# Using expected improvement of gradients for robotic exploration of ocean salinity fronts

**André Julius Hovd Olaisen, Yaolin Ge and Jo Eidsvik**
**Department of Mathematical Sciences, NTNU,**
**7491 Trondheim, Norway**
***Corresponding author: andre.j.h.olaisen@ntnu.no**

**Abstract**

We study opportunities for dynamic sampling designs in spatio-temporal random field. Considering a situation with a robotic agent, we develop an algorithm that enables autonomous exploration of spatial domains with large gradients. The modeling assumptions rely on a spatio-temporal Gaussian random field, which means that the directional derivatives of the field are Gaussian distributed. Using computational tricks at the dimension of relatively sparse data, the robot updates its on-board Gaussian random field model in real-time. Moreover, it computes the expected improvement in directional derivatives along a set of possible paths in a spider-leg search space to choose intelligent exploration designs over time stages. We study statistical properties of this suggested approach in a simulation study, where we compare the design criterion with several other viable design selection criteria. The new algorithm is embedded on an autonomous underwater vehicle which is deployed for characterizing a river plume frontal system in a Norwegian fjord. Using expected improvement for the salinity field derivatives, the vehicle successfully sampled the river front for more than two hours without intervention.

*Keywords: Expected improvement; Gaussian Random Field; Spatial design; Robotics; Oceanography*

## 1. Introduction

Inspired by new sensor technology and small-size computing units, there is currently a drive to develop intelligent monitoring systems. This development is pushed by engineers and multi-disciplinary visions on how to put innovative solutions into practical use. Recent examples include internet-of-things for smart sensor networks monitoring air pollution (Dhingra et al., 2019), embedded systems and AI for agriculture (Shadrin et al., 2019), robotic systems for understanding environmental processes (Dunbabin and Marques, 2012) and cyber-physical systems that can re-configure themselves for ecological monitoring (Schranz et al., 2021).

The capabilities of such embedded systems can clearly be improved by leveraging knowledge from spatio-temporal statistics and design of experiments, see e.g. Mateu and Müller (2012), Wang et al. (2012, 2020) or Brus (2022). In doing so, one can develop more principled approaches for what, where and when to gather additional data samples, and integrate this new information in a consistent statistical modeling framework. Even so, solutions to these situations tend to be case-specific, and often of a heuristic type as the search space is too large to find the optimal solution. Impactful examples of spatial statistics and design for embedded systems include Krause et al. (2008) who studied the NP-hard problem of sensor placement using Gaussian random fields (GRFs) with the goal of finding designs that optimize the mutual information and Manohar et al. (2018) who suggested using machine learning methods to facilitate the search for constructive design patterns.

As robotic units and sensor systems often have limited computing, storage and communication capabilities, one must often simplify the modeling to guide the optimization challenge. Ideas from designing computer experiments and building surrogate models (Gramacy, 2020; Fuhg et al., 2021) are hence also highly relevant in this context of optimal spatio-temporal design and fast integration of data.

In this article we focus on an application of underwater robotics. An autonomous underwater vehicle (AUV) with onboard computing units uses a GRF surrogate model to plan where and when to explore various parts of an ocean domain. The AUV is hence a dynamic agent that can navigate to extract informative samples in an uncertain dynamic environment. In particular, the goal here is to find large derivatives of the field which are indicative of important frontal zones between different water masses. The spatio-temporal variable that we target here is ocean salinity, and via the real-world deployment, we show results of an AUV deployment characterizing salinity changes in a river plume front in a Norwegian fjord.

The main contributions of this paper are

- A spatio-temporal sampling approach with a dynamic agent searching for large derivatives in the field.
- A fast scalable algorithm for updating and planning based on GRFs and sparse observation points in the vicinity of the current position.
- A field deployment with an AUV adaptively sampling a river plume front for 2 hours and 10 minutes.

In Section 2, we describe the motivation for our work and define the necessary notation. In Section 3, we set up the required building blocks from theory on spatio-temporal GRFs and the properties of their derivatives. In Section 4, we present the method and algorithm for adaptive sampling of large directional derivatives. In Section 5, we demonstrate properties of the suggested algorithm in a simulation study. In Section 6, we show results of the AUV deployment in the Trondheim fjord in Norway. In Section 7, we provide conclusions and point to future work.

## 2. Background and notation

Fronts are important in meteorology and oceanography as they tend to be key drivers of the physical dynamic behavior, see e.g. Fedorov (1986) and Catto and Pfahl (2013). Frontal zones in the ocean are further known to be biological hot-spots that shape parts of the marine ecosystem (Belkin et al., 2009). In this paper, the spatio-temporal variable of interest is ocean salinity and its derivatives which capture the frontal zone near river plumes.

Ocean fronts can be detected from satellite data, see e.g. Hopkins et al. (2010), but this is only on the surface and not available on a cloudy day. Numerical ocean models, see e.g. Lermusiaux (2006), can mimic fronts at various scales, but even though they are incredibly useful at predicting ocean variables, they tends to be biased in space and time. AUVs have become an important tool for oceanographic in-situ sampling, and they are commonly used to detect frontal zones. These vehicles can navigate autonomously underwater and can hold a range of sensors such as a standard tool that provides salinity measurements. Many AUVs also have an onboard computer that enables for instance data assimilation in a model, and using this to adapt its trajectory and move in more interesting directions. This is important because communication is limited under water, and the full benefits of an AUV are gained only when it acts on its own as an intelligent agent.

To further motivate the detection of fronts in the ocean, we highlight a few examples. Figure 1 shows three different deployments where researchers aimed to map the frontal zone between water masses. Zhang et al. (2019) used temperature data from satellites along with in-situ AUV measurements to follow the zone of mixing cold and warm water masses in the Monterey Bay, California. Fossum et al. (2021) used an AUV to understand the frontal zone near the ice shelf in Arctic waters. Fonseca et al. (2023) compared satellite imagery and AUV samples to map the front of chlorophyll

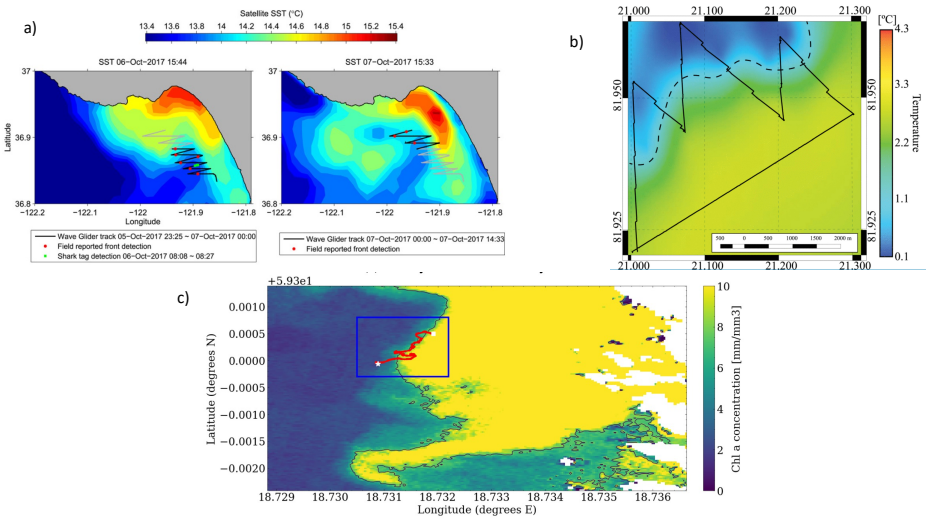



Figure 1: Examples of AUV exploration of fronts. a) Zhang et al. (2019) characterizing the Monterey Bay front between water masses using temperature information processed from satellites and that gathered by an AUV. b) Fossum et al. (2021) conducting frontal AUV sampling in water masses in the Arctic. c) Fonseca et al. (2023) showing an AUV path zig-zagging the chlorophyll front in the Baltic sea as extracted from satellite data.

in the Baltic sea. These studies attempt to find the gradient or derivative in the ocean variable of interest. In doing so, the AUV reacts to data, but none of them use spatio-temporal statistical models or approaches from spatial design, which would likely have improved the mapping performance.

We next define the notation used in our statistical model and sampling design approach. Let $\boldsymbol{s} = (s_e, s_n, s_d, t)$ be a point in space and time. Here, $s_e$, $s_n$ and $s_d$ represent east, north and depth coordinates, respectively, while $t > 0$ is a temporal index. A spatial operational domain is defined so that $(s_e, s_n, s_d) \in \mathcal{D} \subset \mathcal{R}^3$. The spatio-temporal variable of interest is denoted $x(\boldsymbol{s}) \in \mathcal{R}$. In our application this is ocean salinity.

In this paper, we are primarily interested in detecting large derivatives or changes in this spatio-temporal variable in the lateral plane close to the sea surface. In our application this would indicate ocean front zones. We define the directional difference

from location $\boldsymbol{s}$ to $\boldsymbol{s}'$ by

$$g(\boldsymbol{s}, \boldsymbol{s}') = \frac{x(\boldsymbol{s}') - x(\boldsymbol{s})}{d(\boldsymbol{s}, \boldsymbol{s}')}, \tag{1}$$

where $d(\boldsymbol{s}, \boldsymbol{s}')$ is the Euclidean distance between the two locations. Letting this distance go to 0, we obtain the field derivative at $\boldsymbol{s}$ in the direction towards $\boldsymbol{s}'$. In practice, we instead consider the distance $d(\boldsymbol{s}, \boldsymbol{s}')$ as a tuning parameter that can be specified in the context of the application and the operational constraints.

An observation made at space-time location $\boldsymbol{s}$ is denoted $y(\boldsymbol{s})$. Because of sensor noise and positioning error, this observation does not carry perfect information about the salinity. Observations from a set of sampling points $\mathcal{S} = \{\boldsymbol{s}_1, \ldots, \boldsymbol{s}_N\}$ are denoted by $\boldsymbol{y}(\mathcal{S}) = (y(\boldsymbol{s}_1), y(\boldsymbol{s}_2), ..., y(\boldsymbol{s}_N))$. AUV data are gathered sequentially. At stage $k$, the AUV gathers a batch of data size $N_k$, and we denote batch sampling locations by $\mathcal{S}_k = \{\boldsymbol{s}_1^{(k)}, \ldots, \boldsymbol{s}_{N_k}^{(k)}\}$ with associated data $\boldsymbol{y}_k = \{y(\boldsymbol{s}_1^{(k)}), \ldots, y(\boldsymbol{s}_{N_k}^{(k)})\}$. This means that we at stage $k$ have measured at $N_{1:k} = \sum_{l=1}^{k} N_l$ points. We denote the set of sampling locations by

$$\mathcal{S}_{1:k} = \{\mathcal{S}_1, \mathcal{S}_2, \ldots, \mathcal{S}_{k-1}, \mathcal{S}_k\}, \quad k = 1, 2, \ldots, \tag{2}$$

with associated salinity measurements

$$\boldsymbol{y}(\mathcal{S}_{1:k}) = \boldsymbol{y}_{1:k} = \left(\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_{k-1}, \boldsymbol{y}_k\right), \quad k = 1, 2, \ldots. \tag{3}$$

At each stage $k$, the agent computes expected rewards for staying on the same trajectory and for changing its path to another direction. Higher rewards are attained for design directions that have large expected derivatives. Design paths form transects consisting of $M_k^j$ new locations in a set $\mathcal{P}_k^j = \{\boldsymbol{p}_{k,l}^j; j = 1, \ldots, J, l = 1, \ldots, M_k^j\}$ with single locations $\boldsymbol{p}_{k,l}^j \in \mathcal{D}$. For the number of directions $J$, we use a spider-leg formation in the lateral domain. The number of transect points $M_k^j$ is fixed at all stages and for all designs, except at boundary locations. The spacing between single locations along each design direction is also fixed, and it is determined by the agent´s speed

and sampling frequency. In doing so, measurements and plan evaluations are easily comparable.

Figure 2 shows the situation with an agent path consisting of 5 stages. At the current location (blue circle), it makes a decision about where to go next for stage 6. The agent is not constrained to travel the entire segment of length $M_j$. Instead, it

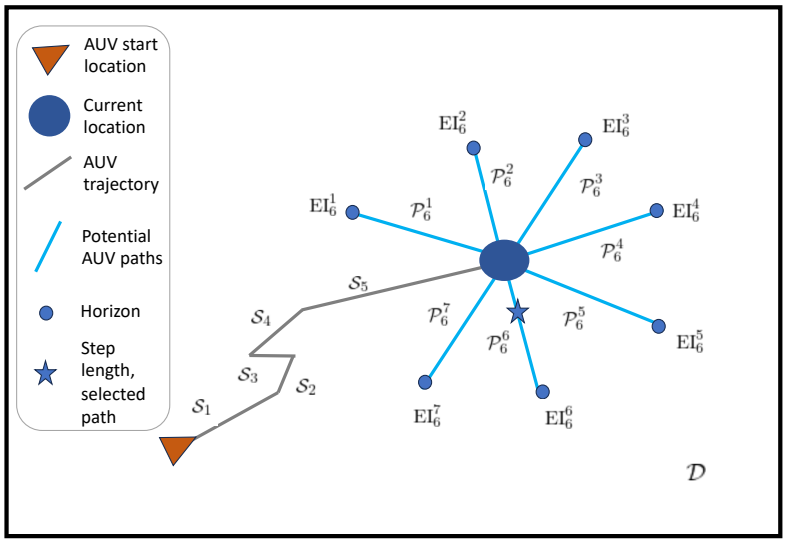



Figure 2: Illustration of an AUV trajectory made up of segments over 5 earlier stages. At the current location, the AUV will choose one of the 7 possible designs. The design selection criteria is expected improvement (EI) for the salinity derivative.

conducts new design evaluations after moving one step-length down the best segment. In this illustration, design 6 is selected and the agent moves a step-length in this direction (marked by a star).

In summary, bringing the model and design choices back to the context of AUV sampling, we assume that:

- The AUV moves significantly faster than the ocean phenomenon develops over time, and this means that the AUV is able to detect changes in space. Hence we focus on differences in space, and not time.

- Salinity changes are often most difficult to characterize in the lateral domain, so without loss of generality, we focus on differences in the east, north plane, assuming the AUV is at a fixed depth (set to 1 meter in the field deployment).
- The AUV is able to maintain a nearly constant velocity. Because the sensors are sampling at a constant frequency, the relatively close locations $\boldsymbol{s}$ and $\boldsymbol{s}'$ are at the same distance during the operation.

Note that the suggested approach does not rely on the usual concepts of a predefined waypoint graph or grid for the path-planning. Instead, data points and variables are allocated to continuous space-time locations, and this occurs when the design criterion is computed according to a spider leg design. Hence, the discretization occurs only along transect lines, and it is formed during the mission, not before the deployment starts. One benefit of this approach is that the AUV maintains a model with relatively few points compared to a waypoint graph, but we can still have a high level of detail close to where we sample. In practice, one might miss the phenomenon by placing a strict waypoint graph onboard the AUV model. Here, the AUV is more free to follow where it is most interesting to sample.

## 3. Spatio-temporal Gaussian random fields

The agent has an onboard spatio-temporal model which is updated with the data that is gathered. The model is also used to compute the expected rewards along potential design trajectory and decisions for adaptive sampling. For ease of data assimilation and real-time decision-making, a GRF is used onboard the agent. Notably, derivatives or differences are then also Gaussian distributed.

### *3.1. Gaussian random fields*

A GRF is fully described by its mean $\mu(\boldsymbol{s}) = \mathrm{E}[x(\boldsymbol{s})]$, $\boldsymbol{s} \in \mathcal{D} \times \mathcal{R}^+$ and a covariance function $C(\boldsymbol{s}, \boldsymbol{s}') = \mathrm{Cov}\left(x(\boldsymbol{s}), x(\boldsymbol{s}')\right)$, see e.g. Cressie and Wikle (2015). In our case

study, the mean function is specified from physical oceanography modeling for the domain of interest, which involves a spatio-temporally varying function in the initial (prior) mean. The covariance function is specified from multiple ocean models as well as previously acquired data from the domain of interest. For the space-time covariance, we assume a separable model so that

$$C(\boldsymbol{s}, \boldsymbol{s}') = \sigma^2 \exp\left(-\left(\frac{d(\boldsymbol{s}, \boldsymbol{s}')}{\phi_s}\right)^2\right) \exp\left(-\left(\frac{|t_1 - t_2|}{\phi_t}\right)^2\right), \tag{4}$$

and as we limit scope to a fixed operational depth, we do not incorporate any kind spatial anisotropy which would be relevant to enhance smaller correlation in depth than in the lateral domain.

For any set of $N$ space-time locations $\mathcal{S} = \{\boldsymbol{s}_1, \boldsymbol{s}_2, \ldots, \boldsymbol{s}_N\}$, the random vector $\boldsymbol{x}_\mathcal{S} = (x(\boldsymbol{s}_1), x(\boldsymbol{s}_2), \ldots, x(\boldsymbol{s}_N))$ is then Gaussian distributed with mean vector

$$\boldsymbol{\mu}_\mathcal{S} = (\mu(\boldsymbol{s}_1), \mu(\boldsymbol{s}_2), \ldots, \mu(\boldsymbol{s}_N)), \tag{5}$$

and a symmetric positive semi-definite covariance matrix

$$\boldsymbol{\Sigma}_\mathcal{S} = \begin{pmatrix} C(\boldsymbol{s}_1, \boldsymbol{s}_1) & C(\boldsymbol{s}_1, \boldsymbol{s}_2) & \ldots & C(\boldsymbol{s}_1, \boldsymbol{s}_N) \\ C(\boldsymbol{s}_2, \boldsymbol{s}_1) & C(\boldsymbol{s}_2, \boldsymbol{s}_2) & & C(\boldsymbol{s}_2, \boldsymbol{s}_N) \\ \vdots & & \ddots & \\ C(\boldsymbol{s}_N, \boldsymbol{s}_1) & C(\boldsymbol{s}_N, \boldsymbol{s}_2) & & C(\boldsymbol{s}_N, \boldsymbol{s}_N) \end{pmatrix}. \tag{6}$$

In short notation, we write this as

$$\boldsymbol{x}_\mathcal{S} \sim \mathcal{N}(\boldsymbol{\mu}_\mathcal{S}, \boldsymbol{\Sigma}_\mathcal{S}). \tag{7}$$

*3.2. Directional differences and GRFs*

Because $g(\boldsymbol{s}, \boldsymbol{s})$ in Equation (1) is a linear combination of two Gaussian distributed variables, and hence $g(\boldsymbol{s}, \boldsymbol{s})$ is also Gaussian. In particular, we have mean

$$E[g(\boldsymbol{s}, \boldsymbol{s}')] = \frac{\mu(\boldsymbol{s}') - \mu(\boldsymbol{s})}{d(\boldsymbol{s}, \boldsymbol{s}')}, \tag{8}$$

and variance

$$\begin{aligned}\text{Var}\left(g(\boldsymbol{s},\boldsymbol{s}')\right) &= \frac{1}{d(\boldsymbol{s},\boldsymbol{s}')^2}\left(\text{Var}(x(\boldsymbol{s}))+\text{Var}(x(\boldsymbol{s}'))-2\text{Cov}(x(\boldsymbol{s}),x(\boldsymbol{s}'))\right)\\ &= \frac{1}{d(\boldsymbol{s},\boldsymbol{s}')^2}\left(C(\boldsymbol{s},\boldsymbol{s})+C(\boldsymbol{s}',\boldsymbol{s}')-2C(\boldsymbol{s},\boldsymbol{s}')\right). \qquad (9)\end{aligned}$$

In doing so, one can further take the difference between any pairs of variables along a transect line in the spatial domain. Then the random vector of directional differences is multivariate Gaussian distributed.

When studying properties of such derivatives we can see one of the main benefits of using a Gaussian covariance function. Figure 3 shows three different 1D GRF realizations, one using a Gaussian covariance function, one using an exponential covariance function and one using a Matérn covariance function. For each of the covariance functions the correlation at 300 is 0.05. The realization using the Gaussian covariance function is the smoothest out of the three, and the gradients are also smooth. For the exponential the derivatives are extremely large. This means that one should impose a smooth correlation function when the goal is to search for hot-spots in gradients. Hence, even though we regard more complicated spatial or spatio-temporal correlation functions as promising models, such as the one with spatially varying anisotropy by Berild and Fuglstad (2023) or the advection-diffusion model of Foss et al. (2022) that have been applied to coastal domain ocean modeling, we did not pursue complex covariance models here.

*3.3. Conditioning to in-situ observations*

As the agent gathers data, it will update the on-board model. In doing so, it needs a model for the data. The measurement model is here defined via

$$y(\boldsymbol{s}) = x(\boldsymbol{s}) + \epsilon(\boldsymbol{s}) \quad \epsilon(\boldsymbol{s}) \sim \mathcal{N}(0,\tau^2), \qquad (10)$$

where the errors at different locations are assumed to be independent. The Gaussian assumption for the error terms crucially means that measurements are jointly Gaus-

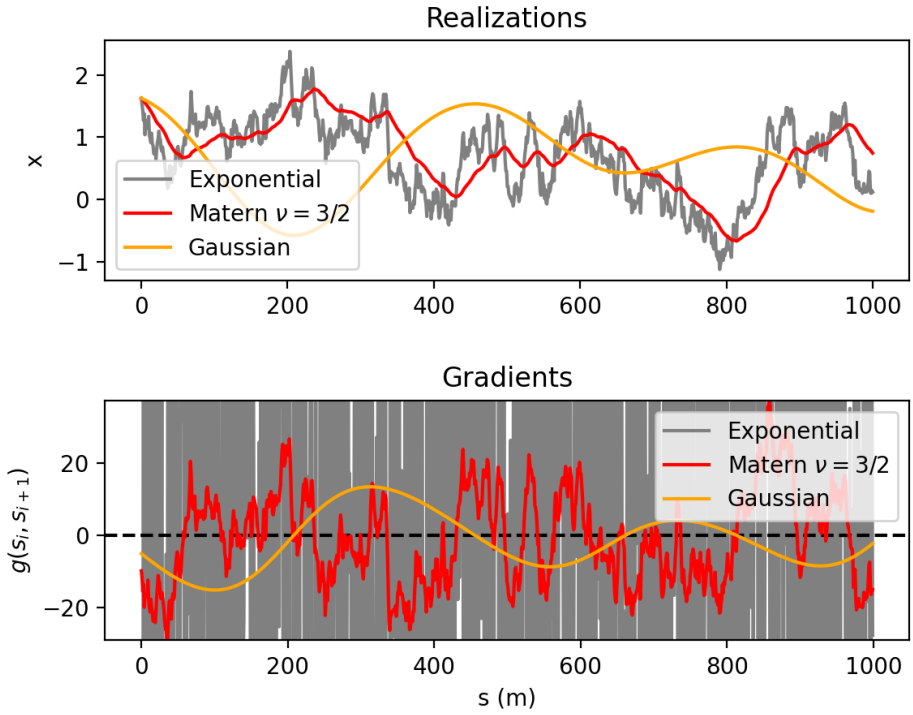



Figure 3: One realization using Gaussian, Matérn (smoothness $\nu = 3/2$) and exponential covariance function. The Gaussian covariance function gives a much smoother realization and gradient. The plotting scale for the gradients is truncated because the gradients for the exponential become extremely large.

sian distributed. Hence, given observations $\boldsymbol{y}_{1:k}$ in Equation (3), we can compute the conditional model using properties of the Gaussian distribution.

For any location set $\mathcal{P} \in \mathcal{P}_{k+1}^{j}$ among all possible design sets at stage $k+1$, we denote the associated variable $\boldsymbol{x}_{\mathcal{P}}$. Given the currently available data, the mean vector and covariance matrix are computed as follows:

$$\boldsymbol{m}_{\mathcal{P}} = \boldsymbol{\mu}_{\mathcal{P}} + \boldsymbol{\Sigma}_{\mathcal{P},\mathcal{S}_{1:k}}(\boldsymbol{\Sigma}_{\mathcal{S}_{1:k}} + \boldsymbol{T}_{\mathcal{S}_{1:k}})^{-1}(\boldsymbol{y}_{\mathcal{S}_{1:k}} - \boldsymbol{\mu}_{\mathcal{S}_{1:k}}), \tag{11}$$

$$\boldsymbol{\Psi}_{\mathcal{P}} = \boldsymbol{\Sigma}_{\mathcal{P}} - \boldsymbol{\Sigma}_{\mathcal{P},\mathcal{S}_{1:k}}(\boldsymbol{\Sigma}_{\mathcal{S}_{1:k}} + \boldsymbol{T}_{\mathcal{S}_{1:k}})^{-1}\boldsymbol{\Sigma}_{\mathcal{P},\mathcal{S}_{1:k}}^{T}, \tag{12}$$

where $\boldsymbol{T}_{\mathcal{S}_{1:k}} = \tau^2 \boldsymbol{I}_{N_{1:k}}$ is the measurement noise covariance matrix and $\boldsymbol{\Sigma}_{\mathcal{P},\mathcal{S}_{1:k}}$ is the cross-covariance between variables at locations $\mathcal{P}$ and those variables at former data

locations $\mathcal{S}_{1:k}$.

In particular, at any two points $\boldsymbol{s}$ and $\boldsymbol{s}'$, the variable $(x(\boldsymbol{s}), x(\boldsymbol{s}'))$ has a joint bivariate Gaussian distribution conditional on the observations $\boldsymbol{y}_{1:k}$. Their scaled difference is then Gaussian distributed, similar to what we have in Equation (8)-(9), and the same holds for variables along a transect.

*3.4. Efficient matrix calculations*

Matrix inversion or factorization can take a considerable amount of time when the number of data increases. Say, in Equation (11) and (12), one must invert the matrix $\boldsymbol{\Sigma}_{\mathcal{S}_{1:k}} + \boldsymbol{T}_{\mathcal{S}_{1:k}}$ of dimension $N_{1:k} \times N_{1:k}$, which is of order $O(N_{1:k}^3)$ calculations, and could quickly stall the agents´s computing performance. We utilize the structure with batch data collection, and then rely on a block version of the Sherman-Woodbury-Morrison formula for efficient matrix factorization (Petersen et al., 2008). In particular, we have that

$$\begin{bmatrix} \boldsymbol{\Sigma}_{\mathcal{S}_{1:k}} + \boldsymbol{T}_{1:k} & \boldsymbol{\Sigma}_{\mathcal{S}_{1:k},\mathcal{P}} \\ \boldsymbol{\Sigma}_{\mathcal{P},\mathcal{S}_{1:k}} & \boldsymbol{\Sigma}_{\mathcal{P}} \end{bmatrix}^{-1} = \begin{bmatrix} \boldsymbol{B} & -\boldsymbol{A}\boldsymbol{\Sigma}_{\mathcal{S}_{1:k},\mathcal{P}}\boldsymbol{C} \\ -\boldsymbol{C}\boldsymbol{\Sigma}_{\mathcal{P},\mathcal{S}_{1:k}}\boldsymbol{A} & \boldsymbol{C}^{-1} \end{bmatrix}, \tag{13}$$

where $\boldsymbol{A} = [\boldsymbol{\Sigma}_{\mathcal{S}_{1:k}} + \boldsymbol{T}_{1:k}]^{-1}$ is assumed to be available from the previous stage, and

$$\boldsymbol{B} = \boldsymbol{A} + \boldsymbol{A}\boldsymbol{\Sigma}_{\mathcal{S}_{1:k},\mathcal{P}}\boldsymbol{C}^{-1}\boldsymbol{\Sigma}_{\mathcal{P},\mathcal{S}_{1:k}}\boldsymbol{A}, \;\; \boldsymbol{C} = \boldsymbol{\Sigma}_{\mathcal{P}} - \boldsymbol{\Sigma}_{\mathcal{P},\mathcal{S}_{1:k}}\boldsymbol{A}\boldsymbol{\Sigma}_{\mathcal{S}_{1:k},\mathcal{P}}.$$

This calculation is used both to evaluate many designs $\{\mathcal{P}_{k+1}^j\}_{j=1,2,\ldots,J}$ and to update the mean and covariance in the data assimilation step. The required inversion is for $\boldsymbol{C}$ is of moderate size as it only involves the variables at the new batch or the potential transect locations.

This trick in Equation (13) allows efficient computing onboard an agent. However, over time the mere size of the covariance matrix leads to evaluation challenges. To approach this challenge further, we implement an on-board algorithm that reduces the data size over space-time by thinning data from far away/long ago. The agent

needs to make a decision in a reasonable amount of time, therefore we set a threshold time for how long the data assimilation and prediction stage should take. If the total time is larger than this threshold time then we thin the data points in memory. This will remove half the points in memory. The points that are thinned are mostly points that are far away in time because these points have a low correlation with the points we want to predict. After the thinning the inverse must be re-computed without using the recursive formula in Equation (13), but with the reduced number of data points this will take a shorter time than the threshold time.

## 4. Adaptive sampling design

For adaptive selection of designs as illustrated by the spider leg design in Figure 2, this shows the agent at some stage $k$ deciding to take one of 7 possible paths. In order to decide what path to take one needs to have an objective function. The agent will then make an optimal decision (direction) from the highest expected reward. We outline expected improvement (EI) in directional differences as our reward function. The onboard algorithm is summarized with the GRF and EI calculations, and involves some tuning parameters that we have tailored for the application.

### *4.1. Expected improvement for spider legs transects*

The spider leg designs shown in Figure 2 illustrate potential sampling designs for the agent at the current stage. For each of these transects, we calculate the conditional distribution, and base the path selection on the optimal expected reward. The expectation for a particular set of design points $\mathcal{P}$ is based on the conditional mean and covariance in Equation (11)-(12). In addition, we are mainly interested in the directional differences, see Equation (8)-(9).

Let $g_{\max}$ denote the largest absolute directional difference observed thus far in the sampling. We first study the probability of finding a larger directional gradient than this $g_{\max}$ along a transect. This probability of improvement (PoI) is chance of

having a difference that is larger than $g_{\max}$. Note that the maximum difference is in the observations and not in the true field. Nevertheless, we compare designs in the variables of interest because it is comparable between the different design transects at this stage. For a transect path $\mathcal{P} = \{\boldsymbol{p}_1, \boldsymbol{p}_2, \dots \boldsymbol{p}_n\}$, two-neigbour locations define the difference. The difference is Gaussian distributed as in Equation (8)-(9). To simplify the notation, we let the conditional distribution of the difference $g_i = g(\boldsymbol{p}_i, \boldsymbol{p}_{i+1}) \sim \mathcal{N}(\zeta_i, \eta_i^2)$, given the available data $\boldsymbol{y}_{1:k}$. Then the probability that $|g(\boldsymbol{p}_i, \boldsymbol{p}_{i+1})|$ is larger that $g_{\max}$ is

$$\text{PoI} = P(|g(\boldsymbol{p}_i, \boldsymbol{p}_{i+1})| \geq g_{max}) = 1 - \Phi\left(\frac{g_{max} - \zeta_i}{\eta_i}\right) + \Phi\left(\frac{-g_{max} - \zeta_i}{\eta_i}\right). \tag{14}$$

Now we compute this for all two-neighbor locations along a transect and compare the different transects to create a decision rule for which direction the agent should choose. The best transect according to the largest PoI is

$$\text{Best direction}_{Prob} = \underset{j \in \{1,2,\dots,J\}}{\arg\max} \max_{\boldsymbol{p}_i \in \mathcal{P}_k^j} P(|g(\boldsymbol{p}_i, \boldsymbol{p}_{i+1})| \geq g_{max}). \tag{15}$$

Note that even if $g_{\max}$ is the highest derivative in the field the PoI still gives a value larger than zero along each transect provided that the variance is larger than zero.

An alternative approach that accounts for the expected gain in the difference is available via the EI which has been used much in for instance the design of complex optimization problems (Zhan and Xing, 2020). The EI has a closed form solution for Gaussian distributions, see e.g. Gramacy and Apley (2015). Let

$$I(g_i) = \max(|g_i| - g_{max}, 0), \tag{16}$$

then the expected value of this improvement becomes

$$\begin{aligned} E[I(g_i)] &= (\zeta_i - g_{max})\left(1 - \Phi\left(\frac{g_{max} - \zeta_i}{\eta_i}\right)\right) + \eta_i \phi\left(\frac{g_{max} - \zeta_i}{\eta_i}\right) \\ &+ (-\zeta_i - g_{max})\Phi\left(\frac{-g_{max} - \zeta_i}{\eta_i}\right) + \eta_i \phi\left(\frac{-g_{max} - \zeta_i}{\eta_i}\right). \end{aligned} \tag{17}$$

Now we compute the EI for all two-neighbor locations along a transect and compare the different transects to create a decision rule for which direction the agent should choose. The best transect according to the largest EI is then

$$\text{Best direction}_{EI} = \underset{j\in\{1,2,\ldots,J\}}{\arg\max} \max_{\boldsymbol{p}_i\in\mathcal{P}_k^j} E(I(g_i)). \tag{18}$$

Later in Section 5 we will compare PoI against EI in a simulation study.

*4.2. Algorithm*

Along with the statistical model and the objective functions some other details are needed to fully describe the algorithm. The algorithm works in a sequential loop; sampling data, data assimilation, predicting gradients along possible paths and then use the objective function to choose one of these paths. The algorithm using EI is shown in Algorithm 1. When the AUV arrives at a waypoint (WP) it first assimilates the new data into the model (Section 3), then it finds 7 possible new paths it can take as defined by the spider-web legs. For each of these paths it predicts the salinity changes for several points along this path. It uses this prediction and EI (described in Section 4) to choose which transect is the best. The last step is to move along this transect, and it samples salinity data with a frequency of 1 Hz. The AUV does not move all the way until the end of the predicted transect, rather it uses a longer horizon and then it moves a shorter step-length. This is illustrated by the star in Figure 2. The reason for predicting far into the future and only moving a short path is that we can react quickly to the new measurements while still looking for derivatives far away.

Figure 4 illustrates some parts of the algorithm from the simulation study. In the figure the AUV has sampled for 20 steps and is making a decision on where to move next. There are 8 candidate transects for the AUV to move in. The next WP is set back towards the where the salinity change is large. Along the observed path

the conditional mean is closer to the true field. The conditional mean for the whole field is not computed during the mission, but here it is included only for illustration. The algorithm does require that we choose a step-length, horizon and a number of transects that fit the application.

Algorithm 1 shows the main steps in the sequential procedure. The AUV is guided by setting target waypoints ($\text{WP}_k$) for stage $k$.

**Algorithm 1** Sampling for derivatives.

**Require:**
$\mu(\boldsymbol{s})$, $C(\boldsymbol{s}, \boldsymbol{s}')$, operational domain $\mathcal{D}$.
$\mathcal{S}_0 = \emptyset$, $\boldsymbol{y}_0 = \emptyset$, $g_{\max} = 0$, $\text{WP}_0 = \boldsymbol{s}_{\text{start}}$.
**repeat** For each time $k = 1, \ldots$:
  Define spider legs $\mathcal{P}_k^j$ for $j = 1, \ldots, J$ transects.
  Define $\text{WP}_{k-1} = \text{WP}_k$ and $\text{EI}_{\max} = 0$
  **repeat** For each spider-leg $\mathcal{P}_k^j$, $j = 1, \ldots, J$
    Predict $\boldsymbol{m}_{\mathcal{P}_k^j}$ and $\boldsymbol{\Psi}_{\mathcal{P}_k^j}$ from $\mathcal{S}_{1:k-1}$ and $\boldsymbol{y}_{1:k-1}$ ▷ Eq. (11)-(12).
    Compute $\text{EI}_k^j = \max_{\boldsymbol{p}_i \in \mathcal{P}_k^j} E[I(g_i)]$ ▷ Eq. (17).
    **if** $\text{EI}_j > \text{EI}_{\max}$ **then**
      $\text{EI}_{\max} = \text{EI}_k^j$
      Set $\text{WP}_k$ one step-length down $\mathcal{P}_k^j$
    **end if**
  **until**
  AUV moves from $\text{WP}_{k-1}$ to $\text{WP}_k$, and it gathers data values $\boldsymbol{y}_k$ at points $\mathcal{S}_k$
  Update $\boldsymbol{y}_{1:k} = (\boldsymbol{y}_{1:k-1}, \boldsymbol{y}_k)$ and $\mathcal{S}_{1:k} = \{\mathcal{S}_{1:k-1}, \mathcal{S}_k\}$
  Update maximum derivative $g_{\max} = \max(g_{\max}, g_k)$
  **if** Update time > max update time **then**
    Points in memory are thinned.
  **end if**
**until**

## 5. Simulation study

Before deploying the AUV in the ocean we want to test different strategies in a simulated case. Here we generate a replicate study where we know the true salinity field. In this section we will go over the setup of the simulation study, the different

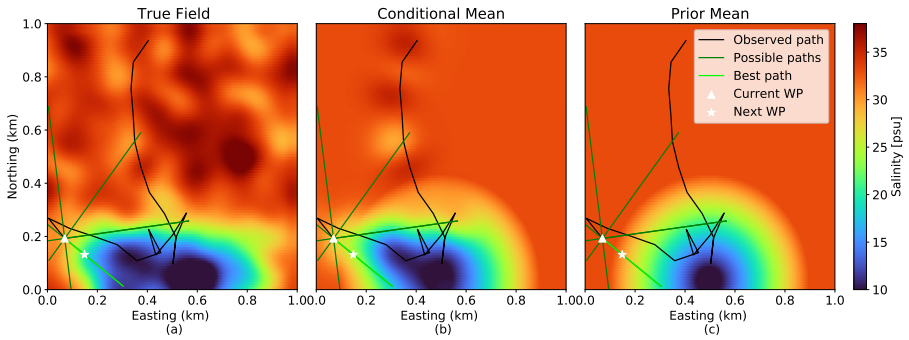



Figure 4: Illustration of spider leg designs over; (a) one realization of the true salinity field, (b) the conditional mean salinity field and (c) the prior mean field. All three plots show the observed path of the AUV in black. The conditional mean changes compared with the prior mean close to the observed path, and becomes closer to the true field. The conditional mean for the full field is only included for illustration here. The possible paths are shown in dark green and the best path in light green. The next waypoint is played one step-length down the best path.

metrics used for evaluating the different strategies, and the conclusions that become recommendations for the real-world setup.

### *5.1. Simulation setup*

For the simulation study we use a setup with a square 2D field as shown in Figure 4, the field is of size 1km×1km. For the simulation we need to have a true-field and a prior mean, both fields are static. The prior should capture some characteristics of the true field, but will not be completely accurate. For all the tests we will run 100 replicates, where each replicate gives a unique realization of the true field.

The starting location for the AUV is different for each replicate and will always be several step-lengths away from the interesting high gradient regions. The speed of the AUV is set to be 1 m/s with a sampling frequency of 1 Hz. For all experiments the AUV will run for a total of 5000 m. The step-length for the AUV is set to be 100 m, and the horizon is 500 m.

### *5.2. Evaluation metrics*

The evaluation metrics look at how large the largest gradient observed is and how many important regions the AUV is able to visit.

One way to check if one strategy is better than another is to see the absolute gradient $g_{\max}$ measured after sampling for a time $t$. Ideally we would want the $g_{\max}$ to be as large as possible for each stage $k$. We will use this to ascertain how well a strategy is performing. We can define $g_{\max}(t)$ as

$$g_{\max}(t) = \max_{t_i<t} |g(\boldsymbol{s}_i, \boldsymbol{s}_{i-1})|. \tag{19}$$

This metric will be a good indicator of whether we have found some large gradient during the mission. But the metric will not tell us how well the AUV is able to explore gradients in different regions of the field. For this purpose we need another evaluation metric. The AUV should be able to explore several regions where the gradient is high. We refer to these regions containing large gradients by important regions. One aims to visit many important regions during the mission. We split the region $\mathcal{D}$ into $N = 400$ equal regions, and then we look at the 20 % regions with the largest absolute gradients. In Figure 5 we illustrate the important regions (marked with red in the right plot) for

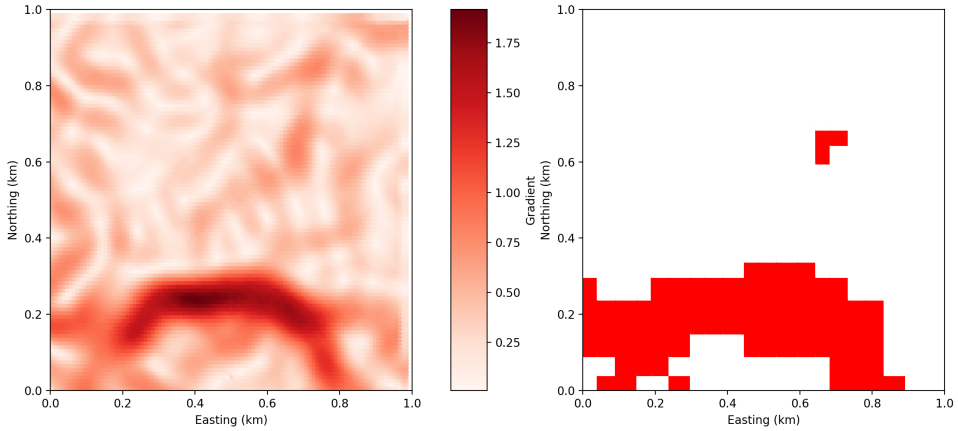



Figure 5: One realization showing absolute gradient (left plot) and the corresponding important regions (right plot). The corresponding regions are the 20 % regions with the highest gradients.

one replicate realization of the field. These important regions are different for each

replicate, but remain static for the whole simulation time. It is considered better if the AUV visits more of these regions. Hence, we construct a performance measure by counting how many of these regions the AUV visits on its exploration of the field. We count this over distance traveled (which is proportional to time).

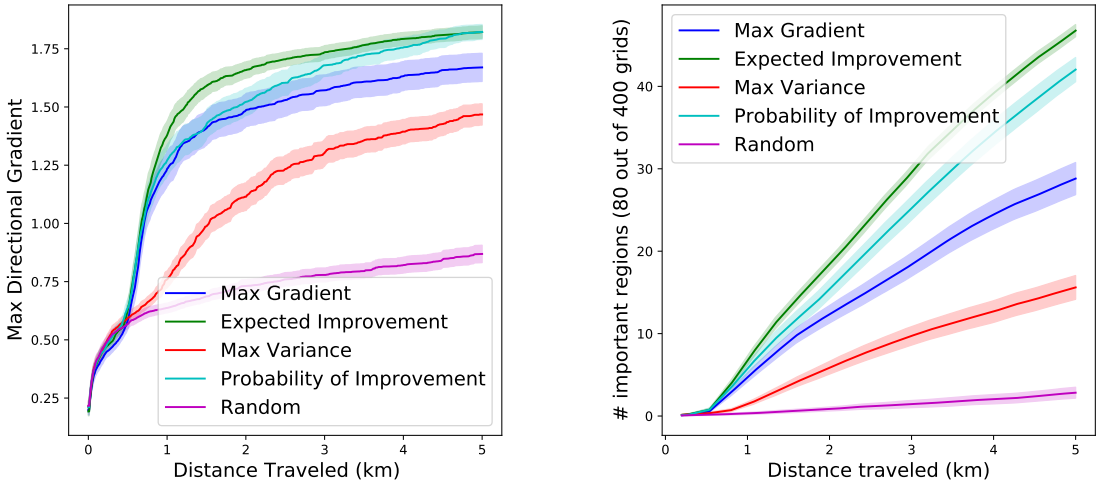
### *5.3. Simulations results*

We have proposed two different objective functions that can be used in looking for large gradients; PoI (14) and EI (17). In addition to these, we compare performance with 3 other strategies; one goes in the direction with the largest expected gradient, one goes to the largest variance in the derivatives and the last is a random walk. All strategies use the same algorithm as described in Algorithm 1 with the statistical model described in Section 3, the only difference is that the objective function is swapped.

The results for running simulations with these 5 different objective functions using a step-length of 100 m and a horizon of 500 m are shown in Figure 6. We first study the AUV's ability to detect large directional derivatives. In Figure 6 Left) we show the maximum derivative discovered by each of the five criteria over time. The thick line is the mean over the 100 replicates while the shaded region represents 2 standard errors in this mean. We clearly see that the PoI and EI work better than just looking for the max gradient, the reason for this is that it might get stuck in a local maxima, and it will not move away. This plot shows that EI does the best out of the five objective functions, although the gap between PoI and EI closes towards the end of the simulation. The random strategy and the one going for largest variance in the field are not performing so well, compared with the others.

In Figure 6 right) we observe how many important regions the AUV has visited after running a given distance. The thick lines indicate the mean number of regions visited for a given distance, and the shaded region represents 2 standard errors in

this mean. The ranking of the different objective functions is the same as in the left plot. The relative performances can be viewed a bit differently. Just looking for the maximum gradients does not really explore the field all that well. Rather, it can get stuck in local minima. Focusing on the maximum variance in the gradients, will eventually sample in the important regions, but it spends too much time in the exploration. Lastly, the gap between PoI and EI does not close down towards the end here. From this test, the conclusion is that EI works best overall.




(a) (b)

Figure 6: Compare the performance of different objective functions with 100 realizations for each. (a) increase in mean $g_{\max}$ and (b) increase in important regions visited. Both (a) and (b) show that expected improvement works best, with probability of improvement following.

To inspect further we take a look at worst-case and best-case outcomes of the simulations to see how they differ for the different objective functions. This can be important because it can be difficult to conduct many experiments in the ocean. Table 1 shows the 5th, 50th and 95th percentiles for the two metrics at the end of the simulation. For the 95th and 50th percentile both EI and PoI have a similar performance, but for the 5th percentile the difference is large. In the 5th percentile

EI is able to sample in 39 (out of 80) important regions, while PoI is only able to sample in 19. This means that EI is much better at exploring the important parts of the field in the worst cases.

| | $g_{\max}$ | | | Important Regions | | |
|---|---|---|---|---|---|---|
| Objective function | 5 % | 50% | 95 % | 5 % | 50% | 95 % |
| EI | 1.52 | 1.81 | 2.13 | 39 | 47 | 56 |
| PoI | 1.49 | 1.83 | 2.18 | 19 | 45 | 55 |
| Max Gradient | 0.61 | 1.78 | 2.15 | 0 | 33 | 47 |
| Max Variance | 0.87 | 1.50 | 1.94 | 2 | 14.5 | 31 |
| Random | 0.61 | 0.81 | 1.54 | 0 | 1.0 | 13 |

Table 1: Percentile table for the different objective functions with the two different evaluation metrics. These percentiles are calculated at the end of the simulations.

Regarding algorithmic parameter tuning we conducted some other tests to evaluate other aspects of the algorithm. One thing to test was how the horizon affected the performance, not surprisingly the longer the horizon the better. The main reason to limit the horizon is the computational cost for each iteration. It was also important that the step length was not too long as the AUV tends to be overstepping the phenomena in that case. It must travel the long way back, and this wastes time.

## 6. Case study

The suggested algorithm was tested in the Trondheim fjord on June 22. 2023. The AUV ran the adaptive mission for 2 hrs 10 min starting at 11:00 am. We first describe the parameter specification in the spatio-temporal model using numerical ocean data. We then describe the AUV setup and finally show results of the deployment.

### *6.1. Prior model specification based on SINMOD*

We have access to a numerical ocean model for the fjord called SINMOD developed by SINTEF Ocean (Slagstad and McClimans, 2005). In out case the model simulates

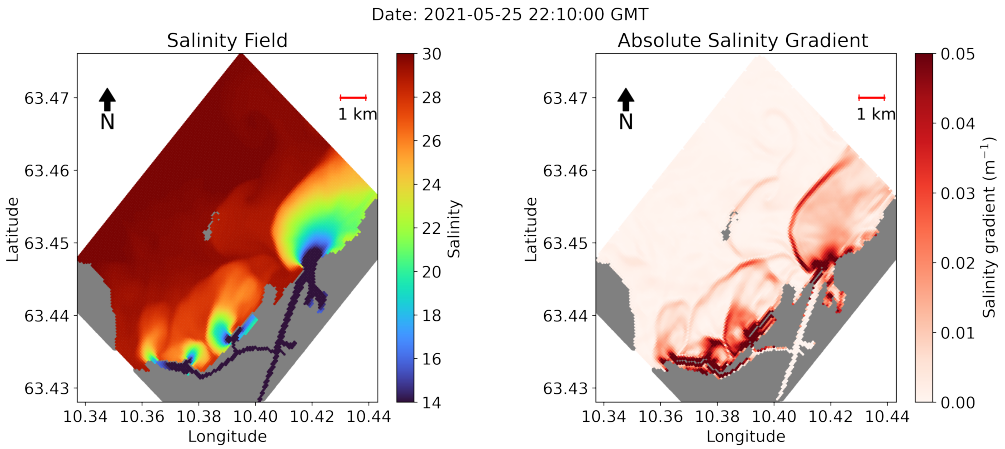



Figure 7: SINMOD simulation for the surface level of the river plume in the Trondheim fjord. Left plot shows the salinity level and the right shows the absolute salinity gradient. Left plot shows that the river has 4 outlets. The right plot shows clear river fronts in dark red. The tide is going from high tide to low tide.

several features of the fjord like currents, temperature, and salinity, but we are mostly interested in the salinity and the spatial salinity changes. Figure 7 shows a snapshot from a simulation, the left plot shows the salinity level and the right plot shows the absolute gradient. At this time in the simulation the water level goes from high tide to low tide[1] The dark red regions in the right plot show the river front. This is the region that is most interesting to sample.

These SINMOD simulations are computationally heavy to run. Therefore we use simulations done some time before the mission. We can use the outcome of the simulation as the prior mean for the surrogate GRF model of the salinity field.

The spatial and temporal covariance parameters $\phi_t$ and $\phi_s$, and the sill $\sigma$ in Equation (4) are estimated from a variogram analysis of SINMOD data. The parameters are $\phi_t = 5400s$, $\phi_s = 530m$ and $\sigma = 2$. The measurement noise for the salinity sensor is estimated from previous AUV deployments in the same location. The variance for

[1]Tide data gathered from https://www.kartverket.no/

the difference between two measurements is

$$\text{Var}[y(\boldsymbol{s}_i) - y(\boldsymbol{s}_{i-1})] = \text{Var}[x(\boldsymbol{s}_i) - x(\boldsymbol{s}_{i-1})] + \text{Var}[\epsilon(\boldsymbol{s}_i) - \epsilon(\boldsymbol{s}_{i-1})].$$

The salinity sensor has a frequency of 1 Hz and the AUV maintains a speed of 1.6 m/s, therefore two consecutive measurements are done within 1.6m and 1s of each-other. Large depth changes are also filtered out, because that salinity change in depth is much larger. Then we assume that $\text{Cov}(x(\boldsymbol{s}_i), x(\boldsymbol{s}_{i-1})) \approx \sigma^2$, therefore $\text{Var}[x(\boldsymbol{s}_i) - x(\boldsymbol{s}_{i-1})] \approx 0$. Then

$$\text{Var}[y(\boldsymbol{s}_i) - y(\boldsymbol{s}_{i-1})] \approx \text{Var}[\epsilon(\boldsymbol{s}_i) - \epsilon(\boldsymbol{s}_{i-1})] = 2\tau^2. \tag{20}$$

We can also get that $E[y(\boldsymbol{s}_i) - y(\boldsymbol{s}_{i-1})] \approx 0$ for such close measurements. This means that we can estimate $\tau$ by using

$$\hat{\tau}^2 = \frac{1}{2(n-1)} \sum_{i=1}^{n-1} \left(y(\boldsymbol{s}_i) - y(\boldsymbol{s}_{i-1})\right)^2. \tag{21}$$

We get $\tau = 0.27$.

The algorithm contains other tuning parameters, the step-length to be 250 m, and the prediction horizon to be 1000 m. The AUV will evaluate at most 7 transects, and the maximum planning time is set at 5 s. The target depth layer for the deployment is set at 1 m.

*6.2. AUV Setup*

In this field experiment, a Light Autonomous Underwater Vehicle (LAUV) from NTNU's Applied Underwater Robotics Laboratory (AURLab) was employed. Pre-launch protocol consisted of standard remote control verification (Figure 8).

The primary computational unit of the LAUV is the NVIDIA Jetson TX2. The vehicle's onboard algorithmic capabilities are augmented through the integration of an adaptive sampling framework (Mo-Bjørkelund et al., 2020), which mediates message exchange between the Robot Operating Systems (ROS) (Quigley, 2009) and

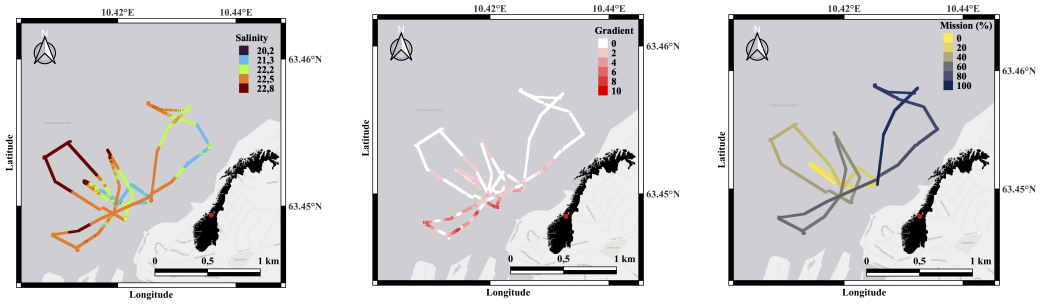
Figure 8: The AUV named Thor is heading towards the river mouth area where potential high gradient might exist.

DUNE (DUNE: Unified Navigation Environment(Pinto et al., 2013)). Communication among the vehicle's components utilizes the Inter-Module Communication (IMC) protocol (LSTS, 2022). The integration follows the scheme outlined in Ge et al. (2023), providing additional insights into the ROS-IMC bridge.

The AUV maneuvered at a depth of 1 m, where the salinity variance is large. Also, the AUV is less prone to colliding with small boats when keeping this depth, so it induces less risk. The AUV is programmed to try to maintain a speed of 1.6 m/s, which is much faster than the dynamics of the plume phenomenon observed in the SINMOD results. It was configured to re-surface at 10-minute intervals for navigational adjustments.

### *6.3. Results*

The trajectory made by the AUV is shown in Figure 9. The left plot shows the measured salinity along the path, the middle plot shows the absolute directional




(a) (b) (c)

Figure 9: (a) Observed salinity along the AUV trajectory. (b) The observed absolute gradient along the trajectory. (c) The mission accomplished in time steps. The AUV takes 39 steps, each step is around 250m.

derivatives, and the right plot shows where the AUV is at any point in the mission. We notice that the AUV spends most of the time close to the river outlet, this is where the salinity change is expected to be the largest. The AUV measures the largest salinity changes in the south-west region of the map, this is around mid-way through the mission.

How the maximum gradient changes over time is shown in Figure 10. This display shows the increase in $g_{\max}$, the observed absolute gradient and the predicted absolute gradients. There are four distinct increases in the $g_{\max}$, in the start, 0.2 hours, 0.75 hours and 1.1 hours into the mission. At around 1 hour to 1.5 hours into the mission the AUV samples a region with a lot of salinity change, this is where the largest gradient is found. After this the AUV does not measure any very large salinity gradients. It is also interesting to look at what gradients the AUV predicts during the mission. The figure shows that when the AUV observes a large gradient, it also predicted a large gradient, however the prediction is often much larger. There are also some points where the model predicts a large gradient, but no large gradient is observed. This means that the model does perform one of the most important tasks which is to guide the agent towards large salinity changes

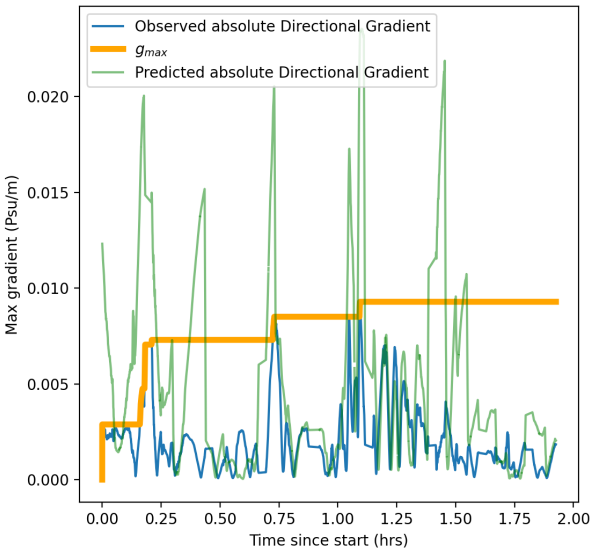



Figure 10: Increase in maximum derivative for the duration of the mission. The display also shows the measured absolute derivative during the mission and the predicted derivative.

During the mission the AUV takes 39 steps. In Figure 11 the value for the EI for each of the transects for a given step. The EI usually starts out with large values while the as $g_{\max}$ is low, this can be seen in step 4. Towards the end the EI drops closer to zero, the AUV will start to explore different regions where it does not necessarily predict that the gradient will be large, but rather that the variance is large.

We next study how predictions for the gradients correlate with the measured gradients. For each step $k$ we predict the salinity distribution along the path $\mathcal{P}_k$, but because of currents and other navigational errors the points we measure $\mathcal{S}_k$ will not be exactly the same, and it can be a large difference. We use the model estimated in step $k-1$ to predict the points $\mathcal{S}_k$ that will be measured in step $k$. Then we look at how well the model can predict the next transect, these contain 160 - 300 data points. The results from this are shown in Figure 12. There is a correlation between the predicted gradient and the observed gradient. Ideally we would like the values to

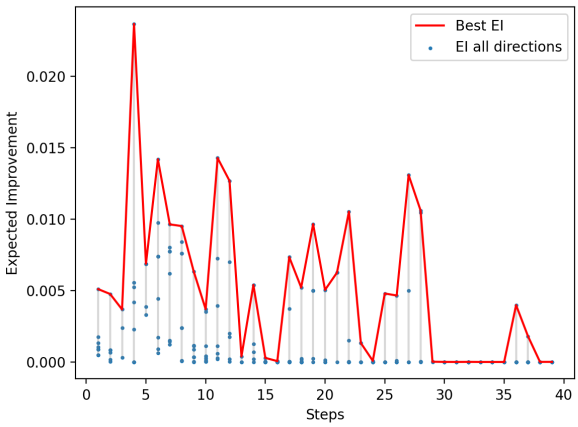



Figure 11: Expected improvement for each step of the mission. The red line represents the highest expected improvement, and this will be the direction the AUV moves in. The blue dots are the expected improvement values for the other possible directions.

lay along the 1:1 line, but the model tends to predict larger gradients than what is observed. This might be because the variance in the salinity for the prior model is larger than what is observed, leading the model to predict larger gradients.

In summary, the AUV was able to sample and measure salinity changes in the river plume. EI worked rather well at finding large gradients and for exploring different parts of the river plume. The onboard model gives a reasonable prediction for the gradients that are going to be observed, but the gradients predicted by the model are larger than what is observed.

## 7. Closing remarks

We have presented an approach for constructing sampling designs by an agent moving in a spatio-temporal domain. The goal is to provide valuable designs, which in our case involves locating regions that exhibit large spatial changes. For our application in oceanography, such locations could indicate transitional zones in water masses which are potentially indicative of much biological activity. The approach for adaptive sampling is based on a Gaussian random field model, and the directional

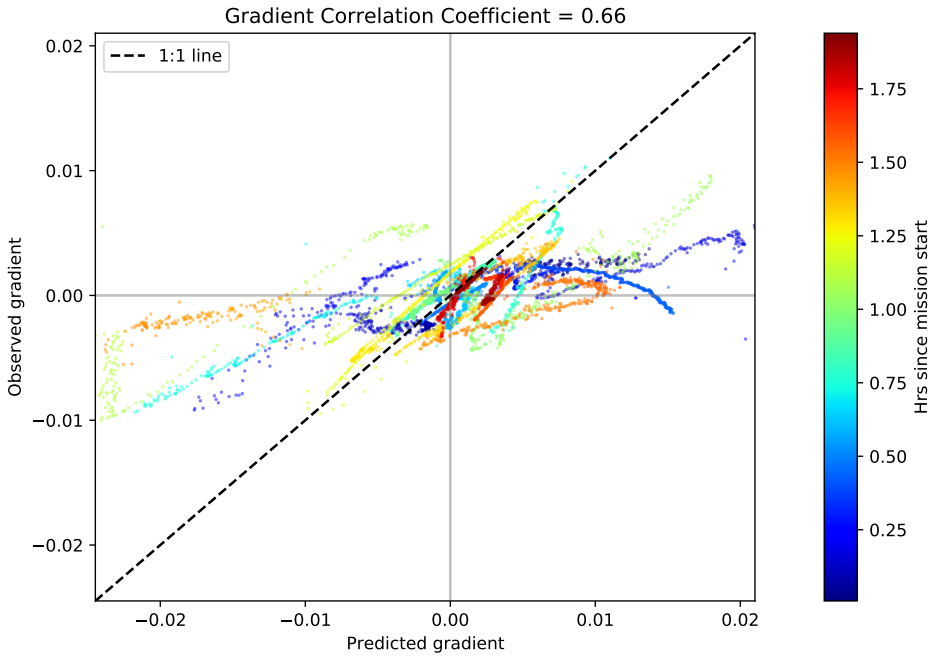



Figure 12: Correlation between the predicted and observed salinity gradients. There is a correlation between the predicted and the observed gradients, but the model predicts larger gradients than what is observed.

changes in the field are then also Gaussian distributed. By using a Gaussian model, one facilitates efficient calculations on the agent's limited computing resources. With the sampling design setting of a moving agent, we suggest a spider leg design at each stage of the adaptive operation, and we use expected improvement in directional differences to guide the adaptive sampling. There is hence no operational grid such as a waypoint graph. Instead, the prior model assumption is effectively updated with data at each stage, limited only by the size of the data vector. In long-term operations, storage problems can occur, and we suggest to fade distant data (in space and time). This allows long-time operations in large spatial domains. We demonstrate the merits of the approach in a simulation study and in a field deployment running an autonomous underwater vehicle in a Norwegian fjord.

The field of robotics and embedded computing is growing quickly with the technological advances in small-size computing units and the current societal focus on AI. Statistics should play a substantial role in the development of new algorithms in this field. We have shown one example of added value in using spatial statistics and spatial design for underwater robotics. There are plenty of other applications where statistics can contribute. In our experience working on this, the engineers see much merit in more formalized statistical methodologies. They are however striving for efficiency and real-time operation, so rather than overly complex statistical models or methods, there seems to be a need for fast and robust systems that still have reasonably good statistical properties.

For future work, we want to investigate more nuanced algorithms where one can tune the distance and design parameters to automatically capture the right scales on the fly. We also aim to look at multivariate fields which requires a re-formulation of the derivatives used here. Rather than just derivatives, one is often interested in volumes (spatial integrals). In oceanographic applications volumes of relevance include high biomass, oxygen production, net primary production, etc., see e.g. Wu et al. (2022). Integral expressions are linear operators and it is hence Gaussian distributed if the variable of interest is Gaussian. Many of the methods described in this paper can hence be used for such applications. We used a relatively standard spatio-temporal model here. It can be extended to more complex temporal dynamics as well as non-stationary spatial elements. Staying within the Gaussian model class, we can for instance build on advection-diffusion processes (Sigrist et al., 2015; Foss et al., 2022) or use links to stochastic partial differential equations in the spatial domain (Berild and Fuglstad, 2023).

**Acknowledgments**

This work is funded by the Research Council of Norway, project number 305445. The authors thank Ingrid Ellingsen and SINTEF for supplying the SINMOD simulations used for parameter estimation and the prior mean. We also than the people in the Aurlab [2] for making the deployment in the fjord possible, especially Karoline Barstein for operating the AUV during the deployment.

**References**

Belkin, I. M., Cornillon, P. C., and Sherman, K. (2009). Fronts in large marine ecosystems. *Progress in Oceanography*, 81(1-4):223–236.

Berild, M. O. and Fuglstad, G.-A. (2023). Spatially varying anisotropy for gaussian random fields in three-dimensional space. *Spatial Statistics*, 55:100750.

Brus, D. J. (2022). *Spatial sampling with R*. CRC Press.

Catto, J. L. and Pfahl, S. (2013). The importance of fronts for extreme precipitation. *Journal of Geophysical Research: Atmospheres*, 118(19):10–791.

Cressie, N. and Wikle, C. K. (2015). *Statistics for spatio-temporal data*. John Wiley & Sons.

Dhingra, S., Madda, R. B., Gandomi, A. H., Patan, R., and Daneshmand, M. (2019). Internet of things mobile–air pollution monitoring system (iot-mobair). *IEEE Internet of Things Journal*, 6(3):5577–5584.

Dunbabin, M. and Marques, L. (2012). Robots for environmental monitoring: Significant advancements and applications. *IEEE Robotics & Automation Magazine*, 19(1):24–39.

[2]https://www.ntnu.edu/aur-lab

Fedorov, K. N. (1986). *The physical nature and structure of oceanic fronts*, volume 333. Springer.

Fonseca, J., Bhat, S., Lock, M., Stenius, I., and Johansson, K. H. (2023). Adaptive sampling of algal blooms using autonomous underwater vehicle and satellite imagery: Experimental validation in the baltic sea. *arXiv preprint arXiv:2305.00774*.

Foss, K. H., Berget, G. E., and Eidsvik, J. (2022). Using an autonomous underwater vehicle with onboard stochastic advection-diffusion models to map excursion sets of environmental variables. *Environmetrics*, 33(1):e2702.

Fossum, T. O., Norgren, P., Fer, I., Nilsen, F., Koenig, Z. C., and Ludvigsen, M. (2021). Adaptive sampling of surface fronts in the arctic using an autonomous underwater vehicle. *IEEE Journal of Oceanic Engineering*, 46(4):1155–1164.

Fuhg, J. N., Fau, A., and Nackenhorst, U. (2021). State-of-the-art and comparative review of adaptive sampling methods for kriging. *Archives of Computational Methods in Engineering*, 28:2689–2747.

Ge, Y., Eidsvik, J., and Mo-Bjørkelund, T. (2023). 3d adaptive auv sampling for classification of water masses. *IEEE Journal of Ocean Engineering*, 48:626–639.

Gramacy, R. B. (2020). *Surrogates: Gaussian process modeling, design, and optimization for the applied sciences*. CRC press.

Gramacy, R. B. and Apley, D. W. (2015). Local gaussian process approximation for large computer experiments. *Journal of Computational and Graphical Statistics*, 24(2):561–578.

Hopkins, J., Challenor, P., and Shaw, A. G. (2010). A new statistical modeling approach to ocean front detection from sst satellite images. *Journal of Atmospheric and Oceanic Technology*, 27(1):173–191.

Krause, A., Singh, A., and Guestrin, C. (2008). Near-optimal sensor placements in gaussian processes: Theory, efficient algorithms and empirical studies. *Journal of Machine Learning Research*, 9(2):235–284.

Lermusiaux, P. F. (2006). Uncertainty estimation and prediction for interdisciplinary ocean dynamics. *Journal of Computational Physics*, 217(1):176–199.

LSTS (2022). Inter module communication protocal.

Manohar, K., Brunton, B. W., Kutz, J. N., and Brunton, S. L. (2018). Data-driven sparse sensor placement for reconstruction: Demonstrating the benefits of exploiting known patterns. *IEEE Control Systems Magazine*, 38(3):63–86.

Mateu, J. and Müller, W. G. (2012). *Spatio-temporal design: Advances in efficient data acquisition*. John Wiley & Sons.

Mo-Bjørkelund, T., Fossum, T. O., Norgren, P., and Ludvigsen, M. (2020). Hexagonal grid graph as a basis for adaptive sampling of ocean gradients using auvs. In *Global Oceans 2020: Singapore – U.S. Gulf Coast*, pages 1–5.

Petersen, K. B., Pedersen, M. S., et al. (2008). The matrix cookbook. *Technical University of Denmark*, 7(15):510.

Pinto, J., Dias, P. S., Martins, R., Fortuna, J., Marques, E., and Sousa, J. (2013). The lsts toolchain for networked vehicle systems. In *2013 MTS/IEEE OCEANS - Bergen*, pages 1–9.

Quigley, M. (2009). Ros: an open-source robot operating system. In *ICRA 2009*.

Schranz, M., Di Caro, G. A., Schmickl, T., Elmenreich, W., Arvin, F., Şekercioğlu, A., and Sende, M. (2021). Swarm intelligence and cyber-physical systems: concepts, challenges and future trends. *Swarm and Evolutionary Computation*, 60:100762.

Shadrin, D., Menshchikov, A., Somov, A., Bornemann, G., Hauslage, J., and Fedorov, M. (2019). Enabling precision agriculture through embedded sensing with artificial intelligence. *IEEE Transactions on Instrumentation and Measurement*, 69(7):4103–4113.

Sigrist, F., Künsch, H. R., and Stahel, W. A. (2015). Stochastic partial differential equation based modelling of large space–time data sets. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 77(1):3–33.

Slagstad, D. and McClimans, T. A. (2005). Modeling the ecosystem dynamics of the barents sea including the marginal ice zone: I. physical and chemical oceanography. *Journal of Marine Systems*, 58(1-2):1–18.

Wang, J.-F., Stein, A., Gao, B.-B., and Ge, Y. (2012). A review of spatial sampling. *Spatial Statistics*, 2:1–14.

Wang, Y., Le, N. D., and Zidek, J. V. (2020). Approximately optimal spatial design: How good is it? *Spatial Statistics*, 37:100409.

Wu, J., Goes, J. I., do Rosario Gomes, H., Lee, Z., Noh, J.-H., Wei, J., Shang, Z., Salisbury, J., Mannino, A., Kim, W., Park, Y.-J., Ondrusek, M., Lance, V. P., Wang, M., and Frouin, R. (2022). Estimates of diurnal and daily net primary productivity using the geostationary ocean color imager (goci) data. *Remote Sensing of Environment*, 280:113183.

Zhan, D. and Xing, H. (2020). Expected improvement for expensive optimization: a review. *Journal of Global Optimization*, 78(3):507–544.

Zhang, Y., Rueda, C., Kieft, B., Ryan, J. P., Wahl, C., O'Reilly, T. C., Maughan, T., and Chavez, F. P. (2019). Autonomous tracking of an oceanic thermal front by a wave glider. *Journal of Field Robotics*, 36(5):940–954.