Available online at www.sciencedirect.com

ScienceDirect

IFAC PapersOnline CONFERENCE PAPER ARCHIVE

IFAC PapersOnLine 55-31 (2022) 124–129

# Long-Horizon Informative Path Planning with Obstacles and Time Constraints

**Yaolin Ge** * **André Julius Hovd Olaisen** * **Jo Eidsvik** *
**R. Praveen Jain** ** **Tor Arne Johansen** **

* *Department of Mathematical Sciences, Norwegian University of Science and Technology (NTNU) (email: {yaolin.ge, andre.j.h.olaisen, jo.eidsvik}@ntnu.no.)*
** *Department of Engineering Cybernetics, NTNU (email: {ravinder.p.k.jain, tor.arne.johansen}@ntnu.no.)*

**Abstract:** We apply non-myopic informative path planning in a simulated river plume case study with several constraints on our agent. A cost valley philosophy is proposed to guide the agent through the field. The purpose of this path planner is to reveal the river plume front with the long-horizon while safely returning home in time. Among others, we employ RRT*, a variant of RRT (rapidly-exploring random trees), as the path planner to determine the least-cost path between locations. The distance budget from start to end destination, the obstacle constraint, and directional change are penalties, whereas the reduced variance of the field and an excursion set are the two rewards. The cost valley is then computed by superimposing those five fields. The simulation results demonstrate the efficiency of such a strategy. They show that the suggested approach balances exploitation and exploration while bearing in mind the go-home constraint.



*Keywords:* Path Planning, Autonomous underwater vehicles, Gaussian Random Field, and adaptive sampling

## 1. INTRODUCTION

Autonomous Underwater Vehicles (AUVs) have been used extensively for the investigation of different oceanographic phenomena (Hwang et al., 2019). AUV adaptive sampling has gained more interest in oceanographic surveying (Zhang et al., 2020; Fossum et al., 2019). Plume and ocean front investigation with AUVs has formed scenarios to validate adaptive sampling methods and algorithms (Fossum et al., 2021; Berget et al., 2018; Fossum et al., 2018). In using only one AUV to conduct the adaptive sampling, dominating methods can be grouped into either myopic (greedy) or non-myopic approaches. Myopic schemes guide the agent (AUV) towards the most informative location selected from a subset of candidate locations within the myopic neighborhood radius (Fossum et al., 2021; Berget et al., 2018). The greediness of such algorithms can make it too short-sighted, and it can fail at revealing other interesting areas. Non-myopic strategies can alleviate such challenges by expanding its horizon to a longer-stage (Bai et al., 2021). Xiao and Wachs (2022) show the effectiveness of such algorithms in a small-scale case where the shapes of the unknown objects are revealed by a robot arm, using global kriging variance reduction as the main criterion. However, the computational cost associated with such non-myopic algorithms is usually very high, and it might not apply to larger-scale onboard computation in the ocean. Suh et al. (2017) provide an idea of using cost-aware RRT* to generate sampling paths using cross-entropy as the cost function.

In our case, we want the agent to continuously determine a path for collecting valuable information from the salinity field to reveal the river plume front, with the goal being to map unknown ocean properties while considering the remaining distance budget and avoiding the potential risks of collision with mapped obstacles. Going beyond common pre-scripted planners and greedy strategies, we propose a long-horizon path planner to solve this problem by constructing a cost valley built on multiple penalty and reward fields.

In Section 2 we describe the background and the simulation case which motivates the study of long-horizon path planning for sampling the river plume front. In Section 3 we outline our modeling approach and the statistical methods for informative sampling with operational constraints. In Section 4 we present the algorithms associated with long-horizon path planning. In Section 5 we show results from a simulation study comparing standard pre-scripted lawnmower and myopic (greedy) strategies. In Section 6 we conclude and point to future research directions.

## 2. PROBLEM STATEMENT

We consider the challenge of AUV adaptive sampling to map a spatial salinity field to uncover the river plume front in a specified domain. The focus is on the ability of the AUV to conduct adaptive sampling when there are constraints at the start and end point of the deployment, limited distance budget for the mission, and static obstacles in the field.

In this simulated case, we consider a two-dimensional domain. The variable of interest is salinity, which is assumed to vary smoothly in the domain. Fig. 1 illustrates a syn-


Peer review under responsibility of International Federation of Automatic Control.
10.1016/j.ifacol.2022.10.419

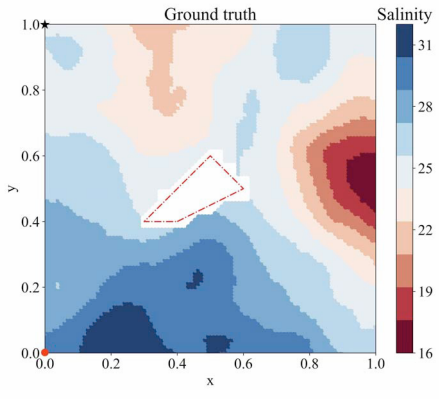


Fig. 1. Exemplary true salinity field on the unit square domain. The starting point is $(0, 0)$ and the endpoint is $(0, 1)$. The obstacle is in the center, marked by red dashed lines.

thetic case for the unit square domain. Here, the surface salinity field is generated from a Gaussian random field (GRF) model. In this salinity realization, there is a very low salinity region to the east and also some low salinity values to the north. In a practical case, such regions could represent water masses from river plumes. The red polygon inside the field shows an obstacle that might represent an island. The agent is deployed at the red dot in the south-west corner (coordinates $(0, 0)$), and one must retrieve the agent after a certain time at the black star in the north-west corner (coordinate $(0, 1)$). The perimeter of the domain is 4 units, and the distance budget for the AUV is set to be 5 units. To achieve its task, the agent needs to conduct adaptive path-planning and make sure that it is maximizing information-gathering objectives while getting back to the desired destination in time and avoiding the collision.

## 3. STATISTICAL MODELS AND METHODS FOR AUV SAMPLING

### *3.1 On-board computing with GPs*

We denote the salinity variable by $\xi_{\boldsymbol{u}}$, with spatial variable $\boldsymbol{u} \in \mathcal{M} \subset \mathcal{R}^2$, where $\mathcal{M}$ denotes the domain of interest. We assume that the salinity field is represented by a GRF. Similar assumptions of Gaussianity on salinity variables have been used in e.g. Das et al. (2013) and Binney et al. (2010). The initial specification then includes the estimation of underlying trends, variability, and spatial dependence. We assume that they can be extracted from ocean model data (Slagstad and McClimans, 2005).

The GRF modeling assumptions enable fast onboard data assimilation and AUV adaptive sampling efforts. For onboard implementation and computing, the domain is $\mathcal{M}$ which is discretized to a set of $n$ grid locations; $\{\boldsymbol{u}_1, \ldots, \boldsymbol{u}_n\}$. The prior GRF model at these grid locations is denoted by

$$\boldsymbol{\xi} = (\xi_{\boldsymbol{u}_1}, \ldots, \xi_{\boldsymbol{u}_n})^T, \qquad \boldsymbol{\xi} \sim N(\boldsymbol{\mu}, \boldsymbol{\Sigma}), \tag{1}$$

with mean vector $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$. We assume a Matern correlation function so that $\Sigma(i, i') = \sigma^2(1 + \phi h(i, i')) \exp(-\phi h(i, i'))$, with variance $\sigma^2$, correlation decay parameter $\phi$ and Euclidean distance $h(i, i')$ between sites $\boldsymbol{u}_i$ and $\boldsymbol{u}_{i'}$.

The measurement $y_j$ at each stage $j = 1, \ldots, N_{\text{steps}}$ is modeled by

$$y_j|\boldsymbol{\xi} \sim N(\boldsymbol{f}_j^T \boldsymbol{\xi}, r^2), \tag{2}$$

where the vector $\boldsymbol{f}_j$ defines the sampling indices at this stage of operation and $r$ is the salinity measurement noise standard deviation. The sampling design $\boldsymbol{D}_j$ at this stage $j$, say directions north, east, west, or south, determines the 0 and 1 structure in vector $\boldsymbol{f}_j$ because it directly defines the measurement location. This aspect will be important for design evaluation in what follows.

Starting with $\boldsymbol{m}_0 = \boldsymbol{\mu}$ and $\boldsymbol{S}_0 = \boldsymbol{\Sigma}$, Bayes' rule is used to achieve data assimilation at stages $j = 1, \ldots, N_{\text{steps}}$. This gives the updated Gaussian model with mean and variance given by

$$\begin{aligned} \boldsymbol{G}_j &= \boldsymbol{S}_{j-1}\boldsymbol{f}_j(\boldsymbol{f}_j^T \boldsymbol{S}_{j-1}\boldsymbol{f}_j + r^2)^{-1} \\ \boldsymbol{m}_j &= \boldsymbol{m}_{j-1} + \boldsymbol{G}_j(y_j - \boldsymbol{f}_j^T \boldsymbol{m}_{j-1}) \\ \boldsymbol{S}_j &= \boldsymbol{S}_{j-1} - \boldsymbol{G}_j \boldsymbol{f}_j^T \boldsymbol{S}_{j-1}. \end{aligned} \tag{3}$$

### *3.2 Information criterion for sampling*

Based on our problem statement in Section 2, we have chosen to use numerous information criteria in the objective function which determines the AUV sampling design. The overall function is defined by a sum of normalized versions of the following criteria that we describe next:

**Integrated variance reduction (IVR)** uses the latter part of the posterior covariance calculation in (3). As an information criterion, the goal now is to provide maximum reduction of the marginal variances at all spatial locations in the grid, see also Binney et al. (2010) and Fossum et al. (2018). For a particular AUV sampling design $\boldsymbol{D}_j$, defined via the sampling design vector $\boldsymbol{f}$, the variance reductions at this stage are given by the diagonal entries of $\boldsymbol{R}_j = \boldsymbol{S}_{j-1}\boldsymbol{f}_j(\boldsymbol{f}_j^T \boldsymbol{S}_{j-1}\boldsymbol{f}_j + r^2)^{-1}\boldsymbol{f}_j^T \boldsymbol{S}_{j-1}$. The sum of these represents an information measure.

**Expected integrated Bernoulli variance reduction (EIBV)** uses the classification of salinity above and below a threshold $t$ according to an excursion set $\text{ES} = \{\boldsymbol{u} \in \mathcal{M} : \xi_{\boldsymbol{u}} \leq t\}$, and the goal is to increase the classification accuracy of salinity levels according to this threshold. The Bernoulli variance (BV) at location $\boldsymbol{u}$ is $p_{\boldsymbol{u}}(1 - p_{\boldsymbol{u}})$, $p_{\boldsymbol{u}} = P(\xi_{\boldsymbol{u}} \leq t)$, which we aim to reduce by the data gathering. The goal of a sampling design for data $y_i$ is then to minimize the expected spatially integrated Bernoulli variance, and for the Gaussian model, there is a closed-form solution (Fossum et al., 2021).

**Obstacle avoidance** ensures that the AUV does not crash into land or islands. There is an infinity cost penalty if the AUV runs into an obstacle. In practice, a finite cost penalty can be applied to account for uncertainty.

**Directional change** sets a penalty for sharp AUV turns. Beyond the $90^o$ limit, there is an increasing penalty for high turning angles, so only very rarely (in situations with conflicting objectives), will we see abrupt angle changes in the AUV path.

**Budget limitations** include constraints that make sure that the AUV gets to its destination in time. The budget penalty will start when the mission approaches the end, and form an elliptical region away from the current AUV location towards its destination. There is infinity loss outside this region. The cost can be finite in practice as mentioned above.

The objective function guiding the AUV sampling is composed of all these measures. In doing so, we aim to balance exploration for uncertainty and salinity boundaries with operational constraints for the vehicle. At each stage $j = 1, \ldots, N_{\text{steps}}$, the AUV updates its calculation of a *Cost Valley* based on evaluating all these criteria for feasible designs. The results are used to compare designs and for selecting the optimal sampling design $\boldsymbol{D}_j$ at the current stage. The procedures involving model updating, design criterion calculation, and data gathering are summarized in Algorithm 1.

**Algorithm 1** Informative sampling algorithm

**Require:** Initial mean $\boldsymbol{m}_0$ and covariance $\boldsymbol{S}_0$
  Set start waypoint $\boldsymbol{D}_1 = \{\boldsymbol{u}_1\}$
  $\mathcal{Y}_0 = \emptyset$
  $j = 1$
  **while** $j \leq N_{\text{steps}}$ **do**
    **Act** :
      Go to waypoint $\boldsymbol{D}_j$.
    **Sense** :
      Gather data $y_j$. $\mathcal{Y}_j = (\mathcal{Y}_{j-1}, y_j)$.
      $\boldsymbol{G}_j = \boldsymbol{S}_{j-1}\boldsymbol{f}_j(\boldsymbol{f}_j^T\boldsymbol{S}_{j-1}\boldsymbol{f}_j + \boldsymbol{R}_j)^{-1}$
      $\boldsymbol{m}_j = \boldsymbol{m}_{j-1} + \boldsymbol{G}_j(y_j - \boldsymbol{f}_j^T\boldsymbol{m}_{j-1})$
      $\boldsymbol{S}_j = \boldsymbol{S}_{j-1} - \boldsymbol{G}_j\boldsymbol{f}_j^T\boldsymbol{S}_{j-1}$
      $j = j + 1$
    **Plan** :
      Budget = Budget - $||\boldsymbol{D}_j - \boldsymbol{D}_{j-1}||_2$
      $\mathbf{CV} = \text{updateCostValley}(\boldsymbol{m}_j, \boldsymbol{S}_j, \text{Budget}, \boldsymbol{u}_j, \boldsymbol{u}_{j-1})$
      $\boldsymbol{u}_j = \text{argmin}_{\boldsymbol{u}\in\mathcal{M}}(\mathbf{CV})$
      $\boldsymbol{T}_j = \text{RRT*}(\mathbf{CV}, \boldsymbol{D}_{j-1}, \boldsymbol{u}_j)$
      $\boldsymbol{D}_j = \boldsymbol{T}_{j1}$
  **end while**

## 4. PATH PLANNER

We design a path planning algorithm to find the cheapest path through a cost valley. This path should take the AUV from the current location to the lowest point in the cost valley. The cost valley is defined by combining information criteria, direction criteria, and budget limitations. The cost valley is updated with data as the AUV gathers salinity information from the field. The cost valley philosophy relies on a long-horizon path plan, which is not only considering the next step but anticipating future steps.

We build such a cost valley based on the overlay of five different information criteria. The weights among them are equal, but before merging them, all the costs are normalized to be within range $[0, 1]$, except those costs which are $\infty$. The RRT* path planner, see e.g. Karaman and Frazzoli (2011) and Hollinger and Sukhatme (2014), is used to determine the optimal least-cost path from the current AUV location to the end destination. From this calculation, the agent selects the next optimal design location. It measures the salinity $y_j$ at this design location and the entire GRF model is updated. This means that the cost valley is also updated, and hence the new starting location will be fed to the RRT* path planner.

The building blocks of the core algorithm are presented in Algorithm 2-8.

**Algorithm 2** RRT*, called by Algorithm 1

**Require:** $\mathbf{CV}, \boldsymbol{D}_{j-1}, \boldsymbol{u}_j$
  **for** $k \in 1 \ldots K$ **do**
    Generate random location $\boldsymbol{u}_{\text{rand}}$ within constraints.
    $\boldsymbol{u}_{\text{nearest}} \leftarrow \mathbf{Nearest}(G = (V, E), \boldsymbol{u}_{\text{rand}})$
    $\boldsymbol{u}_{\text{new}} \leftarrow \mathbf{Steer}(\boldsymbol{u}_{\text{nearest}}, \boldsymbol{u}_{\text{rand}})$
    **if** ObtacleFree($\boldsymbol{u}_{\text{nearest}}, \boldsymbol{u}_{\text{new}}$) **then**
      $\boldsymbol{U}_{\text{near}} \leftarrow \mathbf{Near}(G = (V, E), \boldsymbol{u}_{\text{new}}, R)$
      Add new node: $V \rightarrow V \cup \{\boldsymbol{u}_{\text{new}}\}$
      $\boldsymbol{u}_{\min} \leftarrow \boldsymbol{u}_{\text{nearest}}$
      $\boldsymbol{c}_{\min} \leftarrow \text{getPathCost}(\mathbf{CV}, \boldsymbol{u}_{\min}, \boldsymbol{u}_{\text{new}})$
    **end if**
    **for** $\boldsymbol{u}_{\text{near}} \in \boldsymbol{U}_{\text{near}}$ **do**
      $\boldsymbol{c}_{\text{near}} \leftarrow \text{getPathCost}(\mathbf{CV}, \boldsymbol{u}_{\text{near}}, \boldsymbol{u}_{\text{new}})$
      **if** $\boldsymbol{c}_{\text{near}} < \boldsymbol{c}_{\min}$ **then**
        $\boldsymbol{u}_{\min} \leftarrow \boldsymbol{u}_{\text{near}}$
        $\boldsymbol{c}_{\min} \leftarrow \boldsymbol{c}_{\text{near}}$
      **end if**
    **end for**
    Add new edge: $E \leftarrow E \cup \{(\boldsymbol{u}_{\min}, \boldsymbol{u}_{\text{new}}))\}$
    **for** $\boldsymbol{u}_{\text{near}} \in \boldsymbol{U}_{\text{near}}$ **do**
      $\boldsymbol{c}_{\text{temp}} \leftarrow \text{getPathCost}(\mathbf{CV}, \boldsymbol{u}_{\text{new}}, \boldsymbol{u}_{\text{near}})$
      **if** $\boldsymbol{c}_{\text{temp}} < \boldsymbol{c}_{\text{near}}$ **then**
        $\boldsymbol{c}_{\text{near}} \leftarrow \boldsymbol{c}_{\text{temp}}$
        $\boldsymbol{u}_{\text{parent}} \leftarrow \text{Parent}(\boldsymbol{u}_{\text{near}})$
        $E \leftarrow (E \backslash \{(\boldsymbol{u}_{\text{parent}}, \boldsymbol{u}_{\text{near}})\}) \cup \{(\boldsymbol{u}_{\text{new}}, \boldsymbol{u}_{\text{near}})\})$
      **end if**
    **end for**
    **if** isArrived **then**
      $\text{Parent}(\boldsymbol{u}_j) \leftarrow \boldsymbol{u}_{\text{new}}$
    **end if**
  **end for**
  $\boldsymbol{T} = []$
  **while** $\text{Parent}(\boldsymbol{u}_k) != \emptyset$ **do**
    $\boldsymbol{T}.append(\text{Parent}(\boldsymbol{u}_k))$
    $\boldsymbol{u}_k = \text{Parent}(\boldsymbol{u}_k)$
  **end while**
  **return** $\boldsymbol{T}$

**Algorithm 3** getPathCost, called by Algorithm 2

**Require:** $\mathbf{CV}, \boldsymbol{u}_1, \boldsymbol{u}_2$
  $\mathbf{Cost}_1 = \mathbf{CV}(\boldsymbol{u}_1)$
  $\mathbf{Cost}_2 = \mathbf{CV}(\boldsymbol{u}_2)$
  $\mathbf{Cost}_{path} = \mathbf{Cost}(\boldsymbol{u}1) + ||\boldsymbol{u}_1 - \boldsymbol{u}_2||_2 + (\mathbf{Cost}_1 + \mathbf{Cost}_2)/2 \cdot ||\boldsymbol{u}_1 - \boldsymbol{u}_2||_2$

## 5. SIMULATION RESULTS

To present the performance of the algorithm, we discuss the step-wise behavior of the algorithm to better demonstrate the capability of achieving the information-gathering goal while keeping an eye on the remaining distance budget and avoiding obstacles throughout the entire process.

**Algorithm 4** updateCostValley, called by Algorithm 1

**Require:** $\boldsymbol{m}_j, \boldsymbol{S}_j, \text{Budget}, \boldsymbol{u}_j, \boldsymbol{u}_{j-1}$
  $\textbf{Cost}_{\text{EI}} = \textbf{getEIField}(\boldsymbol{m}_j, \boldsymbol{S}_j)$
  $\textbf{Cost}_{\text{Budget}} = \textbf{getBudgetField}(\text{Budget}, \boldsymbol{u}_j)$
  $\textbf{Cost}_{\text{Obstacle}} = \textbf{getObstacleField}$
  $\textbf{Cost}_{\text{Direction}} = \textbf{getDirectionalField}(\boldsymbol{u}_j, \boldsymbol{u}_{j-1})$
  $\textbf{CostValley} = \sum \textbf{Cost}_{\text{EV,Budget,Obstacle,Direction}}$
  **return CostValley**

**Algorithm 5** getEIField, called by Algorithm 4

**Require:** $\boldsymbol{m}_j, \boldsymbol{S}_j$
  $\textbf{EIBV} = \mathbf{0}^{n\times 1}$
  $\textbf{IVR} = \mathbf{0}^{n\times 1}$
  **for** $i \in 1 \dots n$ **do**
    $\boldsymbol{f}_j = \mathbf{0}^{n\times 1}$, and $\boldsymbol{f}_j[i] = 1$
    $\boldsymbol{R}_j = \boldsymbol{S}_{j-1}\boldsymbol{f}_j(\boldsymbol{f}_j^T\boldsymbol{S}_{j-1}\boldsymbol{f}_j + r^2)^{-1}\boldsymbol{f}_j^T\boldsymbol{S}_{j-1}$
    $\textbf{IVR}[i] = \sum_{i=1}^{n} \textbf{diag}(\boldsymbol{R}_j)$
    $\textbf{EIBV}[i] = \sum_{i=1}^{n} \Phi_2\left(\begin{bmatrix} t \\ -t \end{bmatrix}; \begin{bmatrix} m_{j-1}(i) \\ -m_{j-1}(i) \end{bmatrix}, \boldsymbol{W}_j(i,i)\right),$
    where, $\boldsymbol{W}_j(i,i) = \begin{bmatrix} T(i,i) & -R_j(i,i) \\ -R_j(i,i) & T(i,i) \end{bmatrix}$
    given, $T(i,i) = S_{j-1}(i,i) + R_j(i,i)$
  **end for**
  $\textbf{Cost}_{\text{EI}} = \textbf{norm}(\textbf{EIBV}) + 1 - \textbf{norm}(\textbf{IVR})$
  **return** $\textbf{Cost}_{\text{EI}}$

**Algorithm 6** getBudgetField, called by Algorithm 4

**Require:** $\text{Budget}, \boldsymbol{u}_j$
  Form a budget ellipse with $a, b, c$
  $a = \text{Budget}/2$
  $c = ||\boldsymbol{u}_{goal} - \boldsymbol{u}_j||_2/2$
  $b = \sqrt{a^2 - c^2}$
  $\textbf{Cost}_{\text{Budget}} = \infty^{n\times 1}$
  **for** $i \in 1 \dots n$ **do**
    $\omega = \frac{\boldsymbol{u}_{ix}}{a}^2 + \frac{\boldsymbol{u}_{iy}}{b}^2$
    **if** $\omega <= 1$ **then** $\textbf{Cost}_{\text{Budget}}[i] = 0$
    **end if**
  **end for**
  **return** $\textbf{Cost}_{\text{Budget}}$

**Algorithm 7** getObstacleField, called by Algorithm 4

  $\textbf{Cost}_{\text{obstacle}} = \infty^{n\times 1}$
  **for** $i \in 1 \dots n$ **do**
    **if** $\boldsymbol{u}_i \in$ **Cfree then**
      $\textbf{Cost}_{\text{obstacle}}[i] = 0$
    **end if**
  **end for**
  **return** $\textbf{Cost}_{\text{obstacle}}$

We first show the results of the algorithm for one realized salinity field. We then study the performance of the algorithm over replicate data from the GRF model, where we also compare results with that of a lawnmower algorithm and a myopic exploration algorithm with time operation constraints.

### *5.1 Simulation setup*

The prior mean is produced by Equation (4), with a lower expected salinity near location $\boldsymbol{u} = (u_1, u_2) = (1, 0.5)$

**Algorithm 8** getDirectionalField, called by Algorithm 4

**Require:** $\boldsymbol{u}_{j-1}, \boldsymbol{u}_j$
  $\boldsymbol{b_1} = \boldsymbol{u}_j - \boldsymbol{u}_{j-1}$
  $\textbf{Cost}_{\text{direction}} = \mathbf{10}^{n\times 1}$
  **for** $i \in 1 \dots n$ **do**
    $\boldsymbol{b_2} = \boldsymbol{u}_i - \boldsymbol{u}_j$
    **if** $\boldsymbol{b_1} \cdot \boldsymbol{b_2} >= 0$ **then**
      $\textbf{Cost}_{\text{direction}}[i] = 0$
    **end if**
  **end for**
  **return** $\textbf{Cost}_{\text{direction}}$

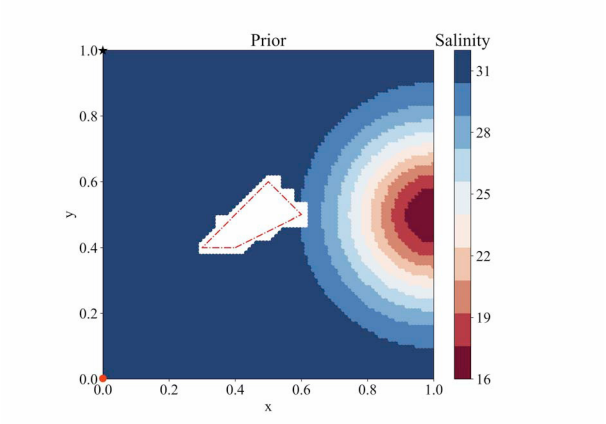


Fig. 2. The prior mean of the salinity at the initial stage.

and visualized in Fig. 2. At each step, the cost valley will be reconstructed based on the overlay of five cost fields including the EIBV cost field, IVR cost field, directional-changing cost field, budget cost field, and obstacle cost field. Then a new tree will be built based on the new cost valley. The ending location will be the place where it has a minimum value in the cost valley. Then the path planner will decide where to go next.

$$\mu_0(\boldsymbol{u}) = 31 - \exp\left(-\frac{(u_1 - 1)^2 + (u_2 - 0.5)^2}{0.07}\right) \quad (4)$$

### *5.2 Simulation results*

In this part, the ground truth is as shown in Fig. 1. Results at several stages of the AUV operation are displayed in Fig. 3 $\sim$ Fig. 5. In each figure, four illustrations are showing various parts of the cost valley calculation. The conditional mean in salinity is shown to the left (in [mg/g]). The normalized cost field based on EIBV and the normalized cost field based on IVR are in the middle. The cost valley with random tree paths for the RRT* algorithm is shown to the right. Over the different data gathering stages $j = 1, \dots, 55$, the sequence of figures shows the balance between different cost fields contributing to the cost valley calculations. The EIBV and IVR cost fields are important when there are no obstacles nearby or still much budget left. EIBV plays a more important role than IVR sometimes when it is more important to exploit, whereas IVR plays a more important role than EIBV when it is more important to explore. When the budget is running out, the other parts of the cost valley calculation are undoubtedly dominating, since it has $\infty$ penalty outside

the budget region. The directional change penalty plays a continuous role in path planning since it guides the agent to move forward with a smooth path, such path planning can avoid sharp turns in practice and hence reduce navigational inaccuracy.

### *5.3 Replicate study*

To remove random effects, 50 replicated simulation results are averaged and shown in Fig. 6. During each iteration, IBV, RMSE, and EV (Expected Variance) are monitored for the comparison of the three strategies including myopic(greedy), RRT*, and pre-scripted lawnmower. Myopic strategy and the RRT* both choose the next candidate locations based on the cost associated with them from the cost valley, whereas the pre-scripted lawnmower just moves along according to its pre-designed paths. The result shows that the RRT* planner outperforms the other two, and all the indicators for RRT* including IBV, RMSE, and EV decrease fastest among the others.

## 6. CONCLUSION

In our simulated case, the agent can explore the field adaptively, and it achieves low spatial variance and precise river plume water classification while avoiding the island obstacle and reaching the destination in time. Via comparison with existing approaches, we learned that the suggested approach has good-quality performance metrics and it satisfies our goal of achieving exploratory path planning with a constraint.

In the future, we aim to conduct similar experiments in the field. This entails fine-tuning the non-myopic path planning strategies requiring faster computation and cautious implementation of such algorithms with a need to be well designed onboard the AUV.

## ACKNOWLEDGEMENT

We acknowledge support from the Norwegian Research Council (RCN) through the MASCOT project 305445 and NTNU AMOS grant No. 223254.

## REFERENCES

Bai, S., Shan, T., Chen, F., Liu, L., and Englot, B. (2021). Information-driven path planning. *Current Robotics Reports*, 2(2), 177–188. doi:10.1007/s43154-021-00045-6.

Berget, G.E., Fossum, T.O., Johansen, T.A., Eidsvik, J., and Rajan, K. (2018). Adaptive sampling of ocean processes using an auv with a gaussian proxy model. *IFAC-PapersOnLine*, 51(29), 238–243. doi: https://doi.org/10.1016/j.ifacol.2018.09.509. 11th IFAC Conference on Control Applications in Marine Systems, Robotics, and Vehicles CAMS 2018.

Binney, J., Krause, A., and Sukhatme, G.S. (2010). Informative path planning for an autonomous underwater vehicle. In *2010 IEEE International Conference on Robotics and Automation*, 4791–4796. IEEE.

Das, J., Harvey, J., Py, F., Vathsangam, H., Graham, R., Rajan, K., and Sukhatme, G.S. (2013). Hierarchical probabilistic regression for auv-based adaptive sampling of marine phenomena. In *2013 IEEE International Conference on Robotics and Automation*, 5571–5578. IEEE.

Fossum, T.O., Fragoso, G.M., Davies, E.J., Ullgren, J.E., Mendes, R., Johnsen, G., Ellingsen, I., Eidsvik, J., Ludvigsen, M., and Rajan, K. (2019). Toward adaptive robotic sampling of phytoplankton in the coastal ocean. *Science Robotics*, 4(27), eaav3041. doi: 10.1126/scirobotics.aav3041.

Fossum, T.O., Eidsvik, J., Ellingsen, I., Alver, M.O., Fragoso, G.M., Johnsen, G., Mendes, R., Ludvigsen, M., and Rajan, K. (2018). Information-driven robotic sampling in the coastal ocean. *Journal of Field Robotics*, 35(7), 1101–1121. doi: https://doi.org/10.1002/rob.21805.

Fossum, T.O., Travelletti, C., Eidsvik, J., Ginsbourger, D., and Rajan, K. (2021). Learning excursion sets of vector-valued Gaussian random fields for autonomous ocean sampling. *The Annals of Applied Statistics*, 15(2), 597 – 618. doi:10.1214/21-AOAS1451.

Hollinger, G.A. and Sukhatme, G.S. (2014). Sampling-based robotic information gathering algorithms. *The International Journal of Robotics Research*, 33(9), 1271–1287.

Hwang, J., Bose, N., and Fan, S. (2019). Auv adaptive sampling methods: A review. *Applied Sciences*, 9(15), 3145.

Karaman, S. and Frazzoli, E. (2011). Sampling-based algorithms for optimal motion planning. *The international journal of robotics research*, 30(7), 846–894.

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.

Suh, J., Gong, J., and Oh, S. (2017). Fast sampling-based cost-aware path planning with nonmyopic extensions using cross entropy. *IEEE Transactions on Robotics*, 33(6), 1313–1326. doi:10.1109/TRO.2017.2738664.

Xiao, C. and Wachs, J. (2022). Nonmyopic informative path planning based on global kriging variance minimization. *IEEE Robotics and Automation Letters*, 7(2), 1768–1775. doi:10.1109/LRA.2022.3141458.

Zhang, Y., Kieft, B., Hobson, B.W., Ryan, J.P., Barone, B., Preston, C.M., Roman, B., Raanan, B.Y., Marin III, R., O'Reilly, T.C., Rueda, C.A., Pargett, D., Yamahara, K.M., Poulos, S., Romano, A., Foreman, G., Ramm, H., Wilson, S.T., DeLong, E.F., Karl, D.M., Birch, J.M., Bellingham, J.G., and Scholin, C.A. (2020). Autonomous tracking and sampling of the deep chlorophyll maximum layer in an open-ocean eddy by a long-range autonomous underwater vehicle. *IEEE Journal of Oceanic Engineering*, 45(4), 1308–1321. doi: 10.1109/JOE.2019.2920217.

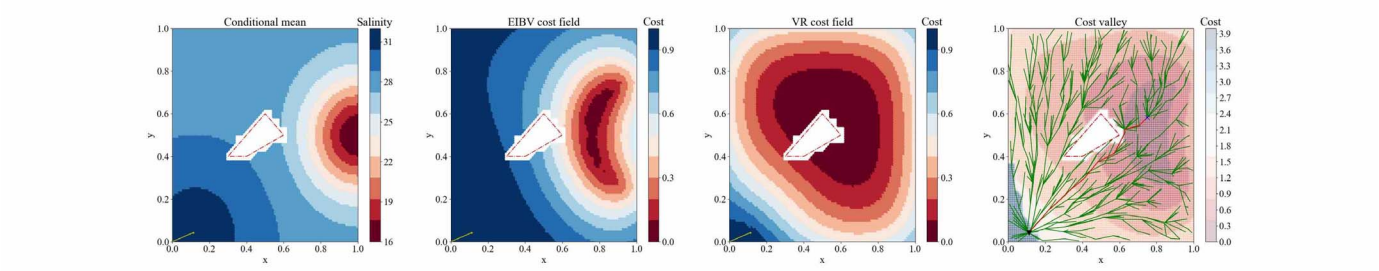


Fig. 3. Step 1, at first, the agent believes the northeast side has very interesting information and hence plans a long almost straight path towards the hotspot. At this step, both EIBV and IVR cost fields play important roles to guide the agent. The trees are distributed in a way that the major stems tend to align along with the low-cost areas.

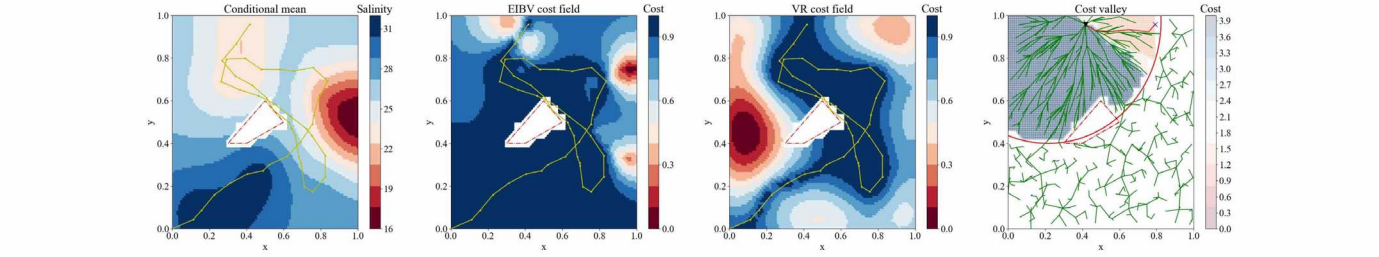


Fig. 4. Step 43, at this stage, the agent moves towards the home direction while still trying to collect as much information as it can. So it tries to reach the boundary since it still has a low cost. The red ellipse shows the remaining distance budget. The penalty outside this budget is infinity, which induces the chaotic behavior of the trees, whereas the trees inside the boundary remain optimally distributed.

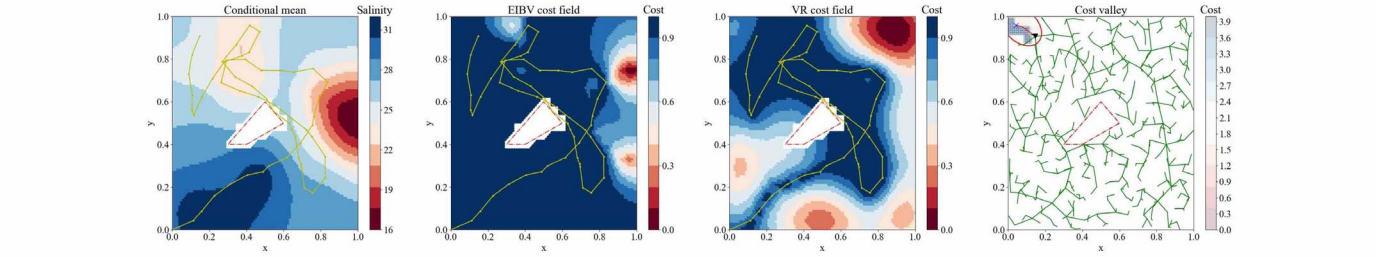


Fig. 5. Step 55, the agent finally decides to go home, and there appears to have a sharp turn in the trajectory since it has to prioritize the mission of going home rather than getting a high penalty from the directional restrictions.

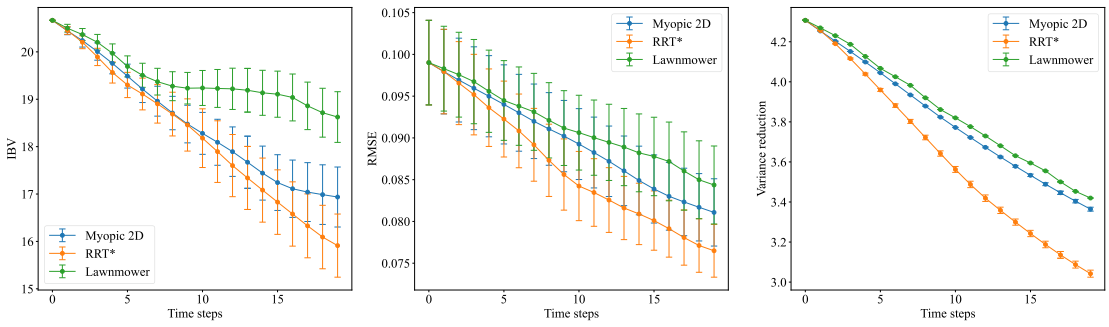


Fig. 6. Simulation results compare myopic, RRT*, and pre-scripted lawnmower strategies in terms of IBV, RMSE, and EV. RRT* performs better than the other two in all of those metrics.