

Adaptive sampling with an autonomous underwater vehicle in static marine environments

Paul Stankiewicz¹  | Yew T. Tan²  | Marin Kobilarov¹

¹Department of Mechanical Engineering,
Johns Hopkins University, Baltimore,
Maryland, USA

²GE Research, Niskayuna, New York, USA

Correspondence

Paul Stankiewicz, Department of Mechanical
Engineering, Johns Hopkins University,
Baltimore, MD 21218, USA.

Email: pstanki2@jhu.edu

Funding information

U.S. Department of Agriculture,
Grant/Award Number: 2017-67007-26154;
National Science Foundation,
Grant/Award Number: 1302360; NSF,
Grant/Award Number: #1302360; USDA,
Grant/Award Number: #2017-67007-26154

Abstract

This paper explores the use of autonomous underwater vehicles (AUVs) equipped with sensors to construct water quality models to aid in the assessment of important environmental hazards, for instance related to point-source pollutants or localized hypoxic regions. Our focus is on problems requiring the autonomous discovery and dense sampling of critical areas of interest in real-time, for which standard (e.g., grid-based) strategies are not practical due to AUV power and computing constraints that limit mission duration. To this end, we consider adaptive sampling strategies on Gaussian process (GP) stochastic models of the measured scalar field to focus sampling on the most promising and informative regions. Specifically, this study employs the GP upper confidence bound as the optimization criteria to adaptively plan sampling paths that balance a trade-off between exploration and exploitation. Two informative path planning algorithms based on (i) branch-and-bound techniques and (ii) cross-entropy optimization are presented for choosing future sampling locations while considering the motion constraints of the sampling platform. The effectiveness of the proposed methods are explored in simulated scalar fields for identifying multiple regions of interest within a three-dimensional environment. Field experiments with an AUV using both virtual measurements on a known scalar field and in situ dissolved oxygen measurements for studying hypoxic zones validate the approach's capability to quickly explore the given area, and then subsequently increase the sampling density around regions of interest without sacrificing model fidelity of the full sampling area.

KEYWORDS

environmental monitoring, motion planning, underwater robotics

1 | INTRODUCTION

1.1 | Motivation

One of the most practical and influential applications of the recent advances in autonomous systems is in marine environmental monitoring. For a given measurement of interest, that is, salinity, dissolved oxygen, and so forth, a natural goal is to use collected data to create an environmental model for further study. These models

can be used to localize areas that contain either high measurement variability or extreme values to help authorities identify potential abnormalities and address the problem.

As a specific use case, consider the study of hypoxia in a large estuary such as the Chesapeake Bay (Figure 1), that in last few decades has become increasingly sensitive to pollution from agricultural and wastewater discharge, urban runoffs, and atmospheric deposition. To establish effective treatment and prevention practices, it is necessary to monitor and assess the complex

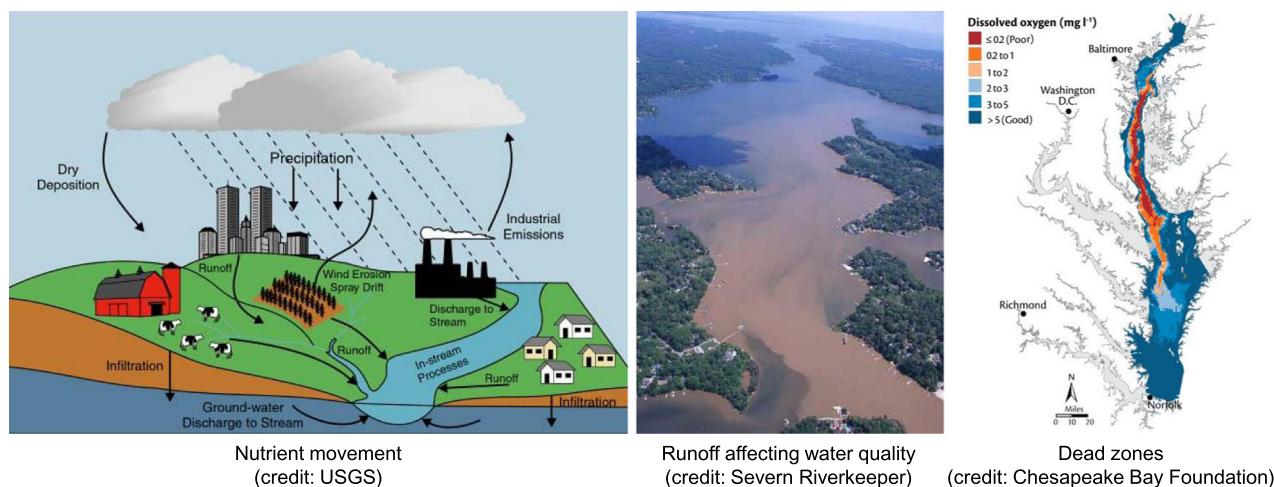


FIGURE 1 Hypoxic dead zone formation in the Chesapeake Bay [Color figure can be viewed at wileyonlinelibrary.com]

dynamic processes affecting water quality. The standard approach is to supplement networks of fixed sampling stations with manual sampling from ships, and perform data integration and analysis in the lab. This approach has several limitations that could hinder water management: short time scale events such as rapid rain-storm runoffs cannot be captured at sufficient resolution; diffusive pollution sources are difficult to track from sparse samples; critical locations such as shallow and obstacle-cluttered areas as well as creeks are difficult to access by ship and require additional infrastructure, integration, and manpower. Additionally, simply interpolating between stations has the potential to miss critical regions of interest (ROIs), that is, regions that exhibit extreme sensory measurements and high spatially correlated variability when compared to the surrounding area.

More recently, autonomous underwater vehicles (AUVs) and autonomous surface vehicles (ASVs) have begun to be used in these scenarios; however, they are typically constrained to area coverage objectives and thus follow predefined sampling trajectories such as a lawnmower pattern (Galceran & Carreras, 2013). These strategies are highly inefficient when the overall goal is to detect and localize ROIs within a larger area. Marine environments exacerbate this problem (particularly for AUVs) because it is difficult for an operator to update the vehicle trajectory due to communication limitations. The result is that the data collected by the system is still only available to provide insights in a postprocessed fashion.

Rather, autonomous systems now have the capability to adapt their trajectories based on in situ data processing to focus sampling efforts on higher-level goals such as localizing ROIs. Recent researchers have made use of the spatial correlation structure of these ROIs to design adaptive sampling strategies for robotic environmental monitoring (Low et al., 2009; Mysorewala et al., 2009). In these informative path planning strategies, the sensor information observed along the vehicle trajectory is assimilated into an environmental model, which is then used to guide future sampling

locations. The ability to adapt the sampling paths using in situ sensor information has shown to be a more effective method for sensing and localizing ROIs in a given environment.

1.2 | Contributions

This study aims to overcome some of the challenges in marine monitoring through adaptive sampling with robotic vehicles guided by real-time information-seeking algorithms, which autonomously decide sampling locations to optimize a given task such as pollution source localization. Specifically, we focus on single-vehicle planning for collecting a dense set of measurements within highly localized ROIs, for example, channels with varying bathymetry in a large estuary, while also fully exploring the environment workspace. This study extends upon a previously developed system (Tan et al., 2018), which was limited to simpler two dimensional (2D) environments and applied to surface vehicles, and includes the following contributions.

1.2.1 | Receding-horizon informative planning

An adaptive sampling framework is proposed in conjunction with two informative path planning strategies for generating high-information trajectories. The first informative path planner is based on branch-and-bound techniques and the second is based on cross-entropy optimization. Both of these planning techniques explicitly consider the motion constraints of an underactuated vehicle operating within a constrained environment. This is in contrast to much of the sampling research in the marine domain (Das et al., 2013; Ma et al., 2018; Smith et al., 2011) which focuses on large-scale, low-resolution environmental monitoring where motion constraints need not be prioritized.

1.2.2 | Constrained 3D environments

Applications of AUV adaptive sampling are explored where the complete 3D environmental area is considered for collecting samples. To date, most studies have focused on environmental sampling in 2D environments. While a handful of studies collect data in 3D, the depth component is typically only controlled via predefined yo-yo patterns (Das et al., 2015; Smith et al., 2011). These approaches simply generate and utilize 2D models of the environment for planning as opposed to considering the added depth component. The methodology presented here allows for dense data collection near the seafloor, which would be difficult for underwater gliders or surface vessels. Further, we also consider environmental constraints, such as bathymetry and bottom structure, and operational constraints, such as limited communication and navigation capabilities, that require the AUV to periodically surface.

1.2.3 | Simulations and real-world deployments

Monte Carlo simulation experiments are used to measure the performance of the proposed methods over a large set of auto-generated 3D environments. Additionally, several field experiments with an AUV show the capability of the proposed algorithms to be implemented on operational platforms for real-time identification of ROIs in a 3D sampling environment.

1.3 | Limitations

While the application and use of AUV sampling platforms allows increased data resolution in constrained environments, these systems typically have limited on-board computational resources. This constraint results in some limitations of the proposed approach that must be acknowledged. To facilitate real-time planning and model-building for a 3D environment, trajectory generation is performed in a receding horizon fashion. This approach, while precluding a globally optimal solution over the full mission profile, is able to generate high-information trajectories in a local sense up to the planning horizon of the sampling platform. The optimality of these solutions is then also dependent on approximations of both the platform's dynamics and reachable space over the local planning horizon.

An additional assumption within this study is that the environmental field is static with respect to time. In other words, we assume that the temporal dynamics of the environment are much slower than the duration of the sampling mission (typical AUV deployments within the application of interest only last on the order of hours). Generalization of the proposed framework for long-term monitoring that considers a dynamic environment is left to future work, perhaps leveraging concepts from the nondimensional approach of (Leonard et al., 2007). In spite of these limitations, the proposed approach

provides significantly improved data collection when compared to traditional methods.

The remainder of the paper is organized as follows: Section 2 presents a review of related work in the areas of adaptive sampling with particular focus on marine environments. Section 3 introduces the problem formulation, followed by Section 4 where the methodology for adaptive trajectory planning is presented. Results from both simulation studies and field experiments with an AUV are showcased in Sections 5 and 6, respectively. Finally, conclusions are summarized in Section 7.

2 | RELATED WORK

The use of robotics for environmental monitoring has become a significant research effort, with a survey of applications provided in Dunbabin and Marques (2012). For example, applications have been studied in diverse domains including volcanic environments (Muscato et al., 2012), data collection in severe weather (Frew et al., 2012), forest fire mapping (Mysorewala et al., 2009), and acoustic predictions in ocean fields (Wang et al., 2009). A typical approach to this application is to use sensor information to build a model of the environment. Predictions from this model and their associated uncertainties are then taken into account when planning sampling paths. When performing sensing in an unknown environment, balancing a trade-off between exploration and exploitation (Powell, 2007) is crucial. Exploration refers to collecting samples that fully survey the environmental area, whereas exploitation refers to collecting samples in areas that contain the desired measurements, that is, in ROIs. Thus, care must be taken to design algorithms that balance these two competing objectives when selecting sampling paths.

A commonly used model for representing environmental phenomena is the Gaussian process (GP) (Das et al., 2015; Gotovos et al., 2013; Ouyang et al., 2014). A GP is a nonparametric regression model that allows estimation and prediction of the spatial correlation structure of an environmental field. Sensor data, as it is acquired, is used to learn the GP by estimating the model's hyperparameters. In turn, the mean and variance predictions of the model are used to guide future sampling locations. This iterative, sequential model update and prediction is well-suited for adaptive sampling missions where sensing platforms must repeatedly select the best sampling path. Such a problem can be formulated as Bayesian optimization by using the GP to maintain a probabilistic belief about the environmental distribution while using its predictions to design an appropriate acquisition function (Bull, 2011; Kawaguchi et al., 2015; Wang et al., 2014). For example, the work in Marchant and Ramos (2014) uses a multilayered Bayesian optimization approach to generate maximum information sampling paths, a field known as informative path planning.

The problem of informative path planning is known to be NP-hard and there are many algorithms aimed at efficiently finding

paths that maximize information gain subject to budget constraints (i.e., time, distance, etc.). Randomized approaches were explored in Hollinger and Sukhatme (2014) by adapting sampling-based motion planning such as RRT* and PRM* (Karaman & Frazzoli, 2011) to include expected information gain in the cost of each path segment. Extensions to reduce planning time for real-time operation can be achieved by restricting the space from which paths are sampled (Arora & Scherer, 2017). To avoid discretization of the environment, Hitz et al. (2017) use evolutionary techniques to optimize a parameterized B-spline in continuous space. Further, Reid et al. (2020) explore hierarchical informative planning with applications to sampling using reconfigurable systems such as a planetary rover. Additional approaches to informative path planning based on hypothesis testing (Lim et al., 2016), fast marching methods (Lawrance et al., 2017), and orienteering (Bottarelli et al., 2019) have also been proposed.

The idea of model-based informative planning has been applied extensively in the marine domain, a recent example of which uses partially observable Markov decision processes for coral reef localization (Flaspohler et al., 2019). Adaptive sampling in marine environments is a natural application due to operational constraints such as limited communication bandwidth. In these instances, it is preferred to design planning methods that allow the system to adapt without human interaction. A series of studies by Manjanna, Van Hoof, et al. (2018) applied adaptive sampling to surface vessels focusing on policy search, multiscale resolution (Manjanna & Dudek, 2017), and integration with static sensor nodes (Manjanna et al., 2017). A second series of studies by Das et al. provide sampling strategies for environmental monitoring off the California coast. The approach in Das

et al. (2012) uses a combination of Lagrangian drifters and AUVs to track and sample algal blooms, where drifters are used to identify a region of interest that is then sent to the AUV for further sampling. Subsequent work in Das et al. (2013) proposes an approach that uses observable environmental features (e.g., salinity or temperature) to inform where unobservable features (e.g., organism abundance) should be collected to reduce uncertainty in a combined environmental model. In addition, Das et al. (2015) employ prior sampling campaigns to inform the collection of a very limited number of plankton samples.

An outstanding challenge in the area of marine monitoring is measurement collection over a sustained period of time to understand the temporal dynamics of the environment. Initial work in this area by Leonard et al. (2010) used coordinated control of a fleet of underwater gliders to intelligently collect samples over several months, while Smith et al. (2011) improved long-term monitoring for gliders by developing planning techniques that considered both information gain and regions of undesirable ocean currents. Further work by Ma et al. (2016, 2018) specifically includes dynamics of the environment to capture temporal effects when planning sampling paths.

Teaming approaches as conceptualized in Figure 2 have also gained popularity due to the challenges of long-term monitoring in large areas (Das et al., 2012; Li et al., 2016; Low et al., 2009; Mukhopadhyay et al., 2014). A key advantage of robot teaming is that the strengths of each individual vehicle can be exploited to improve the overall system performance. For example, the mobility advantage of aerial vehicles can be used to collect sparse samples over large distances that identify general areas for further examination. Subsequently, an AUV could be deployed to

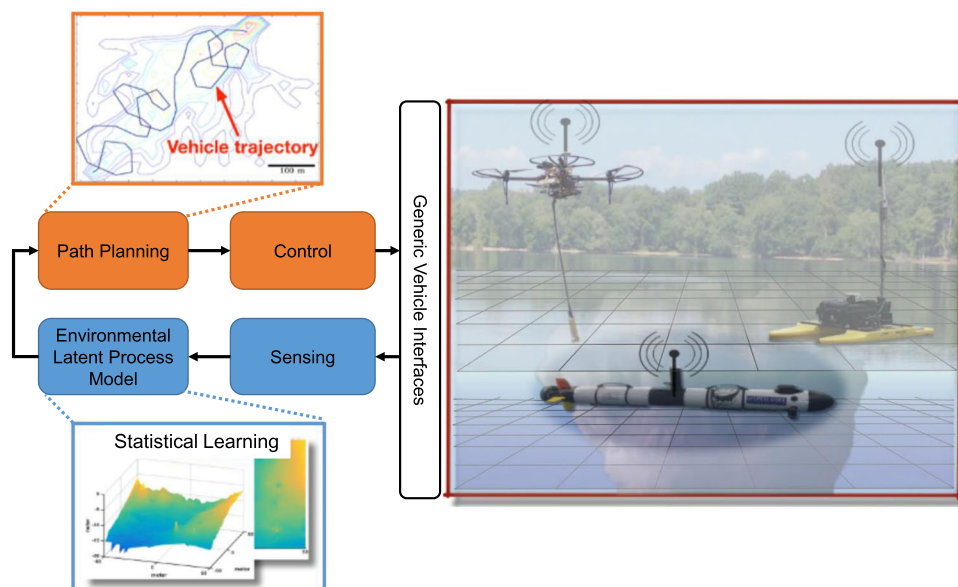


FIGURE 2 Concept of environmental sampling using a heterogeneous robot team. High-mobility vehicles such as AUVs and ASVs collect sparse samples over a large area, whereas AUVs provide focused samples in a region of interest (this paper focuses on the latter). ASV, autonomous surface vehicle; AUV, autonomous surface vehicle [Color figure can be viewed at [wileyonlinelibrary.com](https://onlinelibrary.wiley.com)]

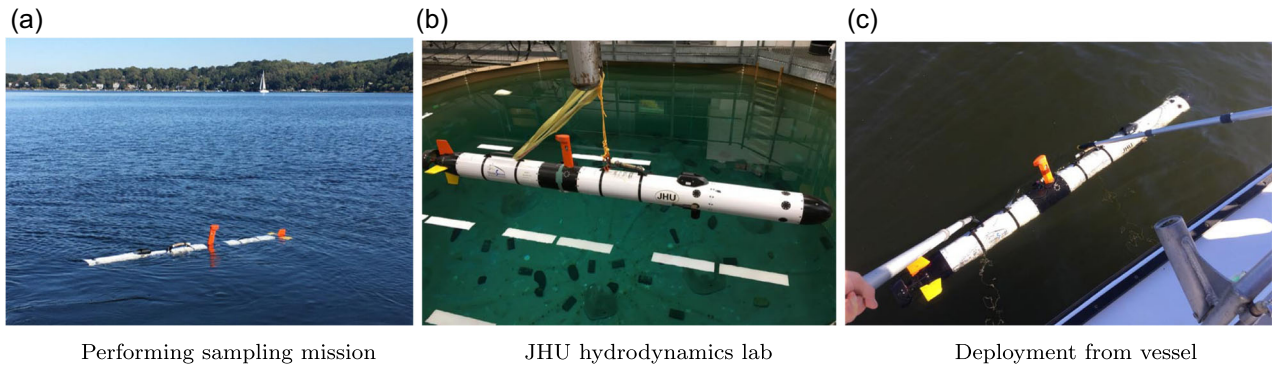


FIGURE 3 Johns Hopkins University L3 OceanServer Iver3 autonomous underwater vehicle [Color figure can be viewed at wileyonlinelibrary.com]

these areas to collect dense measurements on a smaller scale near the seafloor to localize ROIs. Additionally, surface vessels can operate as a charging station and communication hub between vehicles. Manjanna, Li, et al. (2018) employ this concept by using a team of surface vessels to coordinate both exploring the environment and collecting physical water samples. Similarly, the work in Munafò et al. (2011) uses AUV teams to minimize the uncertainty in the estimated GP model while staying within communication range, whereas Ouyang et al. (2014) examines the performance of decentralized coordination as a function of the number of agents in the team. Alternative planning for AUV teams using RRT* and a selective basis Kalman filter is proposed in Cui et al. (2016).

Of particular importance in adaptive sampling methods is the choice of the acquisition function for evaluating candidate sample locations. Common approaches for this optimization criteria are based on entropy (Low et al., 2009), mutual information (Krause et al., 2008), or the combination of the predictive mean and information gain (Liu et al., 2015). Additionally, the work in Cao et al. (2013) compares the effectiveness of entropy and mutual information criteria. In this study, we adopt the GP upper confidence bound (GP-UCB) algorithm (Srinivas et al., 2009) as the criteria for choosing future sampling locations. The GP-UCB algorithm calculates the utility of future sampling locations according to a weighted combination of both their predictive mean and variance, thus allowing control over the exploration-exploitation trade-off at different stages of the sampling mission. The following sections introduce the problem formulation and detail both the GP model estimation as well as the formulation of the GP-UCB as a sequential stochastic optimization problem.

3 | PROBLEM SETUP

We now formalize the informative path planning problem. Consider a vehicle trajectory evolving according to an ordinary differential equation, $\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t))$, where the state and control at time

$t > 0$ are denoted by $\mathbf{x}(t) \in \mathcal{X}$ and $\mathbf{u}(t) \in \mathcal{U}$, respectively, with \mathcal{X} and \mathcal{U} denoting the state and control spaces. Letting t_f be the final time of the trajectory (i.e., the allotted mission time), a given control curve $\mathbf{u}: [0, t_f] \rightarrow \mathcal{U}$ determines a unique state trajectory $\mathbf{x}: [0, t_f] \rightarrow \mathcal{X}$ by evolving the dynamics from an initial state $\mathbf{x}_0 \in \mathcal{X}$. We would like to compute the optimal controls $\mathbf{u}^*(\cdot)$ that maximize a given acquisition function $L: \mathcal{U} \times \mathcal{X} \rightarrow \mathbb{R}$ while driving the system from its initial state \mathbf{x}_0 until the final time t_f , that is,

$$\begin{aligned} & \underset{\mathbf{u}}{\text{maximize}} && \int_0^{t_f} L(\mathbf{x}(t), \mathbf{u}(t)) dt \\ & \text{subject to} && \dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t)), \\ & && \mathbf{x}(0) = \mathbf{x}_0, \\ & && g(\mathbf{x}(t), \mathbf{u}(t)) \leq 0, \end{aligned} \quad (1)$$

for all $t \in [0, t_f]$. Here, $g(\mathbf{x}(t), \mathbf{u}(t)) \leq 0$ are constraints on the state and input arising from actuator bounds and obstacles in the environment. Next, we define the specific vehicle dynamics, acquisition function over the environment, and convert the problem of Equation (1) into a numerically convenient form that serves as the basis for the proposed algorithms.

3.1 | Vehicle dynamic model

The robotic sampling platform employed in this problem formulation is a torpedo-style AUV. Specifically, the Johns Hopkins University (JHU) L3 OceanServer Iver3 (Figure 3) is used as the testbed. This AUV is actuated by a rear propeller for thrust, a pair of rudder fins for yaw control, and a pair of diving planes for pitch control. Because this system is underactuated, it is particularly important to consider the motion constraints of the vehicle during sample collection.

The hydrodynamic model used for this vehicle is a simplified version of the equations of motion provided in Gertler and Hagen (1967). The 4-degree of freedom (DOF) model presented here captures realistic vehicle motion while minimizing the number of required model coefficients. Thus, sufficient fidelity is captured for

simulating high-level motion planning while low-level control such as roll stability is assumed to be controlled by the vehicle's on-board autopilot. The state $\mathbf{x} = (p_x, p_y, p_z, \theta, \psi, u, w, q, r)$ includes the position $\mathbf{p} = (p_x, p_y, p_z) \in \mathbb{R}^3$, pitch θ , yaw ψ , surge velocity u , heave velocity w , pitch rate q , and yaw rate r . The control inputs $\mathbf{u} = (\delta_u, \delta_q, \delta_r)$ define the normalized thrust input, normalized pitch rate input, and normalized yaw rate input, respectively. These control inputs are scaled and constrained to produce high-level motion characteristics at steady-state. The normalized thrust input $\delta_u \in [0, 1]$ represents the range between zero and maximum propeller speed, the normalized pitch rate input $\delta_q \in [-1, 1]$ represents the range between maximal deflections of the vehicle diving planes (and consequently the maximal torques generated due to these deflections), and the normalized yaw rate input $\delta_r \in [-1, 1]$ represents the range between maximal deflections of the vehicle rudder fins. Assuming negligible effects in the sway and roll dynamics, the equations of motion $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u})$ are given as the following:

$$\dot{p}_x = u \cos \psi \cos \theta + w \cos \psi \sin \theta, \quad (2a)$$

$$\dot{p}_y = u \sin \psi \cos \theta + w \sin \psi \sin \theta, \quad (2b)$$

$$\dot{p}_z = w \cos \theta - u \sin \theta, \quad (2c)$$

$$\dot{\theta} = q, \quad (2d)$$

$$\dot{\psi} = r / \cos \theta, \quad (2e)$$

$$\dot{u} = \bar{X}_{uu} u^2 + \bar{k} \delta_u, \quad (2f)$$

$$\dot{w} = \bar{Z}_{w|w|} w |w| + \bar{WB} \cos \theta, \quad (2g)$$

$$\dot{q} = \bar{M}_{uq} u q + \bar{M}_q q - \bar{Bz}_B \sin \theta + \bar{b} u^2 \delta_q, \quad (2h)$$

$$\dot{r} = \bar{N}_{ur} u r + \bar{c} u^2 \delta_r, \quad (2i)$$

where (\cdot) indicates that the hydrodynamic coefficients have been made independent of the vehicle's mass and inertia. Further, \bar{WB} represents an out-of-ballast term based on the vehicle's weight and buoyancy ratio and \bar{Bz}_B represents a buoyancy term that accounts for the center of buoyancy vertical offset from the center of gravity. Values for all coefficients for the JHU Iver3 were determined through experimental data and are listed in Table 1.

3.2 | Finite-dimensional optimization through vehicle motion parameterization

Next, we convert the infinite-dimensional trajectory optimization problem of Equation (1) into a finite-dimensional optimization, by

parameterizing vehicle trajectories as sequences of motion primitives. Let π represent a particular trajectory, with the space of all trajectories originating at point \mathbf{x}_0 and satisfying the dynamics of the ODE denoted by

$$\mathcal{P} = \{\pi: t \in [0, t_f] \rightarrow \{\mathbf{u}(t), \mathbf{x}(t)\} | \dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t)), \mathbf{x}(0) = \mathbf{x}_0, t_f > 0\}. \quad (3)$$

Trajectories are parameterized with a function $\varphi: \mathcal{Z} \rightarrow \mathcal{P}$ according to Kobilarov (2012):

$$\pi = \varphi(\mathbf{z}) \equiv \varphi_z, \quad (4)$$

where $\mathbf{z} \in \mathcal{Z}$ is the parameter space. Let $\Gamma: \mathcal{U} \times \mathcal{X} \rightarrow \mathcal{X}$ project onto the state component, that is, $\Gamma(\mathbf{u}, \mathbf{x}) = \mathbf{x}$. The constrained parameter space $\mathcal{Z}_{con} \subset \mathcal{Z}$ is the set of parameters satisfying the boundary conditions and constraints, defined by

$$\mathcal{Z}_{con} = \{\mathbf{z} \in \mathcal{Z} | g(\Gamma(\varphi_z(t))) \leq 0, t \in [0, t_f]\}. \quad (5)$$

Define the reward γ and the reward function $J: \mathcal{Z} \rightarrow \mathbb{R}$ according to:

$$\gamma = J(\mathbf{z}) = \int_0^{t_f} L(\varphi_z(t)) dt. \quad (6)$$

Equation (1) can now be solved approximately by finding $\{\mathbf{x}^*, \mathbf{u}^*\} = \varphi(\mathbf{z}^*)$ such that

$$\mathbf{z}^* = \underset{\mathbf{z} \in \mathcal{Z}_{con}}{\operatorname{argmax}} J(\mathbf{z}). \quad (7)$$

Similarly, we let γ^* be the reward function optimum over the space \mathcal{Z}_{con} :

$$\gamma^* = \max_{\mathbf{z} \in \mathcal{Z}_{con}} J(\mathbf{z}). \quad (8)$$

For our specific setting, a trajectory is parameterized using a sequence of H motion primitives, where a *primitive* is defined as a section of a helix with constant horizontal forward velocity v_x , depth rate v_z , and turn rate ω_z . More formally, the trajectory parameter is $\mathbf{z} = \{(v_x, v_z, \omega_z)\}_{j=1}^H$. During a time interval, $t \in [t_j, t_{j+1}]$, the trajectory state $\mathbf{x}(t) = \varphi_z(t)$ takes the form

$$\theta(t) = \arctan(v_z/v_x), \quad (9a)$$

$$\psi(t) = \psi(t_j) + \delta t_j \omega_z, \quad (9b)$$

$$u(t) = \sqrt{v_x^2 + v_z^2}, \quad (9c)$$

$$w(t) = (v_z + u(t) \sin \theta(t)) / \cos \theta(t), \quad (9d)$$

$$q(t) = 0, \quad (9e)$$

TABLE 1 Hydrodynamic model coefficients for the Johns Hopkins University Iver3 autonomous underwater vehicle

Coefficient	\bar{X}_{uu}	\bar{k}	$\bar{Z}_{w w }$	\bar{WB}	\bar{M}_{uq}	\bar{M}_q	\bar{Bz}_B	\bar{b}	\bar{N}_{ur}	\bar{c}
Value	-0.179	0.519	0.098	-2.452	-3.519	-0.748	8.947	3.096	-0.441	0.065
Units	1/m	m/s ²	1/m	m/s ²	1/m	1/s	1/s ²	1/m ²	1/m	1/m ²

$$r(t) = \omega_z \cos \theta(t), \quad (9f)$$

$$p_x(t) = p_x(t_j) + \delta t_j (u(t) \cos \psi(t) \cos \theta(t) + w(t) \cos \psi(t) \sin \theta(t)), \quad (9g)$$

$$p_y(t) = p_y(t_j) + \delta t_j (u(t) \sin \psi(t) \cos \theta(t) + w(t) \sin \psi(t) \sin \theta(t)), \quad (9h)$$

$$p_z(t) = \max\{p_z(t_j) + \delta t_j v_z, 0\}, \quad (9i)$$

where $\delta t_j = t - t_j$. In other words, we have established a mapping between the simple and geometrically intuitive parameters \mathbf{z} and trajectory $\mathbf{x}(\cdot)$ in the full state space \mathcal{X} .

3.3 | Environmental modeling and prediction as a GP

The field of interest is defined as a continuous function $h: \mathcal{W} \rightarrow \mathbb{R}$ that maps locations $\mathbf{p} \in \mathcal{W}$ in the environment workspace of interest $\mathcal{W} \subset \mathbb{R}^3$ to the scalar environmental phenomena under study. Each measurement of this phenomena, denoted by $y \in \mathbb{R}$ is assumed to have some spatial regularity (e.g., measurements at nearby points have a physically meaningful correlation) and in addition is assumed to be corrupted by Gaussian noise: $y = h(\mathbf{p}) + \mathcal{N}(0, \sigma(\mathbf{p})^2)$. These assumptions allow the environment to be modeled as a GP: a collection of dependent random variables, one for each $\mathbf{p} \in \mathcal{W}$, every finite subset of which is multivariate Gaussian-distributed (Rasmussen & Williams, 2006). A GP can be fully specified by its mean function $\mu(\mathbf{p}) = \mathbb{E}[h(\mathbf{p})]$ and covariance function $\kappa(\mathbf{p}_i, \mathbf{p}_j) = \text{cov}(\mathbf{p}_i, \mathbf{p}_j | \theta)$ for all $\mathbf{p}_i, \mathbf{p}_j \in \mathcal{W}$, where θ parameterizes the covariance function that models the spatial correlation of the environmental phenomena. A popular form for the covariance function is the squared exponential function (Rasmussen & Williams, 2006):

$$\kappa(\mathbf{p}_i, \mathbf{p}_j) = \sigma_f^2 \exp\left(-\frac{1}{2} \sum_{k=1}^d \frac{(\mathbf{p}_{i,k} - \mathbf{p}_{j,k})^2}{l_k^2}\right) + \delta(\mathbf{p}_i, \mathbf{p}_j) \sigma_w^2. \quad (10)$$

The squared exponential covariance function has a hyperparameter vector $\theta = [l_1, \dots, l_d, \sigma_f, \sigma_w]^T$, where σ_f and σ_w are the variance of the signal and noise, respectively. The parameters l_k are the length-scales in each dimension, which determine the level of correlation between measurements. $\delta(\mathbf{p}_i, \mathbf{p}_j)$ is the Kronecker delta function, which is one if and only if $\mathbf{p}_i = \mathbf{p}_j$ and zero otherwise.

Let $P = \{\mathbf{p}_1, \dots, \mathbf{p}_N\} \subset \mathcal{W}$ be a set of N observed locations with recorded measurements $Y = \{y_1, \dots, y_N\}$. The GP representation allows us to predict measurements Y_* at a finite set of unobserved locations of interest $P_* \subset \mathcal{W}$ by defining the joint distribution of measurements at observed and unobserved locations according to Rasmussen and Williams (2006):

$$\begin{bmatrix} Y \\ Y_* \end{bmatrix} \sim \left(0, \begin{bmatrix} K + \sigma_w^2 I & K_*^T \\ K_* & K_{**} \end{bmatrix}\right), \quad (11)$$

where $K_* = K(P_*, P)$ contains the pairwise covariances between unobserved and observed points, with an analogous definition of

$K = K(P, P)$ and $K_{**} = K(P_*, P_*)$. Conditioning on the known data (Rasmussen & Williams, 2006), the unobserved measurements take the form $Y_* \sim (\hat{Y}_*, \hat{\Sigma}_*)$ with

$$\hat{Y}_* = K_*(K_y)^{-1}Y, \quad (12)$$

$$\hat{\Sigma}_* = K_{**} - K_*(K_y)^{-1}K_*^T, \quad (13)$$

where $K_y = K + \sigma_w^2 I$.

3.3.1 | Hyperparameter adaptation

The GP model accuracy is highly dependent on the hyperparameter vector θ that is used on the covariance function. When operating in an unknown environment, it is unlikely that the correct hyperparameters for a given environmental distribution will be known a priori. Thus, a common approach is to learn the hyperparameter vector over time as measurements are collected. In this study, we employ the maximization of the log of the marginal-likelihood (Murphy, 2012) to obtain the best estimate of the hyperparameter vector θ^* after the collection of n samples:

$$\theta^* = \underset{\theta}{\operatorname{argmax}} \log p(Y|P), \quad (14)$$

where

$$\log p(Y|P) = -\frac{1}{2} Y^T K_y^{-1} Y - \frac{1}{2} \log |K_y| - \frac{n}{2} \log 2\pi,$$

with the optimization performed using the gradient

$$\frac{\partial}{\partial \theta_i} \log p(Y|P) = \frac{1}{2} \operatorname{tr} \left(((K^{-1}Y)(K^{-1}Y)^T - K_y^{-1}) \frac{\partial K_y}{\partial \theta_i} \right).$$

The hyperparameter vector is initialized based on expert knowledge of the environmental area and is then periodically re-estimated every l samples (this study applies $l = 1$).

3.4 | Acquisition function

For a robotic system that collects environmental data sequentially, it is important to balance data collection between exploration and exploitation, that is, sampling at locations with high uncertainty against locations with high predicted reward. Concentrating the sensing efforts within a specific area prematurely may miss ROIs in other areas of the environment, whereas aimlessly exploring the environment would forgo opportunities to collect important samples around the ROIs. The goal is to develop a policy that determines the optimal sequence of future sampling locations, taking into account both the GP model mean (predicted reward) through Equation (12) and the GP model variance (predicted uncertainty) through Equation (13), both of which are calculated using the current best estimate of the hyperparameter vector.

In this study, we adopt the GP upper confidence bound (GP-UCB) (Das et al., 2015; Srinivas et al., 2009) criteria for guiding sample selection. While entropy-based methods tend to reduce uncertainty over the full sampling area, UCB-based acquisition functions have been shown to provide better resolution of ROIs (Marchant & Ramos, 2014). From a GP optimization perspective, the GP-UCB algorithm is designed to minimize the cumulative regret over the collection of N samples, that is, the loss in reward from not selecting the optimal set of N samples. While sensor data can typically be collected at high rates, it becomes computationally infeasible to train the GP model on the entire data set. Downselecting this data to the optimal N samples make the training process tractable. This strategy applies the principle of optimism in the face of uncertainty where the selection of a sample position $\mathbf{p}_n \in \mathcal{W}$ at sampling step n is evaluated according to

$$L(\mathbf{x}_n, \mathbf{u}_n) = \underbrace{\mu_{n-1}(\mathbf{p}_n) + \sqrt{\beta_n} \sigma_{n-1}(\mathbf{p}_n)}_{\text{sampling reward}} - \underbrace{q(\mathbf{x}_n)}_{\text{navigation cost}}, \quad (15)$$

where $\mu = \hat{\mathbf{Y}}_*$ and $\sigma^2 = \hat{\Sigma}_*$ as described in Section 3.3.

The first component of the acquisition function is the sampling reward, which encourages the collection of informative samples. This reward depends only on the sampling locations and is based on the trade-off between potentially maximizing the GP model mean (choosing locations within ROIs characterized by maximal sensory measurements) and maximizing the GP model variance (exploring uncertain and/or unobserved areas of the environment workspace). Alternatively, if an ROI is described by minimal measurement values, as would be the case for detecting hypoxic zones, one would simply maximize the negative of the GP model mean, $-\mu$, to encourage sample collection within ROIs. The term β_n employed in Equation (15) is the mean-variance trade-off parameter at sampling step n , and is described as in Srinivas et al. (2009):

$$\beta_n = 2 \log \left(\frac{|d| n^2 \pi^2}{6\delta} \right), \quad (16)$$

where d is the dimensionality of \mathcal{W} and $\delta \in [0, 1]$ is a parameter that defines the probability that the regret bound given in Srinivas et al. (2009) is satisfied after N sampling steps, for $n = 1 \dots N$.

The second component of Equation (15) is a standard non-negative navigation cost, which heavily penalizes trajectories that exit the environment workspace or approach a forbidden state set \mathcal{O} (e.g., known obstacles or shorelines from nautical charts, the sea-floor, etc.). Here, an infinite step function is applied to undesirable states as such:

$$q(\mathbf{x}_n) = \begin{cases} \infty & \text{if } \mathbf{p}_n \notin \mathcal{W} \text{ or } \mathbf{x}_n \in \mathcal{O} \\ 0 & \text{otherwise.} \end{cases} \quad (17)$$

While this study does not consider real-time obstacle detection, extending the navigation cost in future work is possible by augmenting \mathcal{O} with detected obstacles.

4 | ADAPTIVE TRAJECTORY PLANNING

This section now proposes an algorithmic approach for adaptive environmental sampling and explores two informative path planning methods for solving the optimization of Equations (7) and (8). Algorithm 1 details the overall framework of the proposed GP adaptive sampling (GPAS) method. In this framework the GP is initialized to a known prior or, if no known prior is available, it is set to a uniform constant field with a given variance. The first measurement y_0 at the starting location \mathbf{p}_0 is then added to the model, and subsequent sampling points \mathbf{p}_n at each iteration are selected through the PATHPLANNING subroutine of Algorithm 1.

Algorithm 1 Gaussian process adaptive sampling (GPAS)

```

1: procedure GPAS (GP,  $\mathbf{x}_0$ ,  $t_f$ )
2: ▷ Collect measurement at starting location
3:  $y_0 = \text{MEASUREMENT}(\mathbf{p}_0)$ 
4: ▷ Add starting location and measurement to the GP
5:  $\text{GP}_0 = \text{GP\_INITIALIZE}(\text{GP}, \mathbf{p}_0, y_0)$ 
6: Let  $t = 0$ ;  $n = 0$ 
7: repeat
8: Set  $t = t + \delta_t$ ;  $n = n + 1$ 
9: ▷ Plan a sampling path up to planning horizon  $T$ 
10:  $\mathbf{x}^*([t, t + T]) = \text{PATHPLANNING}(\text{GP}_{n-1}, \mathbf{x}(t))$ 
11: ▷ Collect measurement at first sampling location along path
12:  $y_n = \text{MEASUREMENT}(\mathbf{p}_n^*)$ 
13: ▷ Add sampling location and measurement to the GP
14:  $\text{GP}_n = \text{GP\_ASSIMILATE}(\text{GP}_{n-1}, \mathbf{p}_n^*, y_n)$ 
15: ▷ Estimate GP hyperparameters with updated GP
16:  $\theta_n^* = \text{ESTIMATEHYPERPARAM}(\text{GP}_n)$  ▷ Equation (14)
17: until  $t = t_f$ 
18: end procedure

```

For a realistic sampling mission duration, planning a trajectory over a long horizon is often intractable. Real-time computation can be achieved, however, by calculating a suboptimal approximation of the full trajectory over a local planning horizon T . A receding horizon strategy is implemented here such that at every sampling step n (i.e., after each collected measurement), a trajectory $\mathbf{x}^*([t, t + T])$ is generated containing a sequence of H future sampling points $\{\mathbf{p}_i^*\}_{i=n}^{n+H}$, so that $T = H\delta_t$, where δ_t is a fixed sampling time step. Once sampling is performed at \mathbf{p}_n^* , the corresponding measurement y_n is assimilated into the GP model and the hyperparameters are updated. This process is repeated until the final mission time t_f . While this receding-horizon strategy is

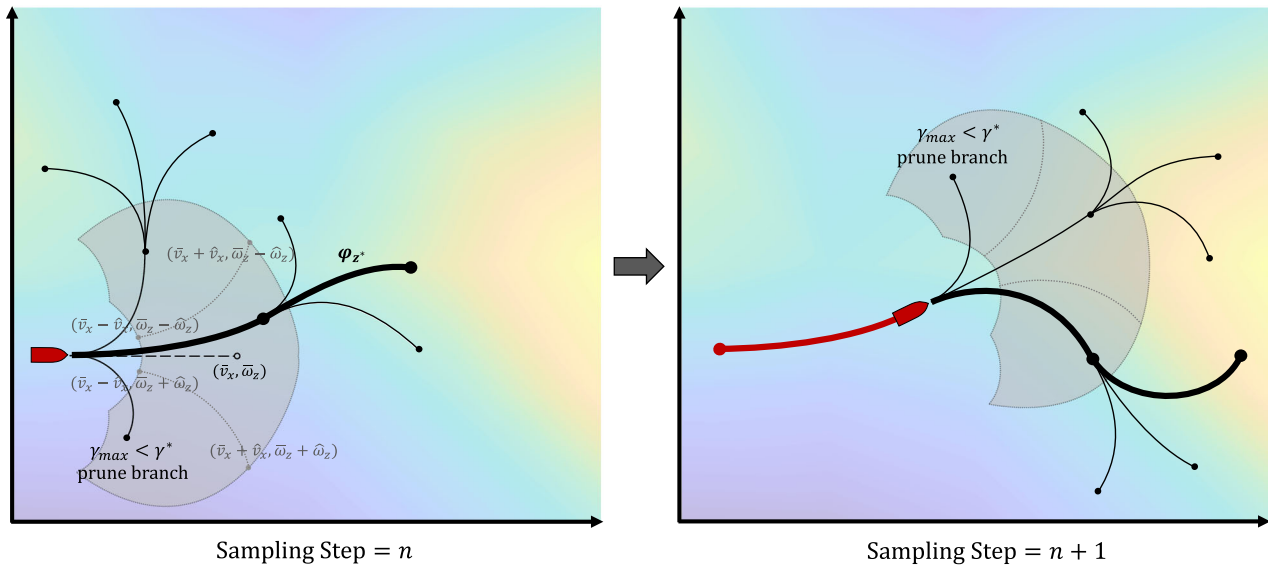


FIGURE 4 Illustrative example of the receding horizon branch-and-bound informative path planning algorithm driving towards a high-reward (yellow) region of interest along the right edge of the sampling area. This 2D example uses $H = 2$ and trajectory parameterizations of $z = (v_x, \omega_z)$ corresponding to speed and turn rate, respectively. For $v_x > 0$, the gray region represents the reachable space for one branching step over $t + \delta_t$ [Color figure can be viewed at wileyonlinelibrary.com]

not globally optimal, it permits real-time execution and in situ field estimation on-board the sampling platform, which is our main objective.

Two approaches are presented to generate the sampling trajectory in the PATHPLANNING subroutine of Algorithm 1. The first method is a modified branch-and-bound informative path planner (Section 4.1) and the second method is a cross-entropy informative path planner (Section 4.2). These two planning algorithms are presented and compared because of their respective strengths and weaknesses in different applications. The foundations of branch-and-bound techniques are well-suited for efficient informative path planning (Binney & Sukhatme, 2012; McMahon et al., 2017), particularly on low-cost platforms with limited computational resources. However, the generation of the candidate planning tree is often dependent on suboptimal heuristics. Alternatively, cross-entropy-based methods, while more computationally demanding, are commonly used in reinforcement learning frameworks due to the desirable quality that they need not make any assumptions about the underlying (potentially black box) system. Thus, both algorithms offer value during informative path planning and the utility of each is analyzed in the following sections.

It should also be noted that the scale of the environmental distributions under study requires that the motion constraints of the sensing platform be taken into consideration during planning. More specifically, the objective is to capture scalar fields that exhibit variability with a resolution on the scale of the vehicle's turning radius. In contrast to other methods (Das et al., 2013; Ma et al., 2018; Smith et al., 2011) where planning is performed

within a sampling grid of several miles, both of the methods presented here explicitly consider vehicle motion constraints with the intention of sampling highly localized ROIs in a 3D environment.

4.1 | Branch-and-bound informative path planning (BB-IPP)

The computation time of a finite-horizon planner is exponential in the length of the horizon. Thus, exhaustive search methods that evaluate the reward function for every possible sampling path are only feasible for short planning horizons. Branch-and-bound algorithms offer an efficient alternative by pruning large portions of the search tree that are unlikely to contain the optimal solution based on an upper bound on the estimated reward.

Binney (Binney & Sukhatme, 2012) explored branch and bound for informative path planning between two points on a coarsely discretized 2D sampling grid. There are three ways in which the method presented here differs from Binney and Sukhatme (2012): (i) we apply branch-and-bound to a search tree constructed of feasible motion primitives over a finite horizon, (ii) there is no fixed goal point towards which the algorithm presented here is driving, and (iii) as opposed to calculating an upper bound based on average variance reduction of a GP model, we calculate the upper bound based on an estimate of the ideal utility (Equation 15) from a set of unobserved query points, that is, potential future sampling locations. The overall methodology for BB-IPP is provided in Algorithm 2 and illustrated with a 2D example in Figure 4.

Algorithm 2 Branch and bound informative path planning (BB-IPP)

```

1: procedure PATHPLANNING_BB (GP,  $\mathbf{x}_{start}$ )
2:  $\triangleright$  set  $\gamma^* = 0$ 
3:  $\triangleright$  Generate a set of  $Q$  query points,  $\{\mathbf{p}_i\}_{i=1}^Q \in \mathcal{W}$ 
4:  $\triangleright$  Predict query point rewards  $\{L(\mathbf{p}_i)\}_{i=1}^Q$  using Equations (12), (13),
   and (15)
5:  $\triangleright$  Sort by increasing  $L_i$  and assign the upper bound as the  $\alpha$ -
   percentile  $L_{UB} = L_{[\alpha Q]}$ 
6:  $[\mathbf{z}^*, \gamma^*] = \text{BB\_RECURSION}(\mathbf{x}_{start}, \emptyset, 0, 0)$ 
   return  $\varphi_{\mathbf{z}^*}$ 
7: end procedure
8: procedure BB_RECURSION ( $\mathbf{x}$ ,  $\mathbf{z}_{parent}$ ,  $\gamma_{parent}$ ,  $j$ )
9: if  $j < H$  then
10:  $\triangleright$  Approximate maximum possible reward over the planning
    horizon
11:  $\gamma_{max} = \gamma_{parent} + (H - j)L_{UB}$ 
12: else
13:  $\gamma_{max} = -\infty$ 
14: end if
15: if  $\gamma_{max} > \gamma^*$  then
16:  $\triangleright$  Sample  $\{\xi_i\}_{i=1}^M$  (consisting of primitives over  $[t + j\delta_t, t + (j + 1)\delta_t]$ )
    according to Equation (18)
17:  $\triangleright$  Extend trajectories,  $Z = \{(\mathbf{z}_{parent}, \xi_i)\}_{i=1}^M$ 
18:  $\triangleright$  Predict reward of each sample  $\{\gamma_i\}_{i=1}^M$  using Equation (12), (13),
    (15), and (6)
19:  $\triangleright$  Sort  $Z$  by decreasing  $\gamma_i$ 
20: for each  $\mathbf{z}_{child} \in Z$  do
21:  $\triangleright$  Update total reward at child,  $\gamma_{child} = \gamma_{parent} + \gamma_i$ 
22: if  $\gamma_{child} > \gamma^*$  then
23:  $\gamma^* = \gamma_{child}$ ,  $\mathbf{z}^* = \mathbf{z}_{child}$ 
24: end if
25:  $[\mathbf{z}^*, \gamma^*] = \text{BB\_RECURSION}(\varphi_{\mathbf{z}_{child}}(t + \delta_t), \mathbf{z}_{child}, \gamma_{child}, j + 1)$ 
26: end for
27: end if
   return  $\mathbf{z}^*, \gamma^*$ 
28: end procedure

```

4.1.1 | Branching procedure

Algorithm 2 is based on a recursive best-first tree search adapted to the specific type of trajectory parametrization and heuristics employed. A tree with candidate trajectories over $[t, t + T]$ is constructed recursively starting from the current state, \mathbf{x}_{start} . States in the tree are extended at each level by a set of M feasible motion

primitives over δ_t that form the edges of the tree. It is desired that the set of candidate trajectories cover the reachable space of the system while minimizing M . To achieve this, trajectory parameterizations are generated through *stratified rejection sampling* by partitioning the domain into several semantically different classes, that is, “forward,” “left,” “right,” “down,” and “up,” to ensure balanced selection of primitives in each direction. Then each class is sampled uniformly, with primitives that violate the constraints (e.g., collisions) discarded. The main reason for stratifying the space is to achieve good coverage of the domain even with few samples, motivated by our real-time computation requirements. The quality of this proposed sample selection is analyzed in Appendix A.

More formally, let $\xi = (v_x, v_z, \omega_z)$ be a primitive parameterization and Ξ be the parameter space over a single stage δ_t . This parameter space is partitioned into $\Xi = \{\Xi_1, \dots, \Xi_M\}$, where

$$\Xi_i = [\bar{v}_{x,i} - \hat{v}_x, \bar{v}_{x,i} + \hat{v}_x] \times [\bar{v}_z - \hat{v}_z, \bar{v}_z + \hat{v}_z] \times [\bar{\omega}_z - \hat{\omega}_z, \bar{\omega}_z + \hat{\omega}_z],$$

with $\bar{v}_x, \bar{v}_z, \bar{\omega}_z$ denoting the chosen nominal velocity, depth rate, and turn rate, respectively, while $\hat{v}_x, \hat{v}_z, \hat{\omega}_z$ can be regarded as perturbations around the nominal values. The nominal values and perturbations are chosen to fully cover the space without any overlap. The set of single-stage primitives $\{\xi_i\}_{i=1}^M$ is then sampled according to

$$\xi_i \sim \text{Uniform}(\{\xi \in \Xi_i | g(\Gamma(\varphi_\xi(\tau))) \leq 0, t \in [t, t + \delta t]\}). \quad (18)$$

The multistage trajectory parameterization over the planning horizon then follows as $\mathbf{z} = (\xi_1, \dots, \xi_H)$, while the parameter sample space is defined as $\mathcal{Z} = \Xi^H$. The mapping from parameter space to a state-space trajectory is achieved by the function φ as outlined in Section 3.2.

4.1.2 | Bounding procedure

Analysis in Appendix A shows that the branching procedure produces desired reachability characteristics that ensure that the proposed method is capable of exploring all regions of the reachable workspace. However, since the ultimate objective is not to visit a single location, a dense branching set is not necessary if it is determined in an early stage that a branch is unlikely to maximize *information*. In this sense, to avoid the exponential number of possible trajectories, it is critical to utilize a heuristic for the upper bound on the reward at any given sampling location. The goal is to choose an upper bound L_{UB} that effectively restricts the search space while not pruning branches that could contain the optimal solution. The upper bound is computed by first generating a space-filling set of Q query points $\{\mathbf{p}_i\}_{i=1}^Q$, where \mathbf{p}_i are candidate sampling locations within the planning horizon reachable space. These query points are evaluated using Equation (15) to produce a set of rewards $\{L_i\}_{i=1}^Q$ that estimate the reward distribution. The upper bound on the reward L_{UB} is then assigned as the α -percentile of L_k , where $L_k \in \{L_1 \geq \dots \geq L_Q\}$. The value chosen for $\alpha \in [0, 1]$ represents a trade-off between computation time and

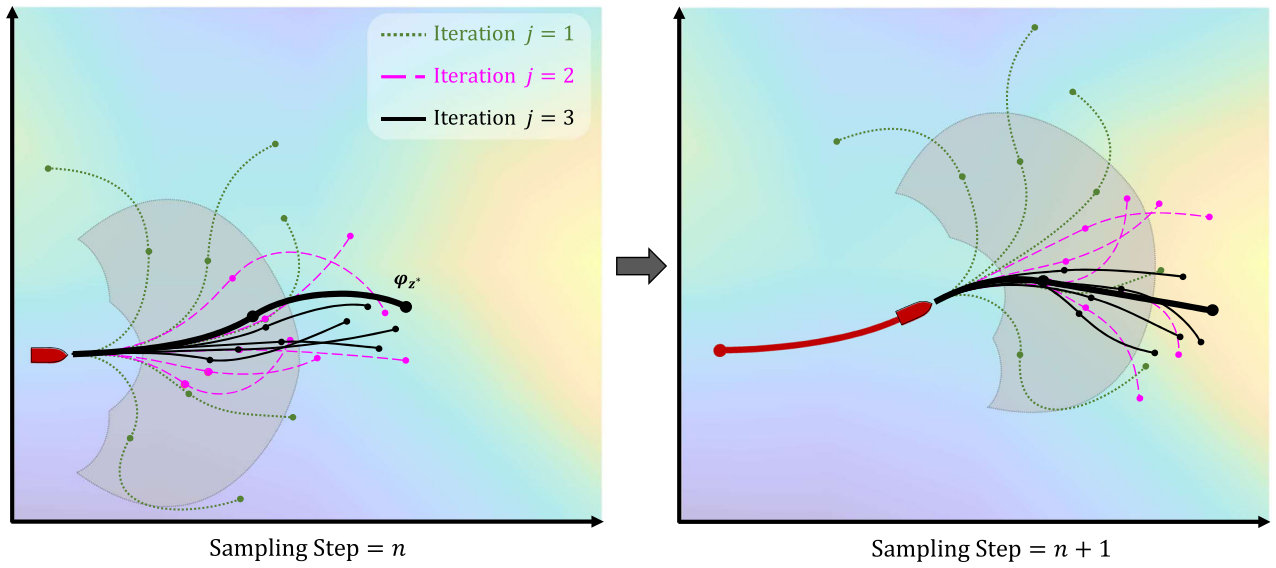


FIGURE 5 Illustrative example of the receding horizon cross-entropy informative path planning algorithm driving towards a high-reward (yellow) region of interest along the right edge of the sampling area. This 2D example uses $H = 2$ and trajectory parameterizations of $\mathbf{z} = (v_x, \omega_z)$ corresponding to speed and turn rate, respectively. For $v_x > 0$, the gray region represents the reachable space for one branching step over $t + \delta_t$ [Color figure can be viewed at wileyonlinelibrary.com]

optimality: small values of α will prune many branches of the tree and reduce the computation time but lead to greater risk of pruning the optimal solution. Conversely, large values of α result in a strict L_{UB} threshold that explores more of the search tree. Empirical parameter tuning experiments indicate that best performance is achieved when $0.9 \leq \alpha \leq 1$. Once all branches have been explored, BB-IPP of Algorithm 2 returns the optimal sampling trajectory.

4.2 | Cross-entropy informative path planning (CE-IPP)

A second method based on cross-entropy optimization offers an alternative for use in the PATHPLANNING procedure of Algorithm 1. The idea behind the cross-entropy method is to treat the optimization problem outlined in Equation (7) as an estimation problem of rare events. In what follows, we present the simplified version of the cross-entropy method and refer interested readers to De Boer et al. (2005), Kobilarov (2012), and Mannor et al. (2003), for the detailed formulation.

Define a collection of indicator functions $\{I_{J(\mathbf{z}) \geq \gamma}\}$ on \mathcal{Z} for various thresholds or levels γ . Consider the case when \mathcal{Z} has a probability density function $p(\cdot; \bar{\mathbf{v}})$ belonging to some parametric family $\{p(\cdot; \mathbf{v}), \mathbf{v} \in \mathcal{V}\}$ where $\bar{\mathbf{v}}$ is the true or nominal parameter. For instance, this could be a mixture of Gaussians. We can associate with Equation (8) the following estimation problem:

$$I(\gamma) = \mathbb{P}_{\bar{\mathbf{v}}}(J(\mathcal{Z}) \geq \gamma) = \mathbb{E}_{\bar{\mathbf{v}}}[I_{J(\mathbf{z}) \geq \gamma}] = \sum_{\mathbf{z}} I_{J(\mathbf{z}) \geq \gamma} p(\mathbf{z}, \bar{\mathbf{v}}). \quad (19)$$

The association comes from the fact that the probability $\mathbb{P}_{\bar{\mathbf{v}}}(J(\mathcal{Z}) \geq \gamma)$ will be very small (rare event) when γ is close to γ^* . By the cross-entropy

method, this rare event can be estimated by iteratively generating and updating a sequence of tuples $\{(\hat{\gamma}_j, \hat{\mathbf{v}}_j)\}$ such that it will converge to a small region of the optimal tuple $\{(\gamma^*, \bar{\mathbf{v}}^*)\}$.

Let \mathbf{v}_0 be the initial parameter, for instance $\mathbf{v}_0 = \bar{\mathbf{v}}$, and employing a multilevel approach using a sequence, the tuples $\{(\hat{\gamma}_j, \hat{\mathbf{v}}_j)\}_{j \geq 1}$ can be updated iteratively by:

- Let γ_j be the $(1 - \rho)$ -percentile of $J(\mathcal{Z})$ under \mathbf{v}_{j-1} . An estimate of γ_j , denoted $\hat{\gamma}_j$, can be obtained by drawing M samples from $p(\mathbf{z}, \mathbf{v}_{j-1})$ and assigning $\hat{\gamma}_j$ as the $(1 - \rho)$ -percentile of $\{\gamma_i\}_{i=1}^M$.
- With fixed $\hat{\gamma}_j$ and \mathbf{v}_{j-1} , the estimate of \mathbf{v}_j , denoted $\hat{\mathbf{v}}_j$, can be derived from Kobilarov (2012):

$$\hat{\mathbf{v}}_j = \underset{\mathbf{v} \in \mathcal{V}}{\operatorname{argmax}} \frac{1}{|\epsilon_j|} \sum_{\mathbf{z}_i \in \epsilon_j} \ln p(\mathbf{z}_i, \mathbf{v}), \quad (20)$$

where ϵ_j is the elite set of the samples, that is, samples \mathbf{z}_i for which $\gamma_i \geq \hat{\gamma}_j$.

Instead of updating $\hat{\mathbf{v}}_j$ directly with Equation (20), we apply a simple smoothing filter:

$$\hat{\mathbf{v}}_j = \kappa \hat{\mathbf{v}}_j + (1 - \kappa) \hat{\mathbf{v}}_{j-1}, \quad (21)$$

where κ is the smoothing parameter with $0.7 < \kappa < 1$. The filter serves two purposes: (i) smoothing the parameter update, and (ii) avoiding $\hat{\mathbf{v}}_j$ from converging prematurely into a local minima, especially during the initial stage of the optimization process. A step-by-step process of the cross-entropy informative path planning (CE-IPP) is given in Algorithm 3 and illustrated with a 2D example in Figure 5. Once again, the sampled trajectory parameterizations for the AUV motion profile consist of a tuple sequence $\mathbf{z} = \{(v_x, v_z, \omega_z)_i\}_{i=1}^H$.

Algorithm 3 Cross entropy informative path planning (CE-IPP)

```

1: procedure PATHPLANNING_CE(GP,  $\mathbf{x}_{start}$ )
2: ▷ Set  $j = 0, \hat{y}_0 = 0$ , and  $\hat{v}_0 = v_0$ 
3: repeat
4: Set  $j = j + 1$ 
5: ▷ Generate samples  $\{\mathbf{z}_i\}_{i=1}^M$  from  $p(\cdot; \hat{v}_{j-1})$ , starting from  $\mathbf{x}_{start}$ 
6: ▷ Reject samples with constraint violations  $g(\varphi_{z_i}) > 0$ 
7: ▷ Predict reward of each sample  $\{\gamma_i\}_{i=1}^M$  using Equations (12), (13), (15), and (6)
8: ▷ Sort  $\gamma_i$  in ascending order and compute the  $(1 - \rho)$ -percentile  $\hat{y}_j = J_{(1-\rho)M}$ 
9: ▷ Update  $\hat{v}_j$  using Equation (20) over the elite set  $\epsilon_j = \{\mathbf{z}_i | \gamma_i \geq \hat{y}_j\}$ 
10: until  $j = K$  ▷  $K$  is number of iterations
11: ▷ Choose the sample with the maximum reward
12:  $\mathbf{z}^* = \underset{\mathbf{z}_i}{\operatorname{argmax}} J(\mathbf{z}_i)$ 
    return  $\varphi_{z^*}$ 
13: end procedure

```

5 | SIMULATION STUDIES

Monte Carlo simulation studies were performed to evaluate the overall GPAS method and each informative path planner in sensing and localizing multiple ROIs. Quantitative and qualitative validation of the GPAS method is provided against conventional sensing approaches in a variety of realistic 3D environments. With regard to the Chesapeake Bay water quality applications, ROIs are described here by hypoxic zones—areas of the water column characterized by a dearth of dissolved oxygen. Values for dissolved oxygen in the Chesapeake Bay typically range between 0 and 9 mg/L, with hypoxic zones designated by dissolved oxygen values less than 2 mg/L (Hagy et al., 2004). Thus, to maintain the assumption that measurements are normalized in the range 0–1, ROIs are defined as areas of the environment that exhibit normalized measurements below a threshold of $\frac{2 \text{ mg/L}}{9 \text{ mg/L}} \approx 0.22$. Hypoxic zones typically occur near the seafloor because pycnocline layers of fresh and salt water prevent oxygen-rich surface water from mixing with deeper water. Decomposition of algae and other phytoplankton beneath the pycnocline further exacerbates the consumption of dissolved oxygen (Hagy et al., 2004; Kemp et al., 2005). To emulate this phenomenon, the ROIs in the virtual environmental fields used for the simulation studies are a linear function of depth and reach their maximum volume at the seafloor. Further, as described in Section 1, all environmental fields are assumed to be static.

Figure 6 shows one such virtual environmental field used for the studies of Sections 5.1 and 5.2: a 300 m × 300 m × 5 m volume with two ROIs near the seafloor. The volume of the ROIs in this

environmental field account for approximately 1.5% of the total sampling volume. No prior information of the environment is assumed and the initial density of the GP model is set to a uniform distribution across the scalar field. For each simulation, a different random seed was used for trajectory generation and the starting location of the AUV was randomly selected to be on the surface within a 25 m × 25 m box in the bottom left corner of the environmental area. Performance metrics used to evaluate the planning approaches are given in Table 2.

Additionally, low-cost AUV operations require periodic surfacing for GPS fixes to maintain navigation accuracy. Thus, in the simulations and field experiments that follow, a heuristic was implemented such that the vehicle surfaces for a GPS reading after traveling a user-specified distance. For instance, to maintain position accuracy of 5 m in nominal conditions, the vehicle should resurface after traveling approximately 500 m. Future work will examine methods for incorporating surfacing decisions into the path planning such as those explored in Kemna and Sukhatme (2018).

5.1 | Algorithm parameter tuning

An initial Monte Carlo simulation study was performed to evaluate parameter variations of each informative path planner. The resulting parameter values chosen for subsequent simulation analysis and field experiments are given in Table 3. Complete details of the methodology and performance for each parameter combination are reserved for Appendix B.

Overall, the parameters for each algorithm were selected to balance planning time, GP model fidelity, and the time spent in ROIs. A comparison between the two planning methods shows that while both produced the same level of GP model fidelity, the CE-IPP

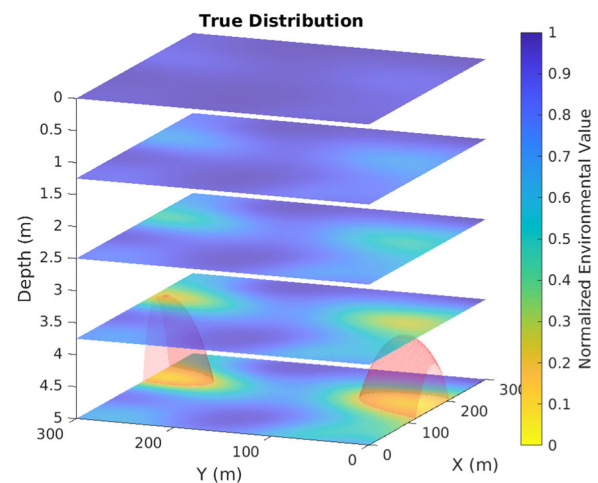


FIGURE 6 True environmental distribution used for simulation studies with a static field normalized in the range 0–1. Region of interests are shown in red as the volume below the desired measurement threshold of 0.22 [Color figure can be viewed at wileyonlinelibrary.com]

TABLE 2 Summary of the performance metrics used to evaluate the planning approaches

Metric	Description
Planning time	The average computation time to select the optimal path at each sampling step n over the set of simulations. The simulations were run on a single laptop with a 2.8 GHz Intel i7-7700HQ processor with 16 GB of RAM. Each algorithm was implemented in Matlab on Ubuntu 18.04 for prototype analysis, although significant speed improvements could be achieved by implementing the algorithms in C++.
Reward per segment	The average optimal reward of the generated trajectory divided by the number of segments in the planning horizon, that is, γ^*/H , over the set of simulations.
GP RMSE	The average RMSE of the estimated GP model at the end of the sampling mission over the set of simulations.
GP variance	The average variance of the estimated GP model at the end of the sampling mission over the set of simulations. The variance of the GP model is calculated as the mean of the covariance matrix diagonal.
Percent mission in ROI	The average time spent within ROIs divided by the total mission time over the set of simulations.

Abbreviations: GP, Gaussian process; RMSE, root-mean-square error; ROI, region of interest.

method produces paths with slightly higher rewards and consistently spends a higher percentage of the mission sampling within ROIs. This comes at the expense of greater computation time, as the BB-IPP method is approximately 5–10× faster than the CE-IPP method, particularly when evaluating trajectories over a longer planning horizon. Neither method, however, is prohibitively expensive to preclude further examination, especially if the algorithms were to be optimized for runtime performance.

5.2 | Performance over the sampling mission

This section further analyzes the performance of each planner over the time history of the sampling mission. A baseline planning approach consisting of a preprogrammed undulating lawnmower (LM) path was also simulated to illustrate the advantages of the proposed GPAS method against conventional sensing approaches. In typical operations, a pre-programmed coverage mission such as this is executed without regard to the collected data. It is not until the data is analyzed after the vehicle returns that ROIs may be identified, at which point the vehicle

Com- ponent	Parameter	Description
General	$N = 160$	Number of sampling points over mission ($N = t_f/\delta_t$)
	$t_f = 6000$ s	Total sampling mission time (including surfacing events)
	$\delta_t = 35$ s	Time step between each collected sample
	$T = 140$ s	Planning horizon time length
	$\delta = 0.9$	Probability regret bound is satisfied after N sampling steps
BB-IPP	$\alpha = 0.95$	Upper bound percentile threshold
	$Q = 2000$	Number of query points used to estimate the utility distribution
	$\bar{v}_x \in \{1.0, 1.4\}$ m/s	Nominal speed trajectory parameterizations
	$\hat{v}_x = 0.2$ m/s	Perturbation magnitude from each \bar{v}_x
	$\bar{v}_z \in \{-0.08, 0, 0.08\}$ m/s	Nominal depth rate trajectory parameterizations
	$\hat{v}_z = 0.04$ m/s	Perturbation magnitude from each \bar{v}_z
	$\bar{\omega}_z \in \{-0.06, 0, 0.06\}$ rad/s	Nominal yaw rate trajectory parameterizations
	$\hat{\omega}_z = 0.03$ rad/s	Perturbation magnitude from $\bar{\omega}_z$
CE-IPP	$K = 9$	Maximum number of iterations
	$M = 300$	Number of trajectories evaluated at each K th iteration
	$\rho = 0.1$	Percentile threshold for specifying elite sample set
	$\kappa = 0.9$	Parameter update smoothing filter value
	$v_x \in [0.8, 1.6]$ m/s	Range of available speeds from which to sample
	$v_z \in [-0.12, 0.12]$ m/s	Range of available depth rates from which to sample
	$r \in [-0.09, 0.09]$ rad/s	Range of available yaw rates from which to sample

TABLE 3 Algorithm parameters used for simulation studies

would need to be redeployed to collect additional data in these areas. The adaptive scheme of the GPAS method addresses this problem by using the acquisition function of Equation (15) to balance the trade-off between data collection in known ROIs (through GP model mean predictions) and exploration of unknown and/or uncertain areas of the environment (through GP model variance predictions).

5.2.1 | Quantitative performance

Figure 7 shows the performance of each path planning algorithm as a function of the sampling step, averaged over a total of 150 simulations. While the planning times for each GPAS method are longer than the baseline LM path, it is evident that the trajectories planned

by the GPAS method consistently produce higher rewards throughout the mission. Although not shown in Figure 7, this is emphasized by the percentage of the total mission time spent in ROIs: 13.1% of the mission for the BB-IPP method, 20.9% of the mission for the CE-IPP method, and 2% of the mission for the baseline LM path, even though the volume of the ROIs account for only 1.5% of the total sampling area volume. Both the mean reward per segment and the mission percentage in ROIs indicate that the trajectories produced by GPAS significantly outperform the baseline LM path in collecting valuable measurements. In particular, the CE-IPP method produces slightly better rewards than the BB-IPP method and is able to allocate the most sampling time to the ROIs.

While these results indicate that the GPAS method is superior in collecting valuable measurements in ROIs, it is important to ensure that

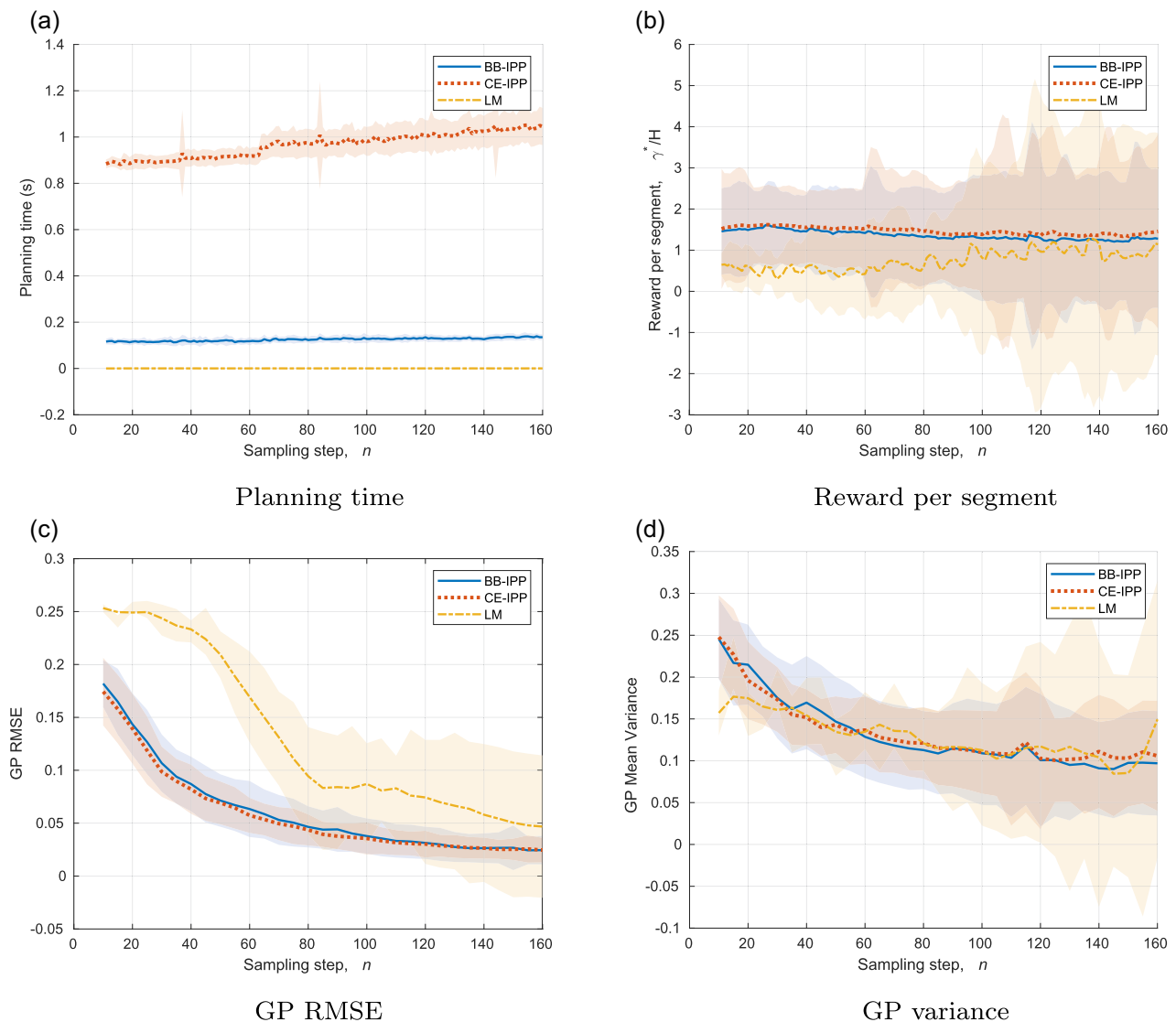


FIGURE 7 Metrics used to evaluate the performance of each path planning method over a set of 150 simulations (LM is the baseline undulating lawnmower path). Plots begin at the 10th sampling step so as to let the GP train with an initial set of measurements. The lines are the mean values and the shaded regions capture values within one SD around the mean. BB-IPP, branch-and-bound informative path planning; CE-IPP, cross-entropy informative path planning; GP, Gaussian process; RMSE, root-mean-square error [Color figure can be viewed at wileyonlinelibrary.com]

prediction over the entire environmental field is maintained. Figure 7c,d show the GP root-mean-square error (RMSE) and GP variance as a function of the sampling step, respectively. While the GP variance is similar for all planning methods, the lower RMSE values of the GPAS method compared to the baseline LM path show that the adaptive trajectories produce more accurate surrogate models over the course of the mission. This is intuitive, as the adaptive trajectories initially choose to explore areas of the environment with high uncertainty. As the mission progresses, a more accurate GP model means that additional effort can be focused on further improving the model in both ROIs (exploitation) and uncertain areas (exploration).

Additionally, an ablation study was performed on the GP model to determine its ability to predict ROIs in the absence of direct measurements within these regions. For each simulation, measurements collected within ROIs were removed from the full set of collected measurements and the GP model was retrained on the subset of measurements. This reduced GP model was then used to predict the values of the removed ROI points as well as the values of the full measurement set, resulting on average RMSE values of 0.078 and 0.055, respectively. These RMSE values are only slightly higher than those observed from the GP model trained on the full measurement set. Further, an average of 83% of the removed measurements were still predicted below the ROI threshold, indicating that the GP model is capable of modeling and predicting these regions even in the absence of direct measurements.

Figure 8 captures the hyperparameter estimates for each path planning method over the same set of simulations. All strategies

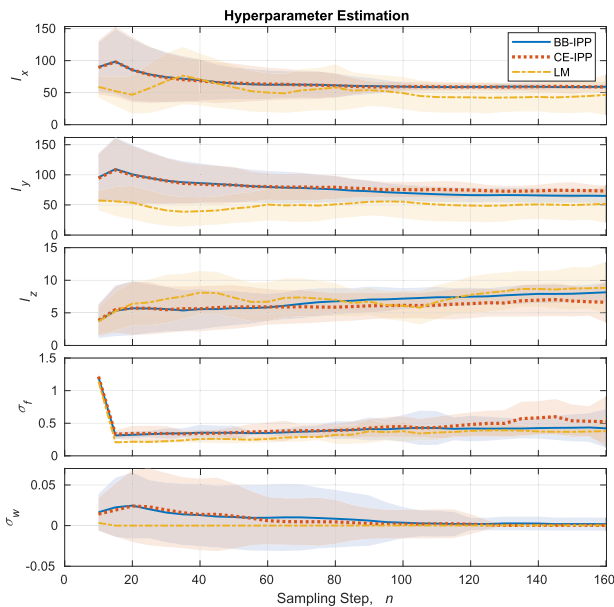


FIGURE 8 Hyperparameter estimates $\theta^* = [l_x, l_y, l_z, \sigma_l, \sigma_w]^T$ for each path planning method over a set of 150 simulations (LM is the baseline undulating lawnmower path). The lines are the mean values and the shaded regions capture values within one SD around the mean. BB-IPP, branch-and-bound informative path planning; CE-IPP, cross-entropy informative path planning [Color figure can be viewed at wileyonlinelibrary.com]

produced relatively similar hyperparameter estimates that began to stabilize towards the end of the sampling mission. To compare the quality of these hyperparameter estimates, however, the RMSE was calculated using new GP models that were retrained with the final, averaged hyperparameters estimated by each path planning method. This resulted on average RMSE values of 0.038 for BB-IPP, 0.039 for CE-IPP, and 0.047 for the baseline LM method. These results are fairly consistent with the trends seen in Figure 7c and provide further evidence that the GPAS method tends to produce models with overall lower RMSE values.

5.2.2 | Example trajectories

Figure 9 shows the resultant trajectories and sampling locations of candidate simulations for each path planning method. These trajectories and sampling locations are overlaid on the estimated GP model of each simulation using the hyperparameter estimates at the end of the mission. Additionally, a time-series plot of the measurement signal is provided for each trajectory with the red portion signifying time spent within an ROI. Even though the trajectories planned by the GPAS method did not provide as complete area coverage as the baseline LM path, they exhibit a good exploration-exploitation trade-off by initially exploring the environment to learn a model of the underlying scalar field, and subsequently focusing samples within the ROIs.

The results shown in Figure 9 also indicate that the GPAS method is capable of detecting and localizing multiple ROIs within a scalar field. Even though the signal level plots show that the first ROI is discovered early in the mission, the GPAS method continues to explore the rest of the field, as the variances around unexplored regions were still high. After the second ROI is discovered by both GPAS planning methods, the sampling is then focused alternatively between the two ROIs, a behavior that results from the GP-UCB criteria to minimize cumulative regret given the number of samples remaining in the mission, the current knowledge/uncertainty of the environment from collected data, and a limited planning horizon. As data is collected in one ROI, the uncertainty of that ROI decreases, and the utility of other areas of the environment (including other ROIs) increases as a result. These results show that the GPAS method is not prone to local minima even though the measurement signal level for both ROIs is comparable. In contrast, since the baseline LM path is preplanned with equal sampling step size, the samples collected along the trajectories are fixed. Furthermore, depending on the vehicle's starting position and the location of the ROI within a given environment, the detection of a ROI is random at best.

5.3 | Performance on randomized environmental fields

While the simulation results explored so far highlight the advantages of the GPAS algorithm for a candidate test environment, it is

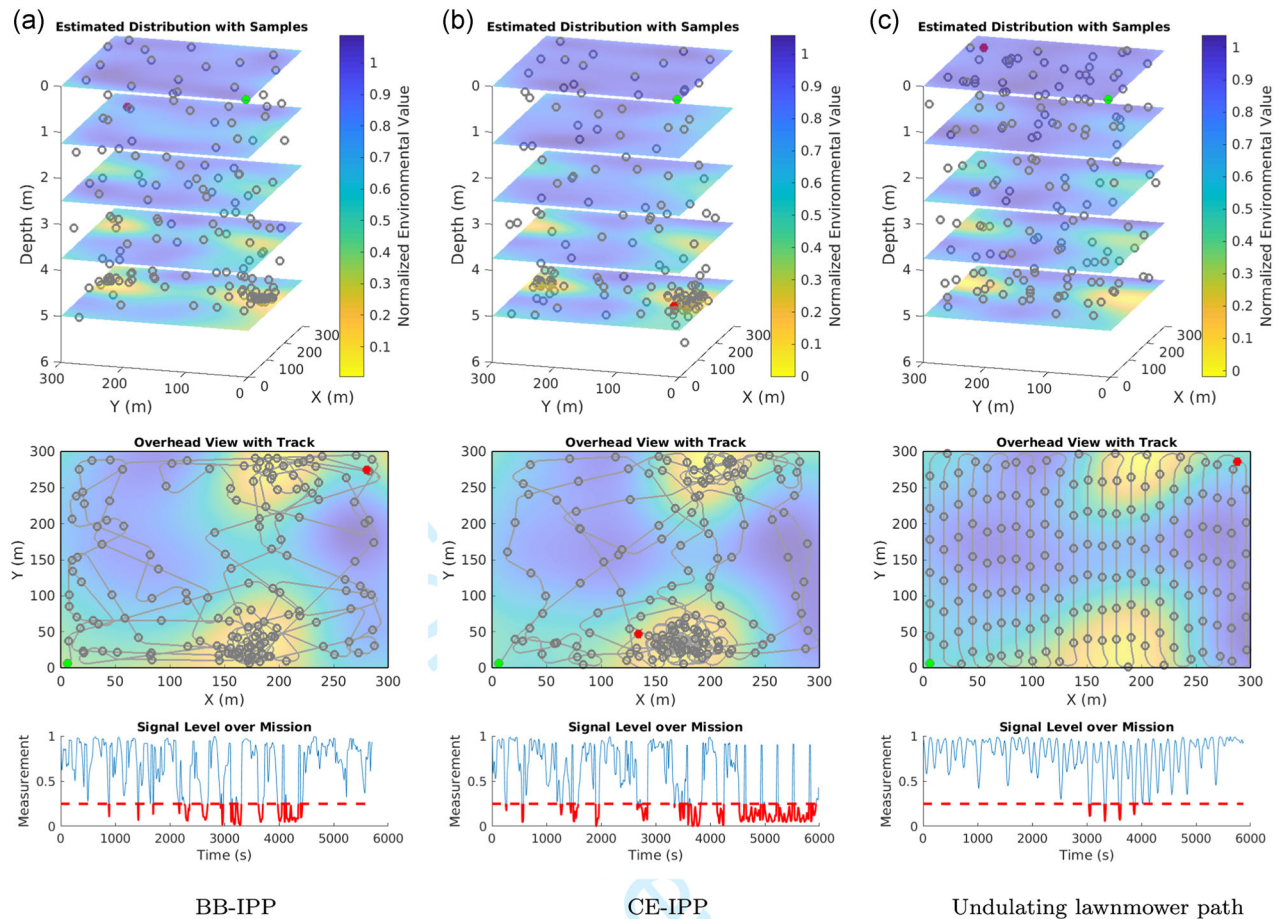


FIGURE 9 Simulation results for three different planning surveys. The top images shows the sampling points in 3D while the middle images show a bird's-eye view of the vehicle trajectory with the estimated distribution at the seafloor. The time series plot on the bottom shows the normalized measurement value over the simulation where measurements within an ROI are highlighted in red. BB-IPP, branch-and-bound informative path planning; CE-IPP, cross-entropy informative path planning [Color figure can be viewed at wileyonlinelibrary.com]

important to study whether the results apply to any realistic sampling environment. This section gathers additional statistical evidence of the performance of the BB-IPP, CE-IPP, and baseline LM approaches on a set of 500 randomized, auto-generated sampling environments. The length, width, and depth of each environment was varied to create a range of sampling volumes between approximately 5×10^4 and 3.5×10^6 m³. The static field distribution was randomly generated in the horizontal plane and, similar to Figure 6, linearly scaled based on the depth.

In addition to randomly generating the environmental field, bathymetry was also generated for added realism. Consideration of local bathymetry is particularly important for operations in the Chesapeake Bay, as it is typical for hypoxic regions to form close to the seafloor. Bathymetry-aware planning is thus necessary to reduce the risk of vehicle loss when operating in environments characterized by large variations in bathymetry, significant bottom vegetation, and muddy soil. An example of a randomly generated sampling environment is shown in Figure 10.

Slight modifications to the test conditions and algorithms were implemented to accommodate the new environments. The

overall mission time for each environment was scaled according to the relative volume of each environment. Similarly, the baseline LM path was precomputed to scale according to the relative volume of each environment, not intersect with the seafloor, and cover the entire sampling volume within the mission time. For the adaptive planners, bathymetry was modeled throughout the sampling mission by incorporating measurements from simulated altimeter and pressure sensors. This bathymetry model was used to predict the vehicle's height from the seafloor along each trajectory considered during path planning. If any trajectory violated a safe minimum height from the seafloor, the depth rate v_z was saturated such that the vehicle never violated the safe height threshold. Any trajectory that violates the bathymetry constraints after saturating v_z is removed from consideration. In this way, the resulting motion primitives remain dynamically feasible while safely navigating variable bathymetry profiles as shown in Figure 10. In addition to considering bathymetry during path planning, the waypoint following autopilot was extended to always keep the current state of the vehicle above the safe minimum height while transiting between waypoints.

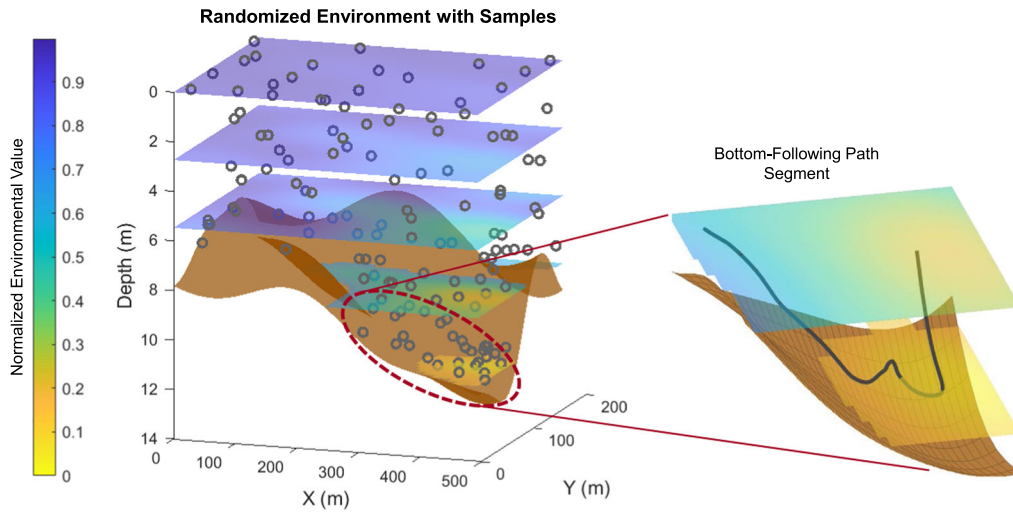


FIGURE 10 An example of a randomly generated environmental distribution and the resulting sampling points. Bathymetry is shown in brown where the right figure illustrates a portion of the trajectory exhibiting bottom-following at a safe minimum height [Color figure can be viewed at [wileyonlinelibrary.com](https://onlinelibrary.wiley.com)]

Evaluation metrics for each planning method are shown in the form of box plots in Figure 11, including a generalized ROI mission ratio (RMR) to account for variation between the randomized environments:

$$\text{RMR} = \frac{(t_{\text{ROI}}/t_f)}{(V_{\text{ROI}}/V)}, \quad (22)$$

where t_{ROI} is the time spent within ROIs, V_{ROI} is the volume of the ROIs, and V is the total volume of the test environment. This metric normalizes the percentage of time spent within ROIs based on the actual volume of the ROIs within different environments. Overall, the performance observed on the randomized environments is consistent with that of Section 5.2, where the BB-IPP and

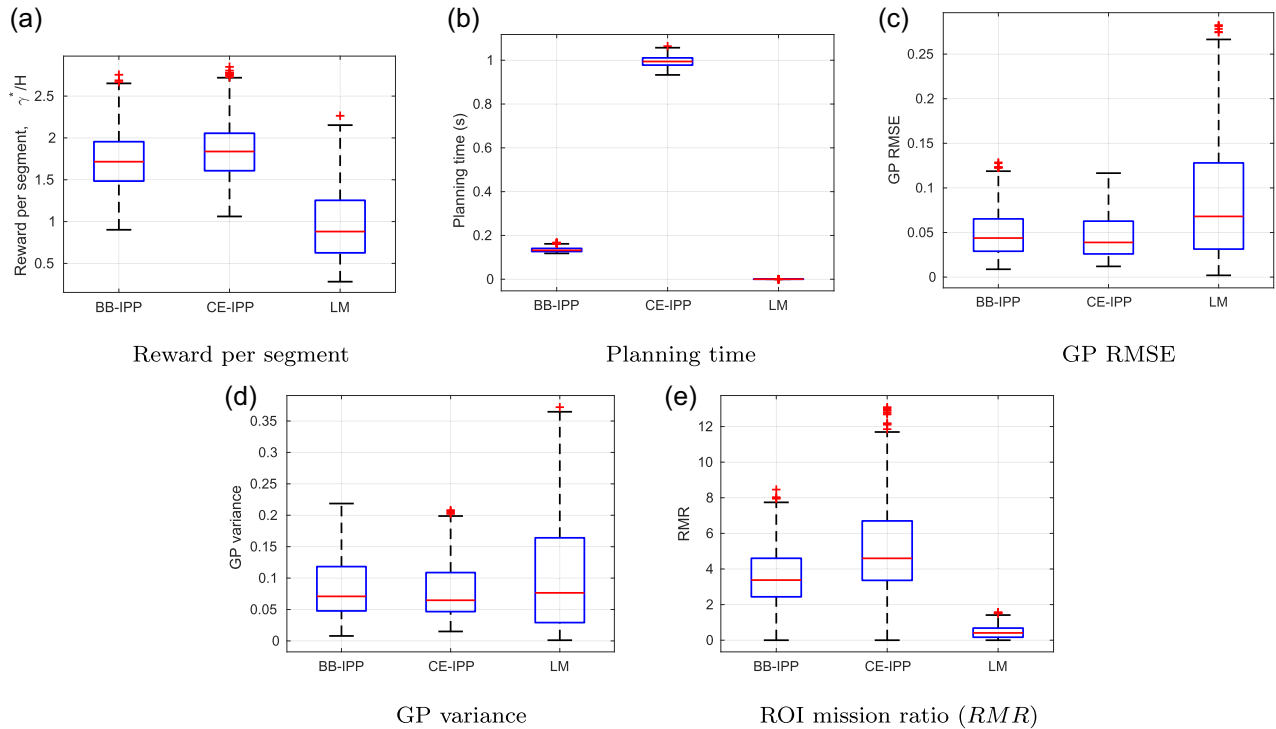


FIGURE 11 Performance metric box plots for each path planning method over a simulation set of 500 randomly generated sampling environments. The GP RMSE and variance results were based on a GP model trained at the end of each mission. GP, Gaussian process; RMSE, root mean square error; ROI, region of interest [Color figure can be viewed at [wileyonlinelibrary.com](https://onlinelibrary.wiley.com)]

CE-IPP algorithms outperform the baseline LM approach on all metrics except planning times (as expected). In particular, the RMR values

illustrate that the GPAS algorithm generalizes to a variety of realistic, constrained sampling environments and is able to dedicate more sampling time within ROIs (approximately 3–5× for BB-IPP and 4–7× for CE-IPP) when compared to the baseline LM approach. Further, the GPAS method produced lower GP RMSE and variance values than the baseline LM coverage pattern, indicating that the adaptive sampling does not compromise the fidelity of the final GP model. The minimal spread of the planning times also indicates that the planning algorithms are agnostic to the environment being sampled and can repeatably produce new sampling points.

6 | FIELD EXPERIMENTS

As a demonstration of the algorithm, two field experiments were carried out in the Severn River, Maryland using the JHU Ocean-server Iver3 AUV of Figure 3. The goal was to use the GPAS method on an operational platform to fully explore the environment workspace while also prioritizing the collection of measurements within ROIs. The first experiment prescribed the AUV to take simulated measurements from the virtual environmental field of Figure 6. This virtual field provides a perfect ground truth from which to evaluate the GP modeling. Additionally, this experiment is used to determine if the results from the simulation studies translate to an operational platform, thus giving confidence to the statistical results of the Monte Carlo testing in Section 5.3. The second experiment then used the GPAS algorithm to collect in situ dissolved oxygen measurements for the purpose of discovering and modeling hypoxic zones. The locations of the two field experiments are shown in Figure 12.

6.1 | Test platform

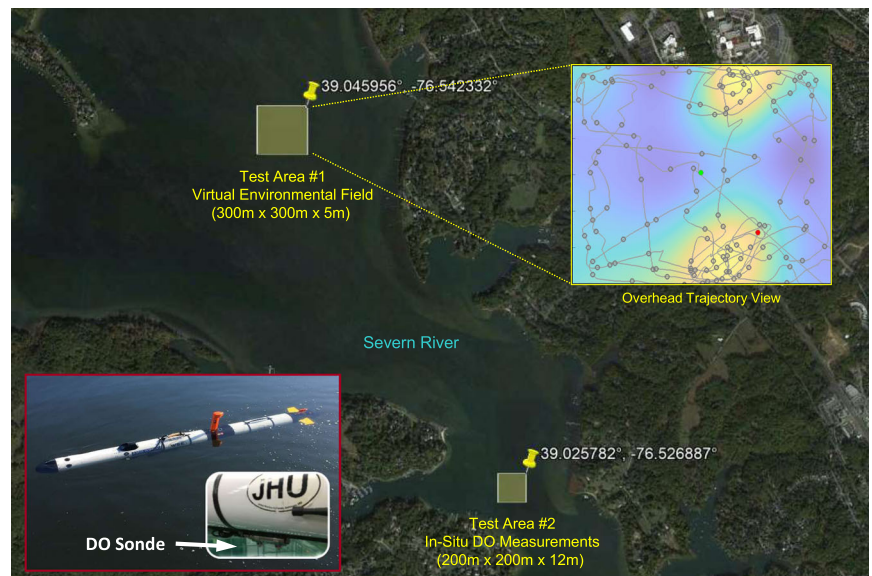
The AUV is equipped with two on-board computers for autonomous control as well as various sensors for vehicle localization and navigation including GPS, IMU, Doppler velocity log (DVL), altimeter, and pressure sensors. In addition, the vehicle was outfitted with a JFE Advantech RINKO-FT dissolved oxygen sonde for measuring the environmental features of interest. Communication with the AUV is achieved through Wi-Fi when it is on the surface and through acoustic communication when it is submerged. Due to the limited bandwidth available through acoustic communication, only vehicle state updates and safety commands (e.g., abort commands) are sent between the operator and the vehicle. Thus, all processing regarding the path planning and GP modeling is performed by the vehicle's on-board computers.

The two on-board computers have separate functionality in a front-seat/back-seat configuration. The front-seat computer acts as the autopilot of the AUV. This computer processes sensor data to estimate the vehicle's state and accepts commands in the form of desired heading, desired speed, and desired depth. The back-seat computer hosts the GPAS and path planning algorithms. This computer receives vehicle state information from the front-seat and provides the navigation commands for the current sampling point. Additionally, the back-seat computer contains software to provide safety oversight including geofencing, abort overrides, safety thresholds, and heartbeat monitoring of all software components. The open-source Robot Operating System (ROS) (Quigley et al., 2009) was used as the framework for managing information between each software component on both on-board computers.

6.2 | Implementation details

As seen in the results of the simulation studies, there exists a trade-off between performance and computational time between the two

FIGURE 12 Autonomous underwater vehicle field experiments were conducted in the Severn River, Maryland. The northernmost test area, referred to as TA1, ran the GPAS algorithm using virtual measurements from a simulated environmental field with multiple region of interests. The southernmost test area, referred to as TA2, ran the GPAS algorithm using in situ dissolved oxygen measurements. GPAS, Gaussian process adaptive sampling [Color figure can be viewed at wileyonlinelibrary.com]



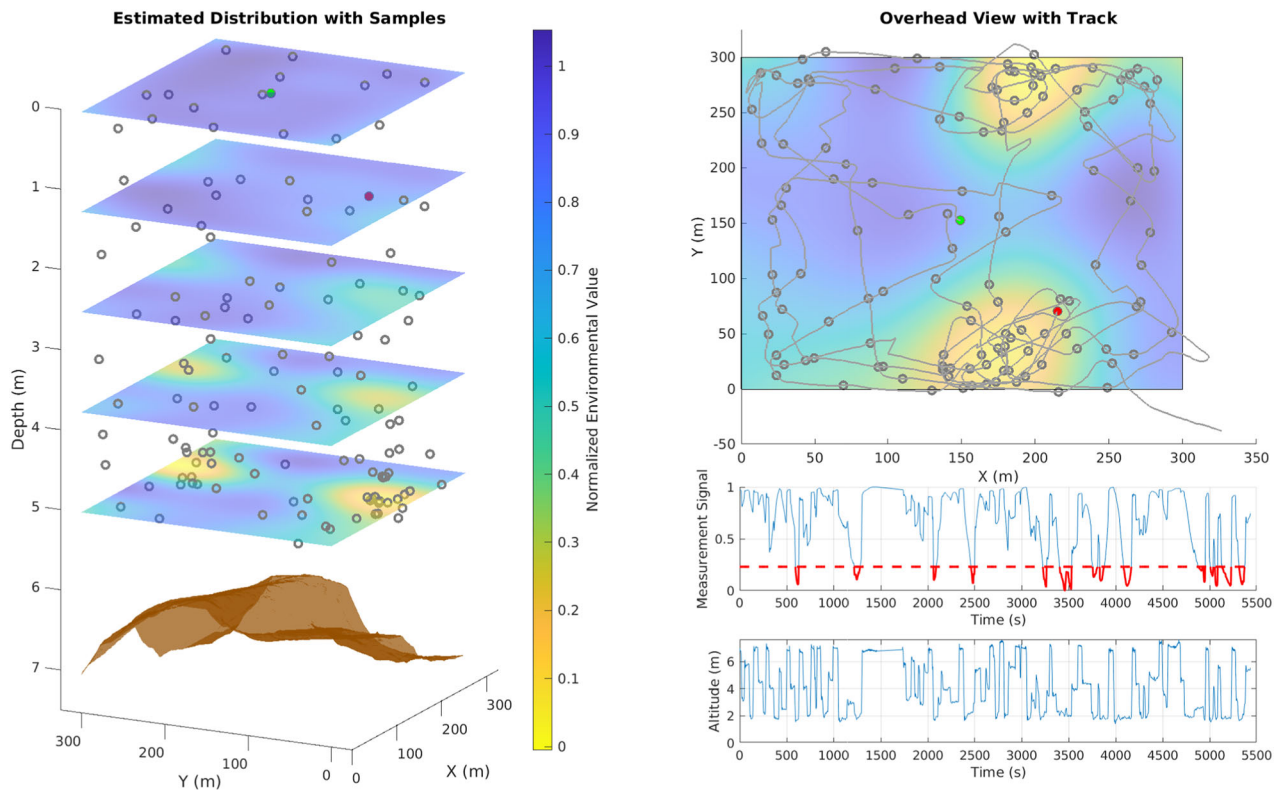


FIGURE 13 Results for the field experiment performed in TA1 using the virtual environmental distribution of Section 5. The left image shows the estimated Gaussian process model with sampling points in 3D. The measured bathymetry of the test area is shown in brown. The top right plot shows a bird's-eye view of the autonomous underwater vehicle trajectory with the estimated distribution at the seafloor. The time series plots in the bottom right show the normalized virtual measurements, where measurements within a region of interest are highlighted in red, and the vehicle altitude [Color figure can be viewed at wileyonlinelibrary.com]

planning algorithms: the CE-IPP algorithm slightly outperforms in collecting higher rewards and spending more time in ROIs, while the BB-IPP algorithm requires less computational expense. This trade-off is important because the AUV on-board computers are significantly less powerful than standard laptop or desktop computers. Thus, due to the faster planning times and the relatively constrained sampling areas, the BB-IPP algorithm was chosen as the path planning subroutine implemented on the AUV for field experiments. The advantages of the more computationally expensive CE-IPP algorithm become more pronounced when performing sampling missions in large areas. Slightly longer processing times are acceptable on larger grids because the sampling waypoints can be spaced further apart. Future work in optimizing the implementation of these algorithms could overcome this limitation. Further, sampling platforms with increased computational capabilities (e.g., unmanned surface vessels) could accept the more expensive CE-IPP algorithm.

Additionally, in a similar manner to Section 5.3, the planning algorithm was required to maintain a minimum height of 1.5 m from the seafloor as a conservative safety condition to prevent vehicle loss, either through entanglement in bottom vegetation or getting stuck in loose mud. Enforcing this condition was done by continuously estimating the local bathymetry using the vehicle's on-board altimeter (to measure height from bottom) and pressure

sensor (to measure depth). This safety condition, however, should not prohibit detection of ROIs, as technical reports of the testing area (Henkart, 2012; Miller et al., 2019) indicate that there is not significant variation in dissolved oxygen measurements within 2 m of the seafloor (typically <1 mg/L). Further, the results of the ablation study in Section 5.2 indicate that ROIs can still be accurately predicted by the resulting GP model using nearby measurements. As additional emergency recovery mechanisms are added, this safety condition could be relaxed to further minimize height from the seafloor.

6.3 | Experiment #1: Virtual environmental field

The first experiment was designed to determine if the results obtained in the simulation studies translated to an operational environment. To this end, the experiment utilized the virtual environmental field of Figure 6 such that the GP model on-board the vehicle was trained via virtual measurements from this distribution based on the AUV's current position. This experiment was performed in the northernmost test area (TA1) of Figure 12. This area was chosen because it is large enough to accommodate the virtual environmental field and consisted of fairly constant bathymetry that would not interfere with sample selection. Algorithm parameters were identical

TABLE 4 Summary of performance metrics observed for each field experiment

Ex-periment	Planning time (s)	GP RMSE	GP variance	Percent mission in ROI (%)
TA1	0.67 ± 0.21	0.056	0.098	17
TA2	0.63 ± 0.20	–	0.42 mg/L	48

Note: Values for the GP RMSE and GP variance are measured at the end of the sampling mission. GP RMSE was not calculated for the experiment in TA2 due to the lack of a ground truth dissolved oxygen source.

Abbreviations: GP, Gaussian process; RMSE, root mean square error; ROI, region of interest.

to those in Table 3 with the exception that $t_f = 5200$ s and $\delta_t = 40$ s. This produces trajectories of $N = 130$ sampling points and allows for additional time required for GPS surfacing events.

The results of the first experiment performed in TA1 are shown in Figure 13 while the performance metrics are summarized in Table 4. The plots show that the resulting sampling points and vehicle trajectory are very similar to those observed in the simulation studies. Additionally, the performance metrics of Table 4 are consistent with the simulation results observed in Figure 7 with the exception of longer planning times due to less processing power on the AUV on-board computers. Both of these facts demonstrate the

practicality of the proposed method in a real-world application while also giving further confidence to the statistical results from the randomized environment testing of Section 5.3.

One can observe in the TA1 results of Figure 13 that the AUV unexpectedly navigated outside of the safety geofence due to a series of failed checksum messages in the commanded inputs. This event occurred approximately 1400 s into the mission and is evident by the break in the AUV's 2D trajectory outside the bottom right corner of the sampling area. At this time, safety mechanisms caused the AUV to travel back to the center of the mission area. The AUV resumed operations once back at the center of the sampling area approximately 1700 s into the mission and successfully completed the remainder of the mission. Overall, this caused a 300 s delay from the originally intended 5200 s total mission time.

6.4 | Experiment #2: In situ measurements from dissolved oxygen field

The second experiment was performed in the southernmost test area (TA2) of Figure 12. This experiment was designed to demonstrate the GPAS algorithm in collecting samples based on an environmental field estimated from in situ dissolved oxygen measurements. As described in Section 5, dissolved oxygen values in the Chesapeake Bay

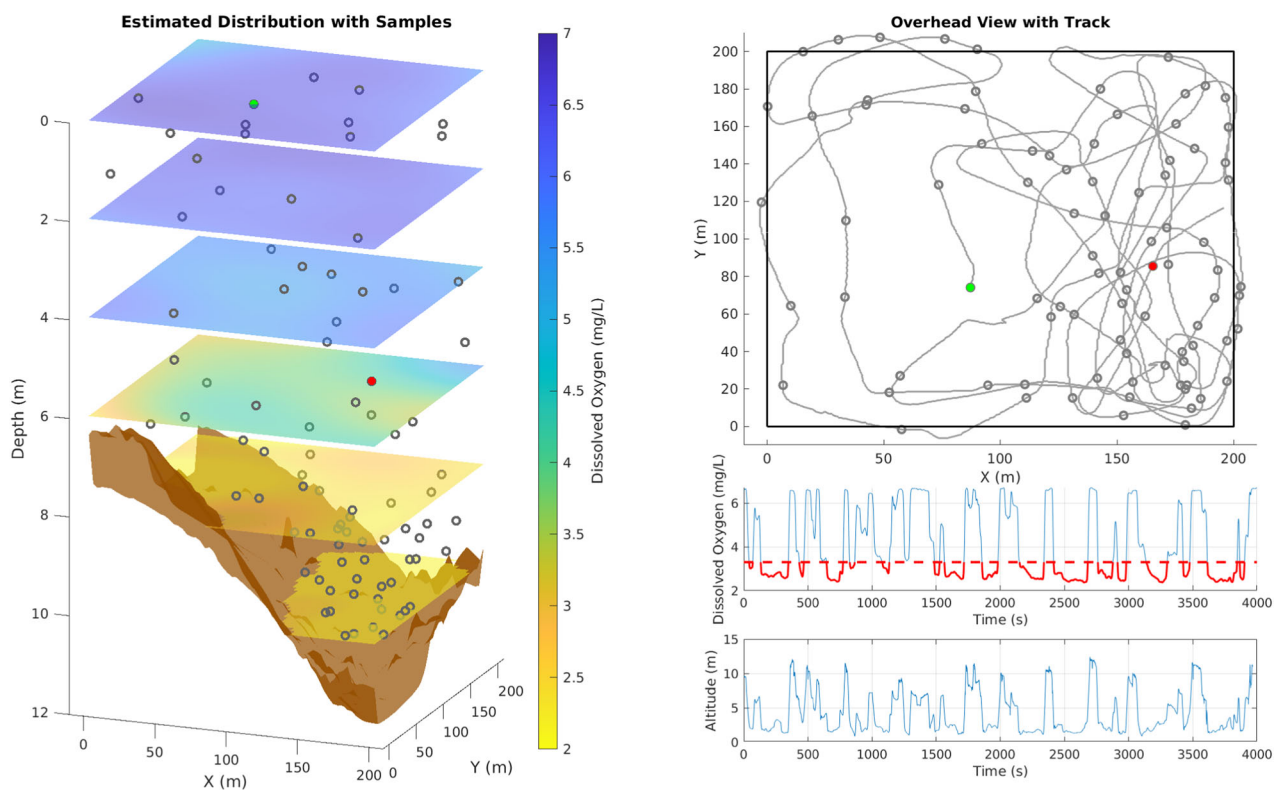


FIGURE 14 Results for the field experiment performed in TA2 using in situ dissolved oxygen measurements. The left image shows the estimated Gaussian process model with sampling points in 3D. The measured bathymetry of the test area is shown in brown. The top right plot shows a bird's-eye view of the autonomous underwater vehicle trajectory. The time series plots in the bottom right show the DO measurements, where measurements below the normalized region of interest range (22nd percentile) are highlighted in red, and the vehicle altitude [Color figure can be viewed at wileyonlinelibrary.com]

typically range between 0 and 9 mg/L. To evaluate sampling locations using the GP-UCB, measurements were normalized based on this range to be between 0 and 1. The dimensions of TA2 are 200 m × 200 m with depth ranges between 6 m on the western edge and 12 m on the eastern edge. This test site lies within a channel of the Severn River that is characterized by large changes in bathymetry and greater than average depths. These characteristics are of interest for measuring dissolved oxygen, as pockets of significant depth at the center of the channel have a tendency to produce hypoxic regions (Muller et al., 2016). Algorithm parameters were identical to those in Table 3 with the exception that $t_f = 4000$ s, $\delta_t = 40$ s, and $N = 85$ to accommodate the smaller sampling volume.

The results obtained for the field experiment in TA2 are shown in Figure 14 while the performance metrics are summarized in Table 4 (with the exception that GP RMSE could not be calculated in the absence of the ground truth dissolved oxygen field). The environmental distribution shown by the GP model in Figure 14 is an estimate based on the collected dissolved oxygen measurements which range between approximately 2.3 mg/L at the bottom of the channel along the eastern edge and 7 mg/L at the surface. The first takeaway from this experiment is that hypoxic conditions (<2 mg/L) were not present at TA2 during the data collection event. This is primarily due to the fact that the field experiment was performed in early October, while hypoxic zones are typically most prevalent in July and August. The observed dissolved oxygen levels, however, were still well below the 3 mg/L threshold that is required to support most marine life including deep-water fish, crabs, and oysters (Uphoff et al., 2011). Further, the observation that these measurements occurred in early October means that the region surrounding TA2 would be highly susceptible to hypoxia during peak summer months.

In the absence of hypoxic zones, an alternative evaluation of the GPAS effectiveness is to consider the samples collected in relation to the *available* measurement range at the test site. We continue to use the previous definition of an ROI as values below a normalized threshold of 0.22 (i.e., the 22nd percentile of the available measurement range), but now in relation to the available measurement range of 2.3–7 mg/L. According to this evaluation, the AUV spent 1946 s within ROIs, accounting for approximately 48% of the total mission time. This translates to an RMR value of 2.4 when considering the volume of the ROI based on the collected data. In other words, the GPAS algorithm performed as expected with the information that was available during collection. This is evidenced in Figure 14 by the concentration of samples collected along the eastern edge of the sampling area, especially towards the end of the sampling mission.

7 | CONCLUSION

This paper presented a GP-based adaptive sampling method with the goal of fully exploring an environment workspace while also localizing and heavily sampling ROIs that exhibit extreme sensory

measurements and high spatially correlated variability when compared to the surrounding area. The proposed algorithm automatically prioritizes a trade-off between exploration and exploitation and dynamically adjusts the sampling density according to sensor measurements. Two informative path planning methods (one based on branch and bound techniques and one based on cross-entropy optimization) were presented to select future sampling locations in a 3D environment while also considering the motion constraints of the system over the planning horizon. Both the adaptive sampling algorithm and the path planning sub-routines were applied to an underactuated AUV and their performance against area-coverage survey paths was studied in simulation. Further, two preliminary field experiments were performed in the Severn River, MD based on both a virtual environmental distribution and a distribution based on in situ dissolved oxygen measurements. Results from both the simulation studies and field experiments show that the proposed methodology is able to be fielded on operational platforms and preferentially collect measurements in desired ROIs without sacrificing the overall model fidelity of the full sampling area.

There are several areas of future work that are natural extensions to the methodology presented in this paper. With regard to adaptive sampling for AUV applications, planning efficiency could be improved by investigating more intelligent surfacing strategies based on communication objectives or uncertainty thresholds. Surfacing only when necessary would limit the number of disruptions when collecting samples at depth. In applications where there is not a fixed mission duration, planning efficiency could also be improved by considering the additional desire to minimize travel distance or mission time while maximizing information gain. Another improvement to the AUV sampling would be to incorporate knowledge of tidal currents into the path planning process. Smaller platforms such as the AUV used in this study are particularly susceptible to even moderate tidal currents which can lead to compromised maneuvering. Exploiting currents when performing vehicle planning could be beneficial when performing sampling missions over a larger area. Finally, extensions of this study are planned to incorporate multiple sampling vehicles within a heterogeneous robot team. Robot teams are able to exploit the strengths of different platforms to collect the desired measurement profile over a larger sampling area.

ACKNOWLEDGMENTS

This study was partially supported by NSF grant #1302360 and USDA grant #2017-67007-26154. The authors would also like to thank Prof. Louis Whitcomb and Zachary Harris for their help in configuring the vehicle and aiding in field experiments.

ORCID

Paul Stankiewicz  <http://orcid.org/0000-0002-1617-3964>

Yew T. Tan  <http://orcid.org/0000-0003-2158-9910>

REFERENCES

- Arora, S., & Scherer, S. (2017). Randomized algorithm for informative path planning with budget constraints. In *2017 IEEE International Conference on Robotics and Automation (ICRA)* (pp. 4997–5004). IEEE.
- Binney, J., & Sukhatme, G. S. (2012). Branch and bound for informative path planning. In *2012 IEEE International Conference on Robotics and Automation (ICRA)* (pp. 2147–2154). IEEE.
- Bottarelli, L., Bicego, M., Blum, J., & Farinelli, A. (2019). Orienteering-based informative path planning for environmental monitoring. *Engineering Applications of Artificial Intelligence*, 77, 46–58.
- Bull, A. D. (2011). Convergence rates of efficient global optimization algorithms. *Journal of Machine Learning Research*, 12(Oct), 2879–2904.
- Cao, N., Low, K. H., & Dolan, J. M. (2013). Multi-robot informative path planning for active sensing of environmental phenomena: A tale of two algorithms. In *Proceedings of the 2013 international conference on Autonomous agents and multi-agent systems* (pp. 7–14). International Foundation for Autonomous Agents and Multiagent Systems.
- Contal, E., Perchet, V., & Vayatis, N. (2014). Gaussian process optimization with mutual information. In *International Conference on Machine Learning* (pp. 253–261).
- Cui, R., Li, Y., & Yan, W. (2016). Mutual information-based multi-auv path planning for scalar field sampling using multidimensional RRT. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 46(7), 993–1004.
- Das, J., Harvey, J., Py, F., Vathsangam, H., Graham, R., Rajan, K., & Sukhatme, G. S. (2013). Hierarchical probabilistic regression for auv-based adaptive sampling of marine phenomena. In *2013 IEEE International Conference on Robotics and Automation (ICRA)* (pp. 5571–5578). IEEE.
- Das, J., Py, F., Harvey, J. B., Ryan, J. P., Gellene, A., Graham, R., Caron, D. A., Rajan, K., & Sukhatme, G. S. (2015). Data-driven robotic sampling for marine ecosystem monitoring. *The International Journal of Robotics Research*, 34(12), 1435–1452.
- Das, J., Py, F., Maughan, T., O'Reilly, T., Messié, M., Ryan, J., Sukhatme, G. S., & Rajan, K. (2012). Coordinated sampling of dynamic oceanographic features with underwater vehicles and drifters. *The International Journal of Robotics Research*, 31(5), 626–646.
- De Boer, P.-T., Kroese, D. P., Mannor, S., & Rubinstein, R. Y. (2005). A tutorial on the cross-entropy method. *Annals of Operations Research*, 134(1), 19–67.
- Dunbabin, M., & Marques, L. (2012). Robots for environmental monitoring: Significant advancements and applications. *IEEE Robotics & Automation Magazine*, 19(1), 24–39.
- Flaspohler, G., Preston, V., Michel, A. P., Girdhar, Y., & Roy, N. (2019). Information-guided robotic maximum seek-and-sample in partially observable continuous environments. *IEEE Robotics and Automation Letters*, 4(4), 3782–3789.
- Frew, E. W., Elston, J., Argrow, B., Houston, A., & Rasmussen, E. (2012). Sampling severe local storms and related phenomena: Using unmanned aircraft systems. *IEEE Robotics & Automation Magazine*, 19(1), 85–95.
- Galceran, E., & Carreras, M. (2013). A survey on coverage path planning for robotics. *Robotics and Autonomous Systems*, 61(12), 1258–1276.
- Gertler, M., & Hagen, G. R. (1967). *Standard equations of motion for submarine simulation*. Technical report, David w Taylor Naval Ship Research and Development Center, Bethesda, MD.
- Gotovos, A., Casati, N., Hitz, G., & Krause, A. (2013). Active learning for level set estimation. In *Twenty-Third International Joint Conference on Artificial Intelligence*.
- Hagy, J. D., Boynton, W. R., Keefe, C. W., & Wood, K. V. (2004). Hypoxia in chesapeake bay, 1950–2001: Long-term change in relation to nutrient loading and river flow. *Estuaries*, 27(4), 634–658.
- Henkart, P. (2012). *Severn riverkeeper water quality monitoring program 2012 report*. Technical report, Severn Riverkeeper.
- Hitz, G., Galceran, E., Garneau, M.-É., Pomerleau, F., & Siegwart, R. (2017). Adaptive continuous-space informative path planning for online environmental monitoring. *Journal of Field Robotics*, 34(8), 1427–1449.
- Hollinger, G. A., & Sukhatme, G. S. (2014). Sampling-based robotic information gathering algorithms. *The International Journal of Robotics Research*, 33(9), 1271–1287.
- Karaman, S., & Frazzoli, E. (2011). Sampling-based algorithms for optimal motion planning. *The International Journal of Robotics Research*, 30(7), 846–894.
- Kawaguchi, K., Kaelbling, L. P., & Lozano-Pérez, T. (2015). Bayesian optimization with exponential convergence. In *Advances in Neural Information Processing Systems* (pp. 2809–2817).
- Kemna, S., & Sukhatme, G. S. (2018). Surfacing strategies for multi-robot adaptive informative sampling with a surface-based data hub. In *OCEANS 2018 MTS/IEEE Charleston* (pp. 1–10). IEEE.
- Kemp, W. M., Boynton, W. R., Adolf, J. E., Boesch, D. F., Boicourt, W. C., Brush, G., Cornwell, J. C., Fisher, T. R., Glibert, P. M., Hagy, J. D., Harding, L. W., Houde, E., Kimmel, D., Miller, W. D., Newell, R. I. E., Roman, M. R., Smith, E. M., & Stevenson, J. C. (2005). Eutrophication of chesapeake bay: Historical trends and ecological interactions. *Marine Ecology Progress Series*, 303, 1–29.
- Kobilarov, M. (2012). Cross-entropy motion planning. *The International Journal of Robotics Research*, 31(7), 855–871.
- Krause, A., Singh, A., & Guestrin, C. (2008). Near-optimal sensor placements in gaussian processes: Theory, efficient algorithms and empirical studies. *Journal of Machine Learning Research*, 9(Feb), 235–284.
- Lawrance, N. R., Chung, J. J., & Hollinger, G. A. (2017). Fast marching adaptive sampling. *IEEE Robotics and Automation Letters*, 2(2), 696–703.
- Leonard, N. E., Paley, D. A., Davis, R. E., Fratantoni, D. M., Lekien, F., & Zhang, F. (2010). Coordinated control of an underwater glider fleet in an adaptive ocean sampling field experiment in Monterey bay. *Journal of Field Robotics*, 27(6), 718–740.
- Leonard, N. E., Paley, D. A., Lekien, F., Sepulchre, R., Fratantoni, D. M., & Davis, R. E. (2007). Collective motion, sensor networks, and ocean sampling. *Proceedings of the IEEE*, 95(1), 48–74.
- Li, A. Q., Rekleitis, I., Manjanna, S., Kakodkar, N., Hansen, J., Dudek, G., Bobadilla, L., Anderson, J., & Smith, R. N. (2016). Data correlation and comparison from multiple sensors over a coral reef with a team of heterogeneous aquatic robots. In *International Symposium on Experimental Robotics* (pp. 717–728). Springer.
- Lim, Z. W., Hsu, D., & Lee, W. S. (2016). Adaptive informative path planning in metric spaces. *The International Journal of Robotics Research*, 35(5), 585–598.
- Liu, S., Yue, Y., & Krishnan, R. (2015). Non-myopic adaptive route planning in uncertain congestion environments. *IEEE Transactions on Knowledge and Data Engineering*, 27(9), 2438–2451.
- Low, K. H., Dolan, J. M., & Khosla, P. K. (2009). Information-theoretic approach to efficient adaptive path planning for mobile robotic environmental sensing. *ICAPS*, 233–240.
- Ma, K.-C., Liu, L., Heidarrson, H. K., & Sukhatme, G. S. (2018). Data-driven learning and planning for environmental sampling. *Journal of Field Robotics*, 35(5), 643–661.
- Ma, K.-C., Liu, L., & Sukhatme, G. S. (2016). An information-driven and disturbance-aware planning method for long-term ocean monitoring. *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, (pp. 2102–2108). IEEE.
- Manjanna, S., & Dudek, G. (2017). Data-driven selective sampling for marine vehicles using multi-scale paths. *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (pp. 6111–6117). IEEE.

- Manjanna, S., Hansen, J., Li, A. Q., Rekleitis, I., & Dudek, G. (2017). Collaborative sampling using heterogeneous marine robots driven by visual cues. *2017 14th Conference on Computer and Robot Vision (CRV)* (pp. 87–94). IEEE.
- Manjanna, S., Li, A. Q., Smith, R. N., Rekleitis, I., & Dudek, G. (2018). Heterogeneous multi-robot system for exploration and strategic water sampling. *2018 IEEE International Conference on Robotics and Automation (ICRA)* (pp. 1–8). IEEE.
- Manjanna, S., Van Hoof, H., & Dudek, G. (2018). Policy search on aggregated state space for active sampling. *International Symposium on Experimental Robotics* (pp. 211–221). Springer.
- Mannor, S., Rubinstein, R. Y., & Gat, Y. (2003). The cross entropy method for fast policy search. In *Proceedings of the 20th International Conference on Machine Learning (ICML-03)* (pp. 512–519).
- Marchant, R., & Ramos, F. (2014). Bayesian optimisation for informative continuous path planning. In *2014 IEEE International Conference on Robotics and Automation (ICRA)* (pp. 6136–6143).
- McMahon, J., Yetkin, H., Wolek, A., Waters, Z. J., & Stilwell, D. J. (2017). Towards real-time search planning in subsea environments. *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (pp. 87–94). IEEE.
- Miller, M., Ketcham, I., & Guay, T. (2019). *A mid-summer 2019 report: Tracking the severn river's dead zone*. Technical report, Severn River Association.
- Mukhopadhyay, S., Wang, C., Patterson, M., Malisoff, M., & Zhang, F. (2014). Collaborative autonomous surveys in marine environments affected by oil spills. In *Cooperative Robots and Sensor Networks 2014* (pp. 87–113). Springer.
- Muller, A. C., Muller, D. L., & Muller, A. (2016). Resolving spatiotemporal characteristics of the seasonal hypoxia cycle in shallow estuarine environments of the Severn river and south river, MD, Chesapeake bay, USA. *Heliyon*, 2(9), e00157.
- Munafò, A., Simetti, E., Turetta, A., Caiti, A., & Casalino, G. (2011). Autonomous underwater vehicle teams for adaptive ocean sampling: A data-driven approach. *Ocean Dynamics*, 61(11), 1981–1994.
- Murphy, K. P. (2012). *Machine learning: A probabilistic perspective*. MIT Press.
- Muscato, G., Bonaccorso, F., Cantelli, L., Longo, D., & Melita, C. D. (2012). Volcanic environments: robots for exploration and measurement. *IEEE Robotics & Automation Magazine*, 19(1), 40–49.
- Mysorewala, M. F., Popa, D. O., & Lewis, F. L. (2009). Multi-scale adaptive sampling with mobile agents for mapping of forest fires. *Journal of Intelligent and Robotic Systems*, 54(4), 535.
- Ouyang, R., Low, K. H., Chen, J., & Jaillet, P. (2014). Multi-robot active sensing of non-stationary gaussian process-based environmental phenomena. In *Proceedings of the 2014 international conference on Autonomous agents and multi-agent systems* (pp. 573–580). International Foundation for Autonomous Agents and Multiagent Systems.
- Powell, W. B. (2007). *Approximate dynamic programming: Solving the curses of dimensionality* (Vol. 703). John Wiley & Sons.
- Quigley, M., Conley, K., Gerkey, B., Faust, J., Foote, T., Leibs, J., Wheeler, R., & Ng, A. Y. (2009). Ros: An open-source robot operating system. *ICRA workshop on open source software* (Vol. 3, p. 5). Kobe, Japan.
- Rasmussen, C. E., & Williams, C. K. (2006). *Gaussian process for machine learning*. MIT Press.
- Reid, W., Fitch, R., Göktoğan, A. H., & Sukkarieh, S. (2020). Sampling-based hierarchical motion planning for a reconfigurable wheel-on-leg planetary analogue exploration rover. *Journal of Field Robotics*, 37(5), 786–811.
- Smith, R. N., Schwager, M., Smith, S. L., Jones, B. H., Rus, D., & Sukhatme, G. S. (2011). Persistent ocean monitoring with underwater gliders: Adapting sampling resolution. *Journal of Field Robotics*, 28(5), 714–741.
- Srinivas, N., Krause, A., Kakade, S. M., & Seeger, M. (2009). Gaussian process optimization in the bandit setting: No regret and experimental design. arXiv preprint arXiv:0912.3995.
- Tan, Y. T., Kunapareddy, A., & Kobilarov, M. (2018). Gaussian process adaptive sampling using the cross-entropy method for environmental sensing and monitoring. In *2018 IEEE International Conference on Robotics and Automation (ICRA)* (pp. 6220–6227). IEEE.
- Uphoff Jr, J. H., McGinty, M., Lukacovic, R., Mowrer, J., & Pyle, B. (2011). Impervious surface, summer dissolved oxygen, and fish distribution in Chesapeake bay subestuaries: Linking watershed development, habitat conditions, and fisheries management. *North American Journal of Fisheries Management*, 31(3), 554–566.
- Wang, D., Lermusiaux, P. F., Haley, P. J., Eickstedt, D., Leslie, W. G., & Schmidt, H. (2009). Acoustically focused adaptive sampling and on-board routing for marine rapid environmental assessment. *Journal of Marine Systems*, 78, S393–S407.
- Wang, Z., Shakibi, B., Jin, L., & de Freitas, N. (2014). *Bayesian multi-scale optimistic optimization*.

How to cite this article: Stankiewicz P, Tan YT, Kobilarov M. Adaptive sampling with an autonomous underwater vehicle in static marine environments. *J Field Robotics*. 2020;1–26. <https://doi.org/10.1002/rob.22005>

APPENDIX A: BB-IPP BRANCHING REACHABILITY ANALYSIS

This section further examines the reachability characteristics of the branching procedure outlined in Section 4.1.1. The branching procedure is based on sampling motion primitives defined in velocity space; however, since the objective function is defined over the position workspace \mathcal{W} , a key requirement is the ability to explore that workspace with high probability to discover regions of high information content, especially since early stages greatly affect the sampling positions of later stages. Therefore, we establish a relationship between the velocity-space sampling process and the resulting coverage of \mathcal{W} to ensure that the branching procedure does not omit key areas of the reachable workspace.

As a guideline, our goal is to ensure that with high probability we can visit any spatial region with a given radius R_{min} over the reachable space for a given planning horizon. A typical choice of this radius corresponds to the vehicle's turning radius of approximately $R_{min} = 15\text{m}$. Further, let $\mathcal{W}_{x_0, T} \subset \mathcal{W}$ denote the reachable workspace starting at some state x_0 over a horizon T and let the set of samples generated by the search algorithm be denoted by $\widehat{\mathcal{W}}_{x_0, T} \subset \mathcal{W}_{x_0, T}$. We require that with high confidence $R(\widehat{\mathcal{W}}_{x_0, T}) < R_{min}$, where

$$R(\widehat{\mathcal{W}}_{x_0, T}) = \sup_{p \in \mathcal{W}_{x_0, T}} \left(\min_{p' \in \widehat{\mathcal{W}}_{x_0, T}} \|p - p'\| \right)$$

is the radius of the largest ball that does not contain a future sampling location over the reachable workspace. We limit the analysis to spatial positions in the horizontal plane due to the significantly smaller length scale of the workspace depth component. Since the distribution of $R(\widehat{\mathcal{W}}_{x_0, T})$ is difficult to compute analytically, that is, it depends on a nonlinear transformation of a probability density function over the

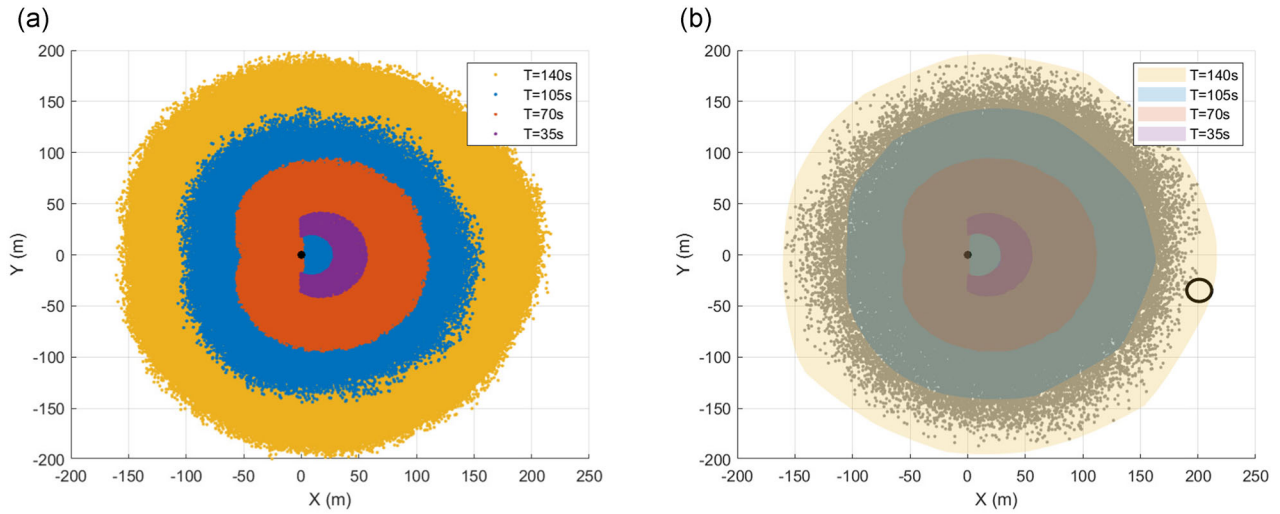


FIGURE A1 (a) The approximate reachable workspace $\mathcal{W}_{x_0,T}$ over different planning horizons for $v_x \in [0.8, 1.6]$ m/s and $\omega_z \in [-0.09, 0.09]$ rad/s. The nonzero minimum speed means that the area immediately in front of the vehicle is not reachable for the shorter horizons. (b) Candidate sampling locations $\widehat{\mathcal{W}}_{x_0,T}$ (gray dots) evaluated for M trajectories over $T = 140$ s. The largest empty circle for the samples is shown on the right

primitive sampling parameters, we instead numerically approximate its expected value and its confidence interval through offline sampling. This is accomplished by executing multiple simulations resulting in different $R(\widehat{\mathcal{W}}_{x_0,T})$ employed for sample-based confidence-interval analysis.

In each simulation, sampled locations are generated using M^H trajectories over a given planning horizon. Figure A1 shows an example of approximated reachable spaces and the largest empty ball for a set of future sampling locations evaluated over $T = 140$ s. Additionally, Figure A2a shows statistics for the maximum empty ball calculated over 30 trials for various combinations of M^H and T . These

results show that the parameters ultimately selected for the BB-IPP algorithm satisfy the desired exploration criteria with high probability. Specifically, after 100 simulations we obtain a sample mean for $R(\widehat{\mathcal{W}}_{x_0,T})$ of 11.47 m and sample SD of 0.98 m. This corresponds to a confidence interval of

$$Cl_R(\widehat{\mathcal{W}}_{x_0,T}) = [11.21, 11.67].$$

Therefore, based on the chosen sampling parameters, with high confidence the branching procedure will not miss a region with radius $R_{\min} = 15$ m. Figure A2b further emphasizes adequate coverage

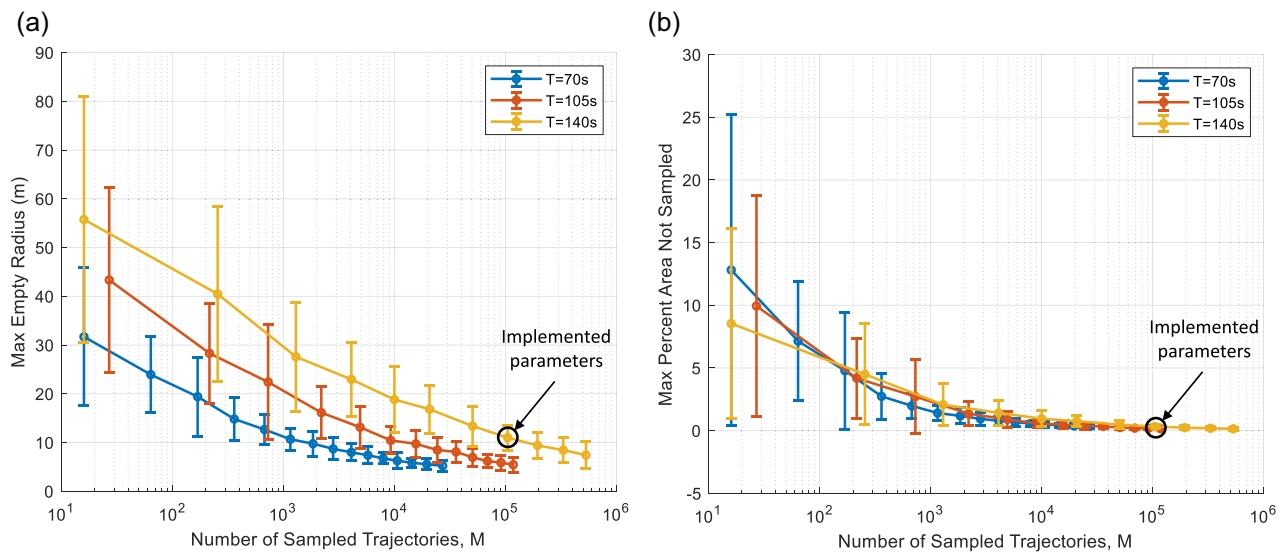


FIGURE A2 Monte Carlo analysis of the maximum empty radius (a) and maximum percentage of empty reachable space (b) as a function of the number of sampled trajectories for various planning horizons. Each data point shows the mean calculated over 30 trials with the error bars representing two SDs. The parameter combination implemented for the branch and bound informative path planning algorithm is highlighted

TABLE B1 Mean and SD of the evaluation metrics (Table 2) for parameter variations of the branch and bound informative path planning algorithm

Parameters (T in s)		Reward per segment	Planning time (s)	GP RMSE	GP variance	Percent mission in ROI
$T = 105$	$\alpha = 0.9$	1.28 ± 0.11	0.11 ± 0.006	0.024 ± 0.011	0.099 ± 0.082	11.6 ± 10.0
	$\alpha = 0.95$	1.31 ± 0.12	0.11 ± 0.007	0.026 ± 0.023	0.113 ± 0.090	13.1 ± 5.4
	$\alpha = 1.0$	1.35 ± 0.11	0.13 ± 0.016	0.026 ± 0.018	0.114 ± 0.240	13.2 ± 8.7
$T = 140$	$\alpha = 0.9$	1.31 ± 0.13	0.12 ± 0.006	0.025 ± 0.023	0.100 ± 0.088	12.5 ± 6.1
	$\alpha = 0.95$	1.37 ± 0.09	0.12 ± 0.007	0.024 ± 0.013	0.095 ± 0.08	13.2 ± 8.9
	$\alpha = 1.0$	1.40 ± 0.15	0.22 ± 0.064	0.026 ± 0.023	0.116 ± 0.093	12.0 ± 3.7
$T = 175$	$\alpha = 0.9$	1.31 ± 0.09	0.13 ± 0.006	0.025 ± 0.014	0.108 ± 0.161	10.4 ± 5.2
	$\alpha = 0.95$	1.36 ± 0.09	0.14 ± 0.009	0.026 ± 0.018	0.120 ± 0.234	12.4 ± 8.7
	$\alpha = 1.0$	1.41 ± 0.08	0.59 ± 0.295	0.024 ± 0.016	0.101 ± 0.071	11.8 ± 4.2

Note: Fifty simulations were performed for each parameter combination. The bolded values are those that were chosen from the parameter tuning for further simulation and field experiments.

Abbreviations: GP, Gaussian process; RMSE, root-mean-square error; ROI, region of interest.

in the position space by showing that the largest empty ball is actually a very small percentage of the total area of the reachable space. Based on this analysis and further studies performed in Appendix B, we pick parameters for T , M , and \mathcal{Z} to achieve the desired reachability over a maximum planning horizon while also balancing computational expense.

APPENDIX B: PATH PLANNING PARAMETER TUNING

This section provides additional detailed results from Monte Carlo simulations used to tune the informative path planner parameters of Section 4. Nominal parameter values for both planners that were not varied for the analysis are presented in Table 3. While the value of δ in Equation (16) was set to 0.9, it should be noted that variations of this parameter may be insignificant as shown in Contal et al. (2014). Fifty simulations were executed for each parameter combination.

B1 | BB-IPP parameter tuning

Performance of the BB-IPP method was evaluated by varying the planning horizon T and the reward upper bound percentile α . Table B1 shows the resulting performance metrics for various combinations of these parameters. These results reveal that both the mean reward per trajectory segment and the mean planning time increase as the values of T and α increase. This is expected, as longer planning horizons and larger values of the reward upper bound percentile threshold correlate with evaluating a greater number of possible trajectories. The remaining metrics of Table B1 including GP RMSE, GP variance, and mission time in

ROIs do not indicate a strong relationship with the values of the BB-IPP parameters.

B2 | CE-IPP parameter tuning

For the CE-IPP method, performance was evaluated by varying the planning horizon T , the elite set percentile threshold ρ , and the number of trajectory samples generated per iteration M . The elite set percentile threshold ρ was chosen for study because it determines the rarity of trajectories that are used to learn subsequent importance samplers, thereby increasing the frequency of generating informative paths. Table B2 shows the resulting performance metrics for various combinations of the CE-IPP parameters. These results exhibit the expected trend that the mean planning time increases as the values of T (longer trajectories) and M (greater number of possible trajectories) are increased. Surprisingly, however, while the mean reward per segment generally increases with longer planning horizons, greater values of M seem to result in a decrease in the mean reward per segment. This could possibly be explained by the fact that larger values of M result in a greater number of trajectories that compose the elite set of samples ϵ_j . In turn, the $(1 - \rho)$ -percentile of this set \hat{y}_j would be higher if the majority of the samples do not produce large rewards. This is supported by a slight increase in the mean reward per segment and mean mission % in ROIs as the value of ρ decreases. The remaining metrics of Table B2 including GP RMSE and GP variance do not indicate a strong relationship with the values of the CE-IPP parameters.

TABLE B2 Mean and SD of the evaluation metrics (Table 2) for parameter variations of the CE-IPP algorithm

Parameters (T in s)			Reward per segment	Planning time (s)	GP RMSE	GP variance	Percent mission in ROI
$T = 105$	$M = 300$	$\rho = 0.05$	1.40 ± 0.09	0.59 ± 0.05	0.029 ± 0.018	0.114 ± 0.010	17.8 ± 10.7
		$\rho = 0.1$	1.40 ± 0.10	0.58 ± 0.05	0.026 ± 0.014	0.184 ± 0.208	18.2 ± 10.0
		$\rho = 0.2$	1.39 ± 0.12	0.58 ± 0.05	0.027 ± 0.014	0.123 ± 0.238	17.7 ± 12.3
	$M = 500$	$\rho = 0.05$	1.40 ± 0.13	1.19 ± 0.13	0.024 ± 0.008	0.105 ± 0.100	20.5 ± 11.3
		$\rho = 0.1$	1.34 ± 0.08	1.18 ± 0.12	0.036 ± 0.021	0.110 ± 0.092	19.6 ± 12.1
		$\rho = 0.2$	1.33 ± 0.14	1.18 ± 0.12	0.022 ± 0.007	0.111 ± 0.211	18.3 ± 8.8
$T = 140$	$M = 300$	$\rho = 0.05$	1.41 ± 0.12	0.96 ± 0.05	0.026 ± 0.015	0.108 ± 0.120	20.88 ± 8.7
		$\rho = 0.1$	1.46 ± 0.09	0.96 ± 0.05	0.025 ± 0.012	0.106 ± 0.126	20.9 ± 11.0
		$\rho = 0.2$	1.40 ± 0.21	0.96 ± 0.05	0.032 ± 0.023	0.115 ± 0.242	22.3 ± 14.8
	$M = 500$	$\rho = 0.05$	1.35 ± 0.11	1.95 ± 0.12	0.022 ± 0.006	0.093 ± 0.070	19.4 ± 8.9
		$\rho = 0.1$	1.40 ± 0.13	1.94 ± 0.13	0.022 ± 0.008	0.093 ± 0.067	18.5 ± 7.4
		$\rho = 0.2$	1.32 ± 0.12	1.95 ± 0.12	0.022 ± 0.006	0.098 ± 0.090	17.8 ± 12.1
$T = 175$	$M = 300$	$\rho = 0.05$	1.44 ± 0.09	1.21 ± 0.06	0.022 ± 0.008	0.105 ± 0.088	21.7 ± 12.7
		$\rho = 0.1$	1.44 ± 0.10	1.22 ± 0.06	0.023 ± 0.011	0.104 ± 0.083	20.7 ± 11.0
		$\rho = 0.2$	1.47 ± 0.11	1.22 ± 0.06	0.023 ± 0.014	0.101 ± 0.077	19.1 ± 12.5
	$M = 500$	$\rho = 0.05$	1.40 ± 0.12	2.39 ± 0.12	0.021 ± 0.008	0.101 ± 0.083	19.2 ± 9.2
		$\rho = 0.1$	1.38 ± 0.09	2.40 ± 0.12	0.028 ± 0.022	0.133 ± 0.092	18.7 ± 8.5
		$\rho = 0.2$	1.32 ± 0.12	2.40 ± 0.12	0.022 ± 0.011	0.103 ± 0.088	18.9 ± 9.7

Note: Fifty simulations were performed for each parameter combination. The bolded values are those that were chosen from the parameter tuning for further simulation experiments.

Abbreviations: GP, Gaussian process; RMSE, root-mean-square error; ROI, region of interest.