ϵ that reflects the desired level of epistemic uncertainty or confidence in the predictions. Specifically, we define this threshold as a percentage error around the estimated mean value, which is aligned with the maximum variability inherent in the estimated process. The definition of ϵ is as follows:

$$\epsilon = e \cdot \Delta, \tag{18}$$

where $\Delta = (\mu_{i,max} - \mu_{i,min})$ is the range between the minimum and maximum values of the estimated mean, and the constant e is typically set to 0.05, corresponding to a 5% tolerance of error. It's important to note that while smaller errors enhance signal estimation accuracy, they necessitate a larger dataset and a higher computational effort.

When the *i*-th robot collects a new sample, we compute the posterior $\mathcal{GP}_i$ in this points and get the epistemic uncertainty $\sigma_{y,i}$. The sample is then added to the dataset only if:

$$\sigma_{y,i} \geq \epsilon / z. \tag{19}$$

To tackle these challenges, we propose a decentralized filtering approach for effectively managing the acquired observations, as outlined in Alg. 1. By employing this filtering strategy, we achieve a harmonious balance between the error in the learned spatial process through $\mathcal{GP}_i$ and the number of noisy observations acquired. The proposed algorithm is intentionally designed to be decentralized, with execution taking place on individual robots indexed by i. Efficient exchange of datasets among neighbors is facilitated through the implementation of a filtering strategy. Each robot can filter the samples shared by its neighbors using the same method employed for filtering its own collected data. Only the samples that enhance the accuracy of the spatial process estimation are integrated into the dataset.

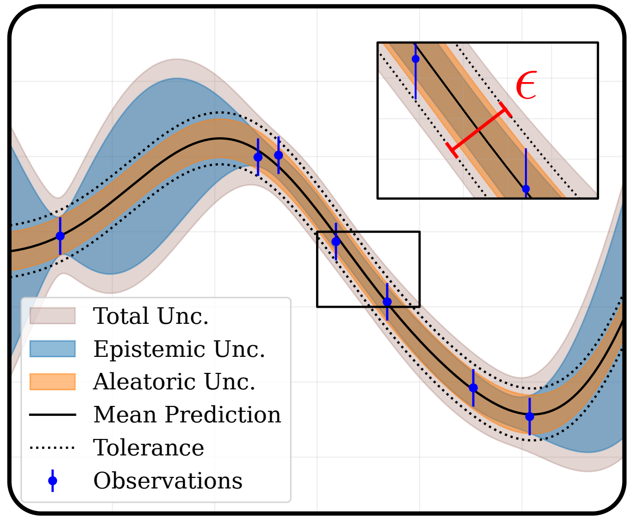


Fig. 1. The figure shows an example GP. Total uncertainty (grey) includes aleatoric (orange) and epistemic (blue) around the mean prediction. They are shown with a 95% confidence interval with $z = 1.96$. Aleatoric uncertainty remains constant once noise in observations is known, while epistemic uncertainty peaks in areas where there is few data. The tolerance (black dotted line) sets the minimum desired epistemic uncertainty for sufficient process knowledge, making additional samples unnecessary.

Algorithm 1: Filter Algorithm on the i-th robot

Input: ϵ, z, n robot neighbors
1 Collect an observation $o_i = (X_i, y_i)$ at time t
2 $\mu_i, \Sigma_i \leftarrow \mathcal{GP}_i(o_i, \mathcal{D}_{t,i})$
3 $\Sigma_i \rightarrow \sigma_{y,i}$
4 **if** $\sigma_{y,i} \geq \epsilon / z$ **then**
5     $\mathcal{D}_{t,i} \cup \{o_i\}$
6   **for** $j = 1$ **to** n **do**
7   **for** o_k **in** $\mathcal{D}_{t,j}$ **do**
8   **if** $o_k \notin \mathcal{D}_{t,i}$ **then**
9                 $\mu_k, \Sigma_k \leftarrow \mathcal{GP}_i(o_k, \mathcal{D}_{t,i})$
10                 $\Sigma_k \rightarrow \sigma_{y,k}$
11   **if** $\sigma_{y,k} \geq \epsilon / z$ **then**
12                     $\mathcal{D}_{t,i} \cup \{o_k\}$
13   **end**
14   **end**
15   **end**
16   **end**
17 **end**

The Alg. 1 operates independently on each robot, and follows a sequence of steps to achieve a precise estimate of the spatial process. The algorithm takes the threshold ϵ as input, as defined in (18), the confidence interval z and the n neighbours of the robot. The robot collects a sample $o_i = (X_i, y_i)$ (line 1). To obtain the mean and the covariance in the collection point, the posterior prediction of the $\mathcal{GP}_i$ is applied (line 2) according to (12) and (13). We calculate the epistemic uncertainty $\sigma_{y,i}$ of the estimated covariance from the $\mathcal{GP}_i$ (line 3) for sample o_i and compare it to ϵ / z according to (19). If it exceeds the threshold, we add the sample to the dataset since this condition suggests the new data enhance the signal estimation (lines 4-5). In the second part of the algorithm, the datasets shared by neighbouring robots are filtered. For each sample in the datasets of neighboring robots (lines 6-7), if it is not already included in the actual dataset $\mathcal{D}_{t,i}$ (line 8), we compute the posterior prediction with (12) and (13) in order to obtain the mean and the covariance (line 9). We obtain the epistemic uncertainty $\sigma_{y,k}$ for sample o_k (line 10) and compare it to the threshold value ϵ/z. If it exceeds the threshold, we add the sample to the dataset since this condition means that the new data can contribute to achieve a better knowledge of the spatial process and the signal estimation (lines 11-12). Once the filtering process is finished, the updated dataset is used to optimize the hyper-parameters of the $\mathcal{GP}_i$ in accordance to (14) and the process can start again.

VII. Experimental Validation

In this section we report the results of simulations and experiments carried out to verify the proposed control strategy. Some representative trials are shown in the attached video. Our approach effectively tackles real-world scenarios. To create a realistic simulation, we generate random processes for the estimation. Conducting real experiments with a fleet of robots for environmental sampling is left to future works.

A. Simulated Experiments

In this study, we evaluated the efficacy of the proposed algorithm through simulations that included exploration-exploitation trade-off and samples filtering. For this purpose, we carefully tuned our hyper-parameters to ensure thorough testing. Specifically, we configured the parameters by setting $C_2 = 0.3$ and then obtaining $C_1 = 6.5$ resulting in $W_t = 10$ after 30 seconds, because there are no physical constraints limiting the robot movement dynamics in simulations. We opt for a robot sensing range denoted by $R = 2$ meters. This choice strikes a well-balanced trade-off, favoring exploration over exploitation in the initial phases of the process, with a subsequent shift towards prioritizing the convergence of the robots to the target area. Additionally, we used different values of the variances in the vector ν to simulate different levels of white Gaussian noise on the sensory observations. Specifically, we randomly selected the parameters in the range between 0.01 (representing a low level of noise) and 0.3 (representing a high level of noise), thus obtaining $\nu = \{0.163, 0.227, 0.044, 0.101, 0.09, 0.130\}$.

Results of a representative trial are reported in Fig. 2. In particular the figure shows how a team of robot deployed in an unknown domain is able to explore and learn it, covering in order to monitor some simple events in the environment. The estimate of the process is affected by an error due to the presence of noise in the robots observations and because they operate in a distributed way. In this context, each robot reaches a sub-optimal solution of the process to be monitored. With that said, thanks to the proposed filter policy, we are able to reduce the size of the dataset and ensure the exclusion of samples that do not improve the understanding of the process.

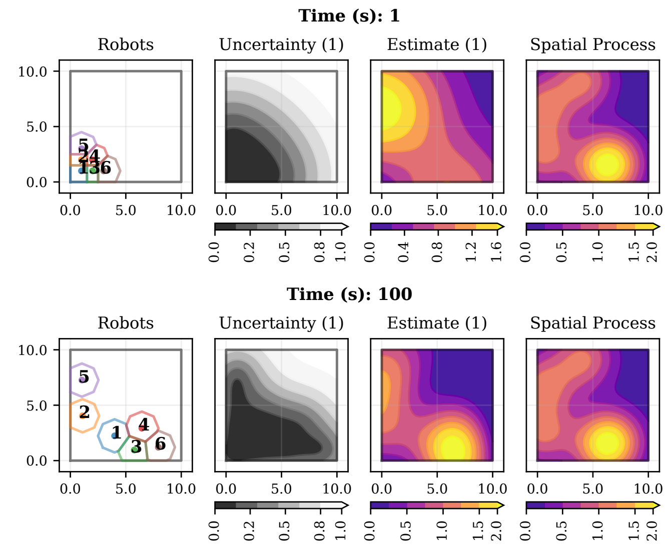
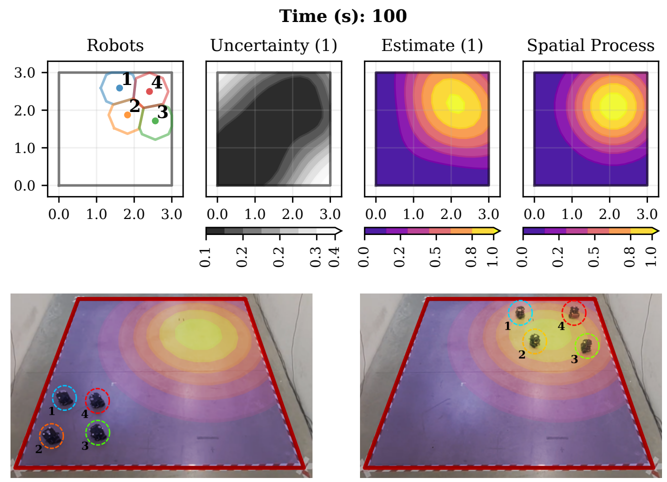


Fig. 2. The figure depicts a simulated robot team in an unfamiliar domain, characterized by an event requiring estimation and monitoring. Each subfigure represents different stages of the process, progressing from left to right. The first subfigure shows the robots (colored dots) with their local partition of the Voronoi diagram (colored perimeters). The second subfigure represents the uncertainty associated with the spatial field estimation of the $\mathcal{GP}_1$. In the third subfigure, the spatial field estimated by the first robot is depicted. The fourth subfigure illustrates the actual spatial process. At $t = 1s$, the initial stage is depicted, with a few collected samples by each robot, while at $t = 100s$, the team optimally covers the interest areas with higher probability, monitoring the event. For clarity, we've shown an example with just the first robot $\mathcal{GP}_1$. We refer to the attached video for a more comprehensive demonstration.

Moreover, this leads to a reduction in the computational load on the individual robot units, as will be discussed in details in Sec. VIII.

B. Hardware Experiments

In this section we report some of the experiments we performed with a team of *TurtleBot 3 Burger* robots. Feedback linearization was exploited to implement the proposed control action on these differential drive robots. The experiments were carried out employing a value of $C_1 = 7.28$ and $C_2 = 0.05$ resulting with the same W_t as described in Sec. VII-A after 100 seconds. We choose these parameter values to achieve a slower dynamic given the limitations of our real environment. We also set the robot sensing range at $R = 1$ meters. To accommodate a moderate level of noise in the system, we set the noise level on the robot sensors at $\nu = \{0.133, 0.207, 0.064, 0.121\}$. Fig. 3(a) shows a team of robots placed in random positions at the start of a representative run of the experiments. In Fig. 3, the spatial process learning is reported, together with the coordinated positioning of the robot team around the high-density peak they have identified. The first subfigure at the top showcases the final stage of the learning process at $t = 100s$, featuring the $\mathcal{GP}_1$ estimate and uncertainty for the first robot. Given that each robot independently computes its own $\mathcal{GP}_i$, we have presented the $\mathcal{GP}_1$ results for the first robot as an illustrative example. Compared to simulations (see Sec. VII-A), the learning process takes longer in a real-world environment because mobile robots move according



(a) $t = 1s$. (b) $t = 100s$.

Fig. 3. The figure shows a team of robot tasked to explore an unknown domain and monitor a simple event on it. The first subfigure displays the spatial process that the group of robots has learned over a time period. The second and third subfigures depict the Turtlebot3 Burger robots in their initial and final positions, respectively. In the final configuration, the robots are evenly distributed around the density peak describing the learnt event.

to their constraints and maximum speeds. This results in a greater number of samples taken, closer together for the same sampling rate, and more time required for coverage.

VIII. Performance Evaluation

To gather data for our evaluations, we conducted multiple simulations involving the same setup described in Sec. VII-A. We varied the process to be estimated by randomly generating it in each simulation. After running multiple simulations, our evaluations were based on several metrics. Firstly, we assessed the Root-Mean-Square Error (RMSE) between the spatial process inferred from each robots' $\mathcal{GP}_i$ and the actual spatial process, as depicted in Fig. 4. This analysis reveals that, over time, each robot's estimate of the spatial process gradually converges toward the true process. Additionally, Fig. 5 displays the dataset size trend, which, over time, tends to stabilize. This demonstrates the effectiveness of our filter strategy, which selectively retains relevant samples collected by the robots. Furthermore, we used the optimization function (2) for coverage control, which evaluates the robots locations in relation to the spatial process location. This helps us to analyze the convergence of the multi-robot system on the event to be monitored. It is worth noting that, with the proposed filter strategy, we are able to achieve results close to the ideal coverage where there is an accurate and complete knowledge of the environment, as shown by the blue and red lines in Fig. 6. It is important to note that, due to the distributed nature of the system, limited communication capabilities and observation noise, achieving perfect coverage and a zero RMSE is unlikely, leading to a sub-optimal outcome. However, with our proposed strategy, each robot selects only key samples, aiding spatial process understanding.

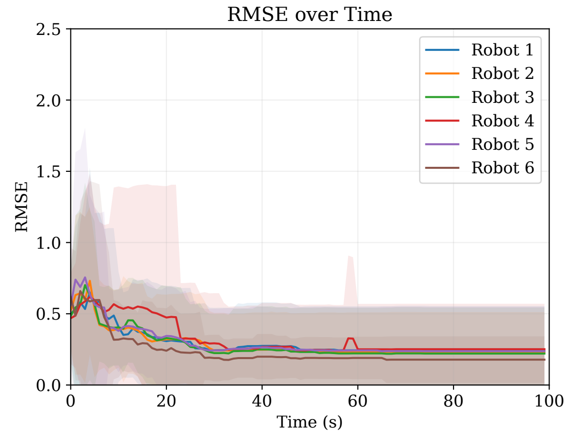


Fig. 4. The figure illustrates the RMSE between the spatial process deduced from each robot's $\mathcal{GP}_i$ and the ground truth spatial process. It presents the RMSE averages for multiple instances across several simulation runs, along with a 95% confidence interval. It is shown that, over time, the RMSE of each robot estimate tends to decrease but never reaching zero due to the conditions under which the robots operate.

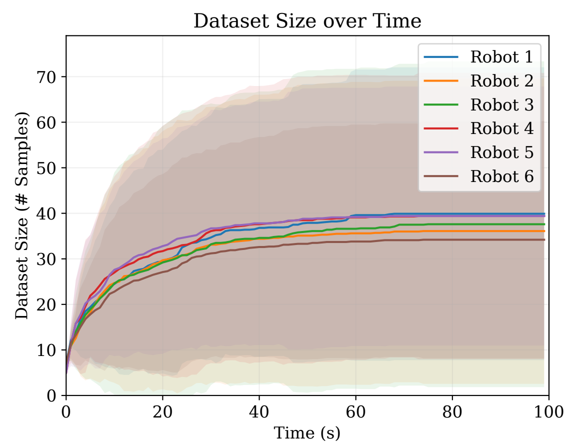


Fig. 5. The graph depicts the size of the robots' datasets $\mathcal{D}_{t,i}$. It presents the averages of $\mathcal{D}_{t,i}$ size for multiple instances across several simulation runs, along with a 95% confidence interval. It is shown that, over time, the size of $\mathcal{D}_{t,i}$ tends to remain constant, demonstrating the effectiveness of the filter in selecting only those samples that contribute to the estimation of the process. The high variance is due to the type of process to estimate, with more complex processes requiring a larger number of samples.

IX. Conclusion and Future Work

This paper proposes an algorithm to control a group of robots in order to efficiently learn and estimate a spatial field, while also covering it effectively in the presence of noisy observations. The algorithm strikes a balance between exploring the environment to estimate the field and exploiting that information to cover the field optimally. To achieve this, the methodology presented in the paper utilizes GPs to model the spatial field from noisy data collected by the team of robots in the environment. We introduced a new approach to limit the size of the dataset, which is necessary to avoid computational issues when dealing with large datasets and noisy observations. As a future work, we aim to create an innovative methodology for addressing time-varying spatial processes. To enhance adaptability to spatial changes, we are

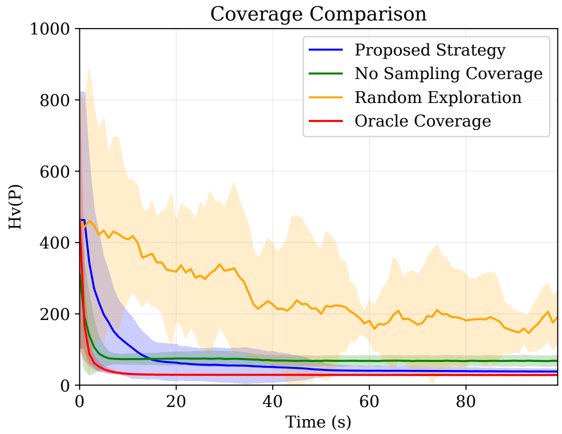


Fig. 6. In a scenario where the robots are tasked with monitoring a spatial process, the performance comparison of the suggested coverage-based control method with three alternative approaches is shown in the figure. The orange line represents the random exploration algorithm where the robots randomly explore the domain. The green line shows the coverage without exploiting any learned information about the spatial process. In this way the robots are not pushed towards the higher interest but they will arrange to optimally cover the entire domain. The red line represents the ideal coverage with accurate and complete knowledge of the environment, and the blue line represents the coverage with the proposed filter strategy.

investigating the implementation of a Neural Network-based learning approach for the parameter W_t in order to make it dynamically responsive to the event variations.

References

[1] J. Cortes, S. Martinez, T. Karatas, and F. Bullo, "Coverage control for mobile sensing networks," *IEEE Transactions on robotics and Automation*, vol. 20, no. 2, pp. 243–255, 2004.

[2] F. Pratissoli, B. Capelli, and L. Sabattini, "On coverage control for limited range multi-robot systems," in *2022 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2022, pp. 9957–9963.

[3] M. Santos and M. Egerstedt, "Coverage control for multi-robot teams with heterogeneous sensing capabilities using limited communications," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 5313–5319.

[4] S. G. Lee, Y. Diaz-Mercado, and M. Egerstedt, "Multirobot control using time-varying density functions," *IEEE Transactions on robotics*, vol. 31, no. 2, pp. 489–493, 2015.

[5] W. Li and C. G. Cassandras, "Distributed cooperative coverage control of sensor networks," in *Proceedings of the 44th IEEE Conference on Decision and Control*. IEEE, 2005, pp. 2542–2547.

[6] K.-C. Ma, L. Liu, and G. S. Sukhatme, "Informative planning and online learning with sparse gaussian processes," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 4292–4298.

[7] M. Lauri, E. Heinänen, and S. Frintrop, "Multi-robot active information gathering with periodic communication," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 851–856.

[8] A. Viseras, D. Shutin, and L. Merino, "Robotic active information gathering for spatial field reconstruction with rapidly-exploring random trees and online learning of gaussian processes," *Sensors*, vol. 19, no. 5, p. 1016, 2019.

[9] C. Xiong, H. Zhou, D. Lu, Z. Zeng, L. Lian, and C. Yu, "Rapidly-exploring adaptive sampling tree*: A sample-based path-planning algorithm for unmanned marine vehicles information gathering in variable ocean environments," *Sensors*, vol. 20, no. 9, p. 2515, 2020.

[10] M. Bresciani, F. Ruscio, S. Tani, G. Peralta, A. Timperi, E. Guerrero-Font, F. Bonin-Font, A. Caiti, and R. Costanzi, "Path planning for underwater information gathering based on genetic algorithms and data stochastic models," *Journal of Marine Science and Engineering*, vol. 9, no. 11, p. 1183, 2021.

[11] E. Guerrero, F. Bonin-Font, and G. Oliver, "Adaptive visual information gathering for autonomous exploration of underwater environments," *IEEE Access*, vol. 9, pp. 136 487–136 506, 2021.

[12] C. Xiong, D. Chen, D. Lu, Z. Zeng, and L. Lian, "Path planning of multiple autonomous marine vehicles for adaptive sampling using voronoi-based ant colony optimization," *Robotics and Autonomous Systems*, vol. 115, pp. 90–103, 2019.

[13] H. Rovina, T. Salam, Y. Kantaros, and M. A. Hsieh, "Asynchronous adaptive sampling and reduced-order modeling of dynamic processes by robot teams via intermittently connected networks," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 4798–4805.

[14] D. A. Paley and A. Wolek, "Mobile sensor networks and control: Adaptive sampling of spatiotemporal processes," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 3, pp. 91–114, 2020.

[15] L. Zuo, Y. Shi, and W. Yan, "Dynamic coverage control in a time-varying environment using bayesian prediction," *IEEE transactions on cybernetics*, vol. 49, no. 1, pp. 354–362, 2017.

[16] M. Santos, U. Madhushani, A. Benevento, and N. E. Leonard, "Multi-robot learning and coverage of unknown spatial fields," in *2021 International Symposium on Multi-Robot and Multi-Agent Systems (MRS)*. IEEE, 2021, pp. 137–145.

[17] A. Benevento, M. Santos, G. Notarstefano, K. Paynabar, M. Bloch, and M. Egerstedt, "Multi-robot coordination for estimation and coverage of unknown spatial fields," in *2020 ieee international conference on robotics and automation (icra)*. IEEE, 2020, pp. 7740–7746.

[18] K. Nakamura, M. Santos, and N. E. Leonard, "Decentralized learning with limited communications for multi-robot coverage of unknown spatial fields," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2022, pp. 9980–9986.

[19] W. Luo, C. Nam, G. Kantor, and K. Sycara, "Distributed environmental modeling and adaptive sampling for multi-robot sensor coverage," in *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems*, 2019, pp. 1488–1496.

[20] A. Viseras, T. Wiedemann, C. Manss, L. Magel, J. Mueller, D. Shutin, and L. Merino, "Decentralized multi-agent exploration with online-learning of gaussian processes," in *International Conference on Robotics and Automation (ICRA)*. IEEE, 2016, pp. 4222–4229.

[21] D. Jang, J. Yoo, C. Y. Son, D. Kim, and H. J. Kim, "Multi-robot active sensing and environmental model learning with distributed gaussian process," *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 5905–5912, 2020.

[22] L. Wei, A. McDonald, and V. Srivastava, "Multi-robot gaussian process estimation and coverage: Deterministic sequencing algorithm and regret analysis," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 9080–9085.

[23] M. Schwager, M. P. Vitus, D. Rus, and C. J. Tomlin, "Robust adaptive coverage for robotic sensor networks," in *Robotics Research*. Springer, 2017, pp. 437–454.

[24] M. Jadaliha and J. Choi, "Environmental monitoring using autonomous aquatic robots: Sampling algorithms and experiments," *IEEE Trans. on Control Systems Technology*, vol. 21, no. 3, pp. 899–905, 2012.

[25] A. Okabe, *Spatial Tessellations*. John Wiley & Sons, Ltd, 2017, pp. 1–11. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/9781118786352.wbieg0601

[26] M. Schwager, J. McLurkin, and D. Rus, "Distributed coverage control with sensory feedback for networked robots." in *robotics: science and systems*, 2006, pp. 49–56.

[27] E. Schulz, M. Speekenbrink, and A. Krause, "A tutorial on gaussian process regression: Modelling, exploring, and exploiting functions," *Journal of Mathematical Psychology*, vol. 85, pp. 1–16, 2018.

[28] M. Liu, G. Chowdhary, B. C. Da Silva, S.-Y. Liu, and J. P. How, "Gaussian processes for learning and control: A tutorial with examples," *IEEE Control Systems Mag.*, vol. 38, no. 5, pp. 53–86, 2018.

[29] C. Park and J. Z. Huang, "Efficient computation of gaussian process regression for large spatial data sets by patching local gaussian processes," *Journal of Machine Learning Research-JMLR*, 2016.

[30] K. Jakkala, "Deep gaussian processes: A survey," *arXiv preprint arXiv:2106.12135*, 2021.

[31] R. Soukieh, I. Shames, and B. Fidan, "Obstacle avoidance of non-holonomic unicycle robots based on fluid mechanical modeling," in *Proceedings of the European Control Conference*, 2009.

[32] D. Lee, A. Franchi, H. Son, C. Ha, H. Bulthoff, and P. Robuffo Giordano, "Semiautonomous haptic teleoperation control architecture of multiple unmanned aerial vehicles," *IEEE/ASME Transactions on Mechatronics*, vol. 18, no. 4, pp. 1334–1345, Aug 2013.