Closing the Data Loop: Real-World AUVs Adaptive Sampling for Improved Ocean Model Predictions

Lucrezia Bernacchi[1], Ana Filipa Duarte[2], Renato Mendes[1,3,4], Leonardo Azevedo[2], João Borges de Sousa[1,3]

[1] *Laboratório de Sistemas e Tecnologia Subaquática (LSTS), Faculdade de Engenharia da Universidade do Porto, Portugal*
[2] *Centro de Recursos Naturais e Ambiente (CERENA), Instituto Superior Técnico, Universidade de Lisboa, Lisboa, Portugal*
[3] *Laboratório Associado de Energia, Transportes e Aeronáutica (LAETA), Porto, Portugal*
[4] *+ATLANTIC CoLAB, Lisboa, Portugal*

2025 Symposium on Maritime Informatics and Robotics (MARIS) | 979-8-3315-1310-8/25/$31.00 ©2025 IEEE | DOI: 10.1109/MARIS64137.2025.11139688

Abstract—**Ocean models' forecasting capabilities heavily depend on the quality of assimilated data. Autonomous Underwater Vehicles (AUVs) are increasingly used to collect data more quickly and cost-effectively, but their endurance limitations make efficient trajectory planning essential to maximize data informativeness. This work presents a cost-effective, multi-vehicle path planning algorithm for adaptive ocean sampling, based on the Prize-Collecting Vehicle Routing Problem (PCVRP) and designed to close the loop between model prediction and assimilation by guiding data collection in data-sensitive regions. The algorithm ensures compliance with key practical constraints typical of real-world missions, including different endurance capabilities of the vehicles, varying deployment and recovery locations, and obstacle avoidance within the operational area. The approach follows a daily cycle: the ocean model generates a one-day forecast, producing a spatial uncertainty map that serves as input to the planning algorithm; the algorithm determines the near-optimal sampling allocations for the vehicles, aiming to maximize the collected information and minimize the travelled distance; finally, the collected data are assimilated to improve the next forecast. The methodology has been tested and validated on a fleet of AUVs in Nazaré, Portugal, within the context of the FRESNEL field campaign conducted in October 2024. Experimental results demonstrate the effectiveness of the proposed methodology in real-world contexts and the benefits of the adaptive sampling strategy in enhancing model capabilities.**

Index Terms—**Adaptive sampling, autonomous underwater vehicle (AUV), ocean modeling, vehicle routing problem (VRP).**

I. Introduction

Ocean models are essential tools for predicting the future state of the oceans, which is critical for climate forecasting, pollution monitoring, and resource management. However, the inherent imperfections of these models make the data assimilation process vital for compensating for model errors and improving their forecast capabilities. This is achieved by integrating real-world observational data with the physical model, resulting in a more accurate representation of the system's state [1]. Diverse technologies are used to collect these data. Satellite data, while highly advantageous for providing data of vast areas, are subject to low spatial resolution and depend on atmospheric conditions. On the other hand, in situ measurements, such as those obtained using ships or buoys, provide more accurate datasets for ocean models. Still, they face limitations in terms of operational cost and the area they can cover, thus limiting the data to point-based measurements in specific locations. Recent developments in technology and automation have made Autonomous Underwater Vehicles (AUVs) an efficient tool for data collection. These vehicles, equipped with different sensors and the ability to move, can improve both the spatial and temporal resolution of a wide range of oceanographic data [2]. However, the endurance limitations of these vehicles make intelligent trajectory planning crucial for focusing on areas of greater relevance for ongoing analysis. Moreover, the high variability of ocean dynamics and the influx of assimilated data necessitate continually refining sampling strategies to adapt to changing conditions. This opens the door to adaptive sampling strategies, where the goal is to predict the types and locations of observations expected to be most useful based on given objectives and the constraints of the available assets [3]. The objectives, targets, and platforms involved in adaptive sampling strategies vary widely, leading to several approaches tailored to specific applications in the context of AUVs [4]. These include, among others, seabed coverage for mine countermeasures [5], sampling for water mass classification [6], tracking coastal upwelling and river plume fronts [7], [8] and ocean phenomena [9], [10]. In [11], a Gaussian Process-based adaptive sampling method for a single AUV is introduced to explore an environment while prioritizing the most informative regions, defined by different potential applications. In [12], a recursive-greedy algorithm is employed to sample the operational area while considering vehicle endurance constraints and high-traffic zones. Both studies focus on single-vehicle scenarios, and a prior definition of waypoints, in number or location, is used for trajectory planning. A multi-vehicle approach is explored in [13], presenting an adaptive sampling strategy for multiple AUVs that utilizes virtual bodies and artificial potentials. This method focuses on cooperative control, with multiple vehicles maintaining formation while sampling the ocean, as tested during the OASN II project [14]. A promising adaptive sampling technique driven by an ocean model is proposed in [15], intending to minimize the error standard deviation of a 2D ocean field to improve ocean predictions using Mixed Integer Linear Programming (MILP). Given the initial deployment positions and the desired trajectory lengths for the vehicles, the method provides an optimal solution that accounts for multiple operational constraints. While the reported computation times for simulations are promising, the solution time increases exponentially with the desired trajectory length, making the approach sensitive to the problem's scale. Despite these advancements, many proposed approaches struggle to address the complexities of real-world deployments, where operational areas vary in size and characteristics. This variability often results in computational challenges and the introduction of essential operational constraints, limiting the effectiveness of existing methods. To address these gaps, this work proposes an operational approach for adaptive sampling designed for real-world applications to close the loop between forecasting and data assimilation to enhance the predictive capabilities of ocean models. Our approach introduces a cost-effective algorithm for computing near-optimal AUV trajectories, accommodating varying fleet sizes while addressing key practical constraints typical of real-world missions. These constraints include different endurance limitations for each vehicle, distinct deployment and recovery locations, known obstacles within the operational area, and the time required by the AUVs to ensure communication and navigation fixes during the missions.

The paper is organized as follows: Section II details the problem under investigation. Section III introduces the key steps of the proposed approach. Section IV describes the experimental activities conducted in both simulation and real-world scenarios and the obtained results. Finally, Section V summarizes the findings and explores potential future developments.

II. Problem Statement

In this work, we focus on enhancing the capabilities of numerical ocean models by leveraging AUVs to collect oceanographic data. The goal is to implement an adaptive sampling strategy that maximizes the informativeness of the data collected by AUVs, enhancing the accuracy of the model's predictions through data assimilation. This will ultimately lead to better-informed AUV missions, closing the data loop illustrated in Fig. 1, and consisting of three fundamental steps iteratively executed on a daily basis:

1) Model output: A daily one-step forecast $\hat{\theta}(k+1,x,y)$ of an oceanic variable θ and its associated uncertainty $\sigma_{\hat{\theta}}(k+1,x,y)$ is computed by the model, where k is the current day and x and y correspond to longitude and latitude values. The outputs of this model are presented as discrete spatial maps, $\mathcal{M}_{\hat{\theta}}(x,y)$ and $\mathcal{M}_{\sigma_{\hat{\theta}}}(x,y)$, covering the geographical area under study, and consisting of gridded representations where each grid cell corresponds to a specific geographic location (x,y), with values indicating the respective output under consideration.

2) Adaptive sampling algorithm: Given as input the spatial map corresponding to the uncertainty associated with the one-day prediction $\mathcal{M}_{\sigma_{\hat{\theta}}}(x,y)$ and a fleet of N vehicles equipped with sensors capable of point-wise sampling the environmental quantity θ, the objective is to determine the set of vehicle trajectories for the following day, as a series of waypoints, that maximize the accumulated uncertainty along the paths while adhering to the operational constraints of the vehicles.

3) Data assimilation: At the end of the day, the measurements collected along the vehicle trajectories are assimilated into the model and a new forecast *1)* is generated.

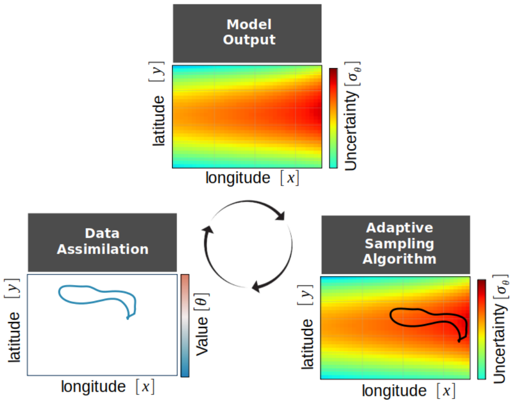


Fig. 1. Data loop closure: (1) model output generation - $\mathcal{M}_{\sigma_{\hat{\theta}}}(x,y)$ uncertainty map associated with the one-day forecast of an oceanic variable θ; (2) adaptive sampling - optimization of vehicle trajectories to maximize uncertainty collection; and (3) data assimilation - integration of vehicle measurements into the model.

Concerning the adaptive sampling algorithm, the problem can be generalized and formally stated as follows:

$$\mathcal{T} = \max \sum_{i=1}^{N} \sum_{j=0}^{L_i} \mathcal{M}_I(x_{ij}, y_{ij}) \tag{1}$$

Given a discrete spatial map $\mathcal{M}_I(x,y)$ containing information I related to a variable, the objective is to generate a set of trajectories $\mathcal{T} = \{T_1, \ldots, T_N\}$, where N is the number of vehicles, aimed at maximizing the total information I collected along the paths of length L_i.

In this formulation, (x_{ij}, y_{ij}) represents the coordinates of the $j-th$ point along the trajectory of the $i-th$ vehicle. In contrast, $\mathcal{M}_I(x,y)$ can be interpreted in different ways according to the specific objectives of the problem. However, in our case, it corresponds to the spatial map containing the uncertainty values associated with the one-day forecast, $\mathcal{M}_{\sigma_{\hat{\theta}}}(x,y)$.

In order to make the problem relevant from an application perspective, it is subject to the following operational constraints:

- Each trajectory has a maximum length $Lmax_i$, determined by the mission's endurance capacity of each vehicle:
$$L_i \leq Lmax_i = v_i \times Tmax_i(v_i) \quad \forall i \tag{2}$$
Where v_i represents the desired vehicle's velocity for the mission and $Tmax_i$ represents the maximum operational time achievable at that velocity.
- The starting and ending locations of the trajectories differ for each vehicle, providing flexibility in the choice of deployment and recovery locations:
$$(x_{i0}, y_{i0}) \neq (x_{iLi}, y_{iLi}) \quad \forall i \tag{3}$$

The introduction of this constraint also allows for planning trajectories over multiple days, with the vehicle's position at the time of the next planning session serving as the starting point.

- The trajectories must avoid static obstacles within the operational area to prevent collisions with potentially hazardous elements such as rocks, shallow water, nets, or unknown objects:

$$[(x_{ij}, y_{ij}), (x_{i(j+1)}, y_{i(j+1)})] \notin \mathcal{O} \quad \forall i, \forall j \tag{4}$$

Where $\mathcal{O}$ is the region within the operational area occupied by obstacles, which is assumed to be known a priori.

- The vehicles can spend an optional time, t_{wp}, at each waypoint along the trajectory for communication and navigation correction purposes.

III. Adaptive Sampling Planner

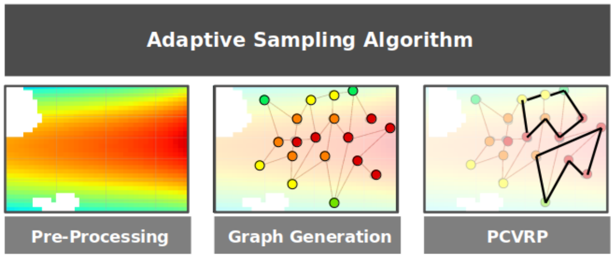


Fig. 2. Workflow of the adaptive sampling algorithm developed by the authors: The input map $\mathcal{M}_I(x, y)$ undergoes obstacle mapping and filtering; The map is transformed into a weighted graph, where grid cells serve as nodes representing potential waypoints, with assigned rewards; The PCVRP is solved to generate near-optimal trajectories.

To solve the described problem, we propose a method based on graph optimization. The challenge of finding the N trajectories that maximize the collected information in $\mathcal{M}_I(x, y)$ while adhering to the operational constraints is decomposed into three main steps, shown in Fig. 2, and summarized below:

- A pre-processing step is applied to the input map for filtering and obstacle mapping.
- By adopting a points-of-interest selection procedure, the spatial map $\mathcal{M}_I(x, y)$ is transformed into a weighted graph, where grid cells serve as nodes representing potential waypoints for the vehicle trajectories. The node weights define the reward of visiting a specific node, while the edge weights represent the cost of traveling between nodes.
- We solve the Prize Collecting Vehicle Routing Problem (PCVRP) [16] to determine how to connect the selected nodes to generate the desired trajectories, maximizing the reward and minimizing the travelled distance while adhering to the operational constraints.

In the following paragraphs, the three steps are explained in detail.

A. Pre-processing

The safety of vehicles is a priority when executing a mission. Since the operational area is rarely free of obstacles, their identification is crucial for safe navigation and mission success. These obstacles can vary in geometry and nature and can be subject to temporal variations:

- Morphological obstacles include natural elements such as landmasses, intertidal zones and submerged or emergent rocks.
- Anthropogenic obstacles may include fishing nets, which can easily entangle vehicles, and high-traffic maritime zones where the risk of collisions is considerably high.

The first step of the algorithm involves identifying these obstacles by utilizing satellite imagery, bathymetric data, maritime traffic flow analysis, and information gathered from local sources. Upon identification, the obstacles are masked on the map provided as input to the algorithm by assigning a $null$ value to any grid cell that intersects with the obstacle, even partially, enabling a complete and safe representation of the locations of the obstacles and the free space for navigation. Although the mentioned obstacles can be subject to temporal variability, this was not considered in our approach.

The pre-processing step also includes filtering the input map. A Gaussian filter is applied to smooth the map while preserving its main features. The standard deviation along the x and y axes is kept as a tunable parameter of the filter, defined according to the specific characteristics of the input map.

B. Graph Generation

The second step of the algorithm involves the generation of the weighted graph $G(V, E, W_v, W_e)$ from the input map $\mathcal{M}_I(x, y)$, where V is the set of nodes, E is the set of edges, and W_v and W_e represent the weights of the nodes and edges, respectively. The first goal of this phase is to identify key points within the map corresponding to potential waypoints, which will serve as nodes of the graph on which optimal planning will subsequently be performed. What criteria should be used to select the nodes? The first intuition might be to consider every central point of each grid cell. However, this would result in an excessively large graph, increasing the computational time required to find a solution and potentially making the problem unsolvable for large areas. Another possible approach could be to select only a certain number of the most informative points on the map, thus constraining the planning to the most uncertain areas. However, this method could select zones far from the desired vehicle deployment locations, resulting in significant energy consumption to reach those areas.

Therefore, it is important to select the nodes in a way that ensures good area coverage while keeping their number limited to ensure the scalability of the proposed method for different operational areas. The proposed solution combines contour lines and Voronoi diagrams for selecting the graph nodes, as shown in Fig. 3.

Contour lines - By applying contour lines to the spatial

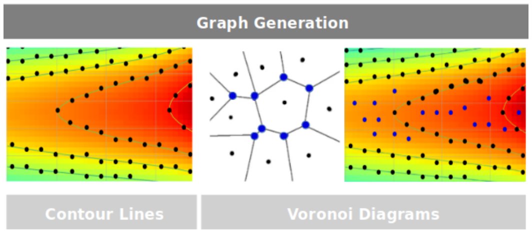


Fig. 3. Nodes selection: Contour lines and set of initial nodes V1 (black) - Voronoi diagram applied to V1 and Voronoi vertices V2 (blue).

map, it is possible to identify the spatial distribution of the uncertainty field. These patterns serve as the basis for generating the initial set of graph nodes V_1. Once the contour lines are defined, each point along the curves is approximated to the center of the nearest grid cell. Graph nodes are then selected from these points, ensuring a minimum distance d_{min} is maintained between adjacent nodes. This approach allows us to keep the number of nodes manageable while accurately representing the contour lines. Therefore, each point p_i belonging to the set of contour lines C is included in the node set V_1, if it satisfies the following condition:

$$d(p_i, p_k) \geq d_{\min} \quad \forall p_k \in V_1, \quad \forall p_i \in C \tag{5}$$

Voronoi diagrams - Contour lines can take on highly varied shapes. Voronoi diagrams are employed subsequently to ensure better coverage and guarantee the presence of nodes even within the areas enclosed by the level curves. Starting with a set of generator points, the Voronoi diagram is a partition of the plane into cells, where each cell contains all the points closer to a specific generator than any other. The vertices of the Voronoi diagram, on the other hand, are the points where the edges of at least three Voronoi cells meet. In other words, a Voronoi vertex is a point equidistant from three or more generators. For this reason, these vertices are ideal candidates for enhancing coverage within the areas inside the level curves. Considering the set of nodes V_1 as generator nodes, the Voronoi diagrams are computed, and a new set of nodes V_2 is defined from the vertices of the diagram. Also in this case, the condition expressed in equation (5) is applied and only the vertices that maintain a minimum distance d_{min} from the already selected graph nodes and the generated vertices will be added to the new set V_2.

The result of this phase is a graph in which the nodes $(V_1 \cup V_2)$ serve as candidate waypoints for trajectory planning. The definition of the weights for the nodes and edges is closely related to the trajectory optimization and will be detailed in the following subsection.

C. Prize-Collecting Vehicle Routing Problem

This step aims to determine how to connect the selected nodes to form the N trajectories that maximize the collected information while respecting the operational constraints. This problem belongs to the broader family of Vehicle Routing Problems (VRP) [21], a popular class of combinatorial optimization problems, generalization of the Traveling Salesman Problem (TSP) to the multi-vehicle case. The VRP's objective is to determine the optimal set of routes to be performed by a fleet of vehicles to serve a given set of customers. The route set is optimal if it meets all customer requirements and operational constraints while minimizing the total travel cost. This cost can be measured in economic terms, distance traveled, or other factors, depending on the specific context of the application. In particular, the problem addressed in this work is formulated with the Prize Collecting Vehicle Routing Problem (PCVRP) variant [16]. In this variant, it is not mandatory to visit all customers, and each customer has a prize that is collected when the customer is visited. The goal is to balance the competing goals of minimizing travel costs while maximizing the collected prizes and satisfying the operational constraints. To formulate our problem as a PCVRP we define the customers as the identified graph nodes, the travel costs as the cumulative distance between the nodes of the routes, and the rewards for visiting each customer, based on nodes' respective uncertainty values defined by the input map. The problem is, therefore, formulated as follows:

Given $\mathcal{R}$, the set of non-empty routes and V, the set of nodes, each route $r \in \mathcal{R}$ is defined as a sequence of nodes $V_r \in V$ with a specific starting and ending point. The objective function to be minimized is given by:

$$J = \min(\sum_{r \in \mathcal{R}} d_r + \sum_{i \in V} p_i - \sum_{r \in \mathcal{R}} \sum_{i \in V_r} p_i) \tag{6}$$

Where the first term is related to the total distance of each route d_r, while the second and third terms together represent the uncollected prizes of unvisited nodes, with p_i the prize associated with each node. The operational requirement (2) on the maximum distance $Lmax_i$ traveled by each vehicle is introduced by adding a penalty term to the cost function:

$$\lambda \cdot \max(0, d_r - L_{\max_i}) \tag{7}$$

In this way, if a route's distance exceeds the limit, its cost increases significantly by the penalty coefficient λ, discouraging such solutions. The process of assigning distance costs and prizes is translated into weighting the edges and nodes of the graph, as explained below.

Weights assignment - Edge weights are used to define the cost of the path between two nodes. Assuming we are operating in areas of sufficiently small size such that the curvature of the Earth can be neglected without significant errors, the distance between nodes is calculated using the Euclidean distance. However, to meet the operational requirement (4) outlined in the problem definition, which aims to ensure obstacle-free trajectories, this distance is set to infinity if the line connecting two nodes intersects an obstacle. This approach drastically increases the cost function (6), effectively guiding the optimization towards obstacle-free trajectories. The weights of the edges are assigned as follows:

$$W_e(e_{ij}) = d_{ij} \quad \forall i, j \in \mathcal{G}(V) \tag{8}$$

$$\text{where} \quad d_{ij} = \begin{cases} \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} & \text{if } \overline{IJ} \notin \mathcal{O}, \\ \infty & \text{if } \overline{IJ} \in \mathcal{O} \end{cases} \tag{9}$$

Node weight represents the reward p_i obtained by visiting a specific node and is evaluated to determine which nodes are worth visiting and which should be avoided. Therefore, given the goal of maximizing information, the reward associated with a node at coordinates (x_k, y_k) is assigned proportionally to the uncertainty value of that node, $I(x_k, y_k)$, as identified by the spatial map in input:

$$W_v(v_k) \propto I(x_k, y_k) \quad \forall k \in \mathcal{G}(V) \tag{10}$$

Finally, the constraint regarding the time t_{wp} that the vehicles need to spend at each waypoint is considered by solving the optimization problem twice. Since the total time T_{wp} each vehicle will spend at the waypoints during the mission depends on the number of waypoints on the route, the first time, the problem is solved just to estimate this number. The second time, the final optimization is obtained considering the maximum operational time $Tmax$ in (2), subtracted by the estimated total time spent at the waypoints, T_{wp}.

Solver - VRP problems are known to have NP-hard complexity, meaning that finding an optimal solution for large instances becomes infeasible due to the exponential growth of the computation time needed to solve them. Numerous heuristics and metaheuristics methods have been developed to obtain sub-optimal solutions. In this work, we use the VRP solver PyVRP [20], which provides a high-performance implementation of the Hybrid Genetic Search (HGS) algorithm presented in [17].

IV. Experiments

In this section, we present the experimental activities carried out. First, a simulation testing phase was conducted using data from an ocean model to evaluate the algorithm's feasibility under varying conditions, including mission duration, fleet size, obstacles, and deployment and recovery positions. Subsequently, the approach was tested and validated in a real-world context during the FRESNEL (Field expeRiments for modEling, aSsimilatioN, and adaptive sampLing) field campaign, where the full loop closure, presented in Section II, was implemented daily.

A. Simulation

The scenario used to test the algorithm is based on the Harvard Ocean Prediction System (HOPS) model [18] for the Setúbal Bay, south of Lisbon, Portugal. That scenario used a similar model implementation that was then applied for the FRESNEL field campaign [19].

This model implementation presents a grid spatial resolution of $700m$. The temperature forecast error field, normalized by the variable's variance within the domain, serves as the spatial map, $\mathcal{M}_I(x, y)$, for testing the algorithm. From this map, several sub-areas were selected to assess the algorithm's performance in operational regions with varying morphological characteristics and different error patterns. Some examples of these tests are presented below. For each simulation, the standard deviation values $\sigma_x = \sigma_y = 1$ were used for filtering the input map, while the d_{min} value was selected from an operational perspective to have a minimum distance between waypoints equal to $1km$ for simulation a) and b) and $2km$ for simulation c). Land masses, along with fictitious objects, were mapped as obstacles. The mission parameters provided as input to the algorithm, selectable based on each mission's specific needs and requirements, are reported in Table I. The resulting trajectories, generated by the algorithm, and their specifications are presented in Fig. 4. The simulations were conducted considering different deployment and recovery locations for each vehicle. These were not reported in the table but in the images below, with stars indicating the starting points. The x and y axes indicate grid cell indices, with a resolution of $700m$ per cell, while the value of each cell represents the temperature forecast error field e_T. As shown in Fig. 4, the generated trajectories head towards areas with greater error, aiming to maximize the collected values while adhering to the desired mission durations. Indeed, the results show a deviation of approximately 10 minutes from the target duration across all missions. To effectively compare the desired duration with that of the generated trajectories, the time spent at each waypoint, t_{wp}, was set to zero in these simulations. In simulations a) and b), trajectories avoiding the mapped morphological obstacles can be observed. To further evaluate the algorithm's ability to generate obstacle-free trajectories, fictitious obstacles were introduced in the center of the operational area, as shown in simulation c). These obstacles are handled as described in (9), and the results confirm the effectiveness of the algorithm in generating obstacle-free trajectories.

TABLE I
Input algorithm - Desired Mission Parameters

	Fleet size	**Speed [m/s]**	**Duration [h]**
a)	2 AUVs	1	16 - 10
b)	3 AUVs	1	15 - 21 - 9
c)	1 AUV	1.2	12

The hardware exploited to produce the results is an Intel® Core™ i7-1065G7 CPU @ 1.30GHz with 16GB of RAM. Table II presents the number of nodes and the computational costs of the algorithm for each simulation, with the PyVRP Solver configured to stop after 20 iterations without improvement in the best solution, as this value provides a good balance between solution quality and computation time for the specific case study. However, different values can be used to achieve shorter computation times.

TABLE II
Simulation Computational Times

	a	**b**	**c**
Nodes	219	235	132
Time [s]	81	125	22

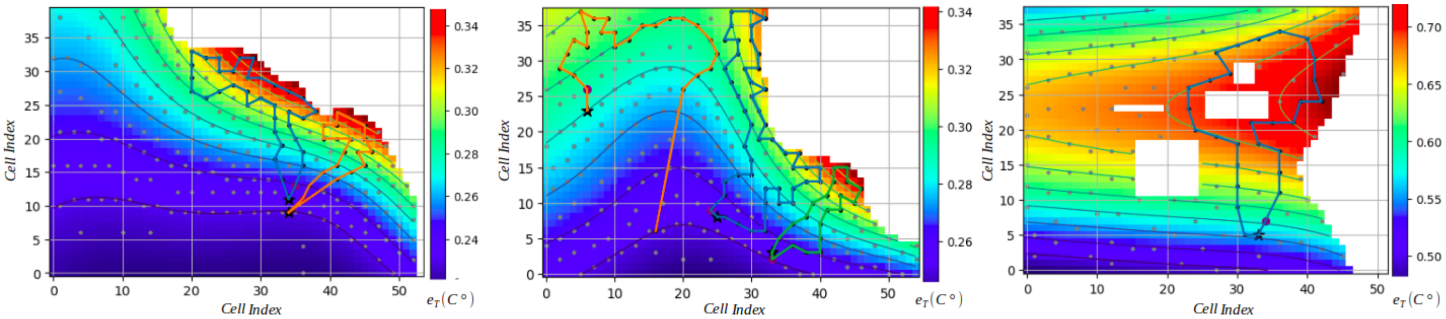


Fig. 4. Adaptive sampling algorithm results: **a)** 2 AUVs – 1 [m/s] speed – 15[h] 53[min], 9[h] 48[min] mission durations – 17.84 [°C] total reward; **b)** 3 AUVs – 1 [m/s] speed – 20[h] 53[min], 14[h] 51[min], 8[h] 52[min] mission durations – 28.92 [°C] total reward; **c)** 1 AUV – 1.2 [m/s] speed – 11[h] 53[min] mission duration – 20.06 [°C] total reward.

B. *FRESNEL field campaign*

The proposed approach was tested and validated in a real-world context during the FRESNEL field campaign. The goal of the campaign was to close the observe-assimilate-predict-sample loop by demonstrating the applicability of adaptively controlled marine robots in sampling the water column, driven by ocean models with increasing predictive skill. The experimental campaign took place in the Nazaré area (Portugal), in October 2024 and for two weeks, it involved the deployment of multiple assets to collect oceanographic data, including AUVs, an oceanographic vessel, and gliders.

Asset - The asset deployed for the adaptive sampling strategy presented in this work consisted of a fleet of 3 Light Autonomous Underwater Vehicles (LAUVs), Fig. 5, supported by the open-source LSTS software toolchain developed by the Laboratório de Sistemas e Tecnologia Subaquática (LSTS) [22]. The toolchain comprises three key components: DUNE, the onboard vehicle software responsible for navigation, control, and communications; NEPTUS, the command and control software used for mission execution and analysis; and RIPPLES, the web infrastructure designed to enhance situational awareness, facilitate communication, and provide a shared operational overview for all participants in the field campaign. Finally, IMC [23], the communication protocol used for exchanging information between vehicles and sensors.

The vehicles were equipped with CTD (Conductivity, Temperature, and Depth) sensors to sample the water column throughout the mission, while the communication sensor suite included an Iridium modem for satellite communication, as well as WiFi, GSM, and acoustic modems.

Fig. 5. FRESNEL field campaign: LAUVs-Xplore during deployment operations and mission execution.

Models - The planner was tested using two different models for the assimilation and prediction of ocean variables. A numerical model (NM) based on the HOPS model implementation in the Nazaré Canyon area of influence [19], was configured to cover the area from $8.9\,°W$ to $10\,°W$ and from $39.2\,°N$ to $39.9\,°N$, with a horizontal grid cell resolution of $300m$ in both latitude and longitude and a vertical discretization based on 30 double-sigma vertical levels.

Additionally, a statistical model (SM) [25], which leveraged 14 days prior information, Copernicus Marine Environment Monitoring Service (CMEMS) products [27], and Direct Sequential Simulation [26], was used to spatially predict ocean variables for the following day. The model covered an area from $11.75\,°W$ to $8.43\,°W$ and from $39\,°N$ to $41.16\,°N$ with a horizontal cell resolution of $7km$ in latitude and $9km$ in longitude, while a vertical discretization based on 17 levels at constant depth.

Operational Setup - A sub-area of the model grids, centered on the Nazaré Canyon and shown in blue in Fig. 6, was selected for LAUVs operations. Following an analysis of maritime traffic density using AIS data from October, the orange area, covering approximately $67.4\ km^2$, was identified as a low traffic zone and therefore designated as the operational area for adaptive sampling missions for safety reasons. Meanwhile, the purple area, closer to the coast, was used for deployment and recovery procedures. In accordance with the 3D characteristics of the models and in order to collect data throughout the water column, the movement of the AUVs along the algorithm-generated trajectories was defined as a

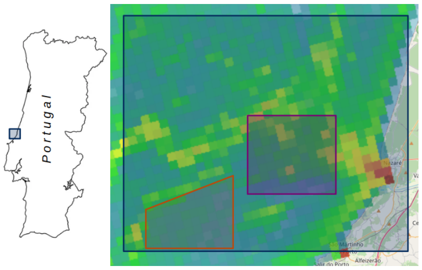


Fig. 6. Snapshot from Ripples showing AIS data density for the month of October in the study area (blue) along with the operational area for adaptive sampling (orange) and the deployment and recovery area (purple).

'yo-yo', varying its depth between the surface and a safe depth while reaching each mission waypoint.

The procedure followed to execute the missions is presented in Fig. 7. Once a new one-step forecast became available, the adaptive planning algorithm was executed to generate the missions for the next day; the plans were transferred to Ripples and transmitted via satellite to the vehicles; CTD profiles were collected during the missions and, upon the vehicles' recovery, the data were assimilated into the models, enabling the generation of a new forecast.

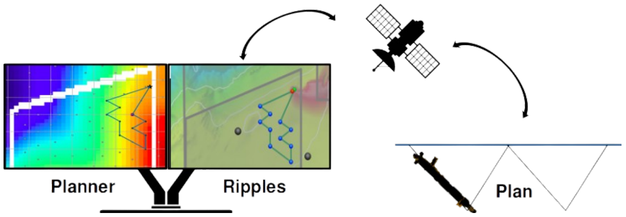


Fig. 7. Operational workflow for mission execution.

Adaptive Sampling Planner - The spatial maps $\mathcal{M}_I(x,y)$ selected as input for the planner consisted of the temperature forecast error field (from HOPS) and the temperature forecast standard deviation field (from SM). To account for the 3D characteristics of the models, the 2D map used as input to the algorithm was derived by computing, for each cell of the plane, the average error/uncertainty value across the depth levels actually covered during the vehicle's mission. Since the planning was constrained to the operational area described above, and no other obstacles were present, the ability of the algorithm to account for obstacles was exploited by defining an artificial barrier within which the planning was confined. A value of $\sigma_x = \sigma_y = 1$ was used for the Gaussian filter applied to the NM map, while a value of $\sigma_x = \sigma_y = 3$ was applied to the SM map after performing an interpolation to achieve the same cell resolution as the NM. For all missions, a minimum waypoint distance of $d_{min} = 1\ km$ was selected to ensure sufficient space for the vehicle to perform yo-yo motion between waypoints and a time $t_{wp} = 1\ min$ was requested to be spent at each waypoint for communication and navigation fix purposes. The remaining algorithm input parameters used to generate missions according to the desired specifications are reported in Table III for each day.

TABLE III
INPUT ALGORITHM - DESIRED MISSION PARAMETERS

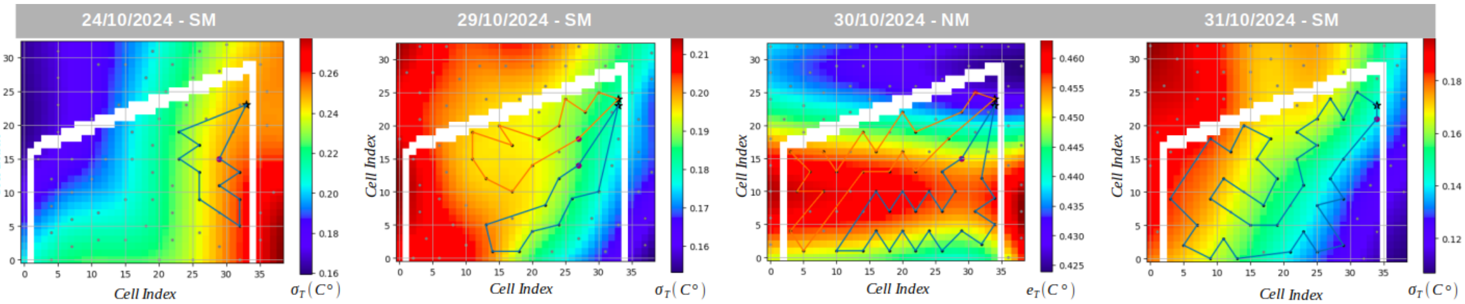
Mission Date	Fleet size	Speed [m/s]	Duration [h]
24/10/2024	1 AUV	1	5
29/10/2024	2 AUVs	1	6 - 6
30/10/2024	2 AUVs	1	10 - 10
31/10/2024	1 AUV	0.9	16

In Fig. 8, the corresponding trajectories generated by the algorithm and their specifications are shown. Both the x and y axes indicate grid cell indices, with a resolution of 300 m per cell, while the value of each cell represents either the temperature forecast error field e_T or the temperature forecast standard deviation field σ_T on the day of the mission. The algorithm successfully generated missions that met the required timeframes, with the reported durations also including the total time spent at each waypoint of the routes. To minimize computational time, the estimation of the number of waypoints was conducted by setting a fixed solver stop time of 60 s, while the second optimization followed the same stop criteria as in the simulation case. On the 24th and 29th, where the mission duration was more limited, the algorithm's ability to generate trajectories driven by the uncertainty values becomes particularly evident.

An example of the temperature values collected during the mission on the 24th by the onboard CTD sensor is shown in Fig. 9, where the characteristic yo-yo motion performed by the vehicle throughout the mission can also be observed. These data represented the input for the assimilation process. Analyses were conducted to assess the benefits of AUV temperature data assimilation in enhancing the model's forecasting ability. Specifically, we examined the one-step predictions made by the SM model regarding the temperature values for the 30th at various depth levels, both with and without the assimilation of data collected during the LAUV mission on the 29th. By comparing these two predictions with the actual values collected by the LAUV on the 30th, an RMSE of 0.6752 resulted for the case without assimilation, while the case with assimilation yielded an RMSE of 0.4012, highlighting the effectiveness of this approach in improving the model's forecasting capability.

V. Conclusion and Future Works

This work introduces a multi-vehicle path planning algorithm for adaptive ocean sampling based on PCVRP, aimed at maximizing the informativeness of the data collected by AUVs, with the goal of enhancing the accuracy of ocean model predictions through the data assimilation process, ultimately closing the loop between assimilation and prediction. The proposed approach was tested in both simulations and real-world deployments. The results demonstrate the algorithm's ability to generate sub-optimal trajectories that maximize the collected uncertainty values along the paths in a cost-effective manner for different operational scenarios. It has proven effective for fleets of varying sizes, accommodating different desired deployment and recovery positions while ensuring compliance with each vehicle's endurance specifications. Additionally, the algorithm is applicable in operational areas that contain obstacles, ensuring the generation of collision-free trajectories. Although the study included real-world deployments, logistical constraints and inherent risks limited the number of tests conducted and the number of consecutive assimilation days. Future developments will focus on extending this approach to account for the three-dimensional nature of ocean models, generating trajectories in three-dimensional space. Furthermore, given the algorithm's promising computational efficiency, it will be adapted for online use on the vehicles, enabling near real-time data exchange between the vehicle and the model. Re-planning trajectories throughout the mission as


24/10/2024 - SM

29/10/2024 - SM

30/10/2024 - NM

31/10/2024 - SM

Fig. 8. Adaptive sampling algorithm results: **24/10)** 1 AUV – 1 [m/s] speed – 4[h] 53[min] mission duration – 17 [km] planar traveled distance; **29/10)** 2 AUVs – 1 [m/s] speed – 5[h] 42[min], 5[h] 53[min] mission durations – 19.95 [km], 20.40 [km] planar traveled distances; **30/10)** 2 AUVs – 1 [m/s] speed – 9[h] 24[min], 9[h] 32[min] mission durations – 32.48[km], 33.23 [km] planar traveled distances; **31/10)** 1 AUV – 0.9 [m/s] speed – 15[h] 45[min] mission duration – 49.21 [km] planar traveled distance.

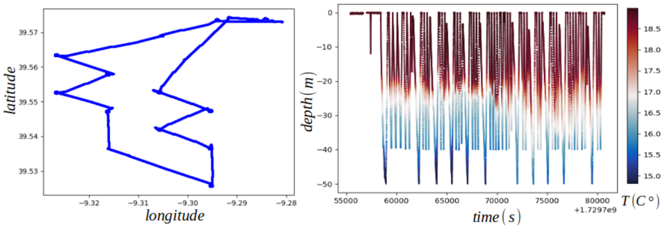


Fig. 9. LAUV Xplore trajectory (left) and temperature data collected during the mission (right).

new predictions become available will allow for taking into account ocean variability during the sampling activities. In this context, the dynamic nature of the real environment will also be addressed, including the incorporation of dynamic obstacles to ensure vehicle safety throughout the mission.

VI. Acknowledgments

This work was supported by FRESNEL project, funded by the Office of Naval Research (ONR) under ONR award number N00014-22-1-2796, as well as SMART (diStributed AI systeM for mArine plastic debRis moniToring) project, funded by the Portuguese Space Agency in the framework program for research and innovation. The work was also supported by JUNO—Robotic Exploration of Atlantic Waters project—Refª 2021/0008—from FLAD and by the DiverSea project, supported by the European Commission under grant agreement 101082004.

References

[1] Anderson DL, Sheinbaum J, Haines K., ”Data assimilation in ocean models,” Reports on Progress in Physics, 1996 Oct.

[2] Wynn RB, et al. ”Autonomous Underwater Vehicles (AUVs): Their past, present and future contributions to the advancement of marine geoscience,” Marine geology, 2014 Jun.

[3] Lermusiaux PF. ”Adaptive modeling, adaptive data assimilation and adaptive sampling,” Physica D, 230(1-2):172-96, 2007 Jun.

[4] Hwang J, Bose N, Fan S. ”AUV adaptive sampling methods: A review,” Applied Sciences,9(15):3145, 2019 Aug.

[5] Paull L, Saeedi S, Seto M, Li H. ”Sensor-driven online coverage planning for autonomous underwater vehicles,” IEEE/ASME Transactions on Mechatronics, 18(6):1827-38, 2012 Sep.

[6] Ge Y, Eidsvik J, Mo-Bjørkelund T, ”3-D adaptive AUV sampling for classification of water masses,” IEEE Journal of Oceanic Engineering, 48(3):626-39, 2023 Apr.

[7] Zhang Y, Godin MA, Bellingham JG, Ryan JP, ”Using an autonomous underwater vehicle to track a coastal upwelling front,” IEEE Journal of Oceanic Engineering, 37(3):338-47, 2012 Jun.

[8] Teixeira D, de Sousa JB, Mendes R, Fonseca J,”3D Tracking of a River Plume Front with an AUV,” OCEANS 2021: San Diego–Porto, 2021.

[9] Costa, Maria Joao, et al., ”Field report: Exploring fronts with multiple robots,” IEEE/OES Autonomous Underwater Vehicle Workshop, 2018.

[10] Pinto, Jose, et al,”Coordinated robotic exploration of dynamic open ocean phenomena,” Field Robotics, 2, 843-871, 2022.

[11] Stankiewicz P, Tan YT, Kobilarov M, ”Adaptive sampling with an autonomous underwater vehicle in static marine environments,” Journal of Field Robotics,38(4):572-97, 2021 Jun.

[12] Binney J, Krause A, Sukhatme GS, ”Informative path planning for an autonomous underwater vehicle,” IEEE International Conference on Robotics and Automation, pp. 4791-4796, 2010 May.

[13] Fiorelli E, Leonard NE, Bhatta P, Paley DA, Bachmayer R, Fratantoni DM,”Multi-AUV control and adaptive sampling in Monterey Bay,”IEEE journal of oceanic engineering,31(4):935-48, 2006 Oct.

[14] Monterey Bay Aquarium Research Institute (MBARI), “Autonomous Ocean Sampling Network II (AOSN II)” Collaborative Project, 2003.

[15] Yilmaz NK, Evangelinos C, Lermusiaux PF, Patrikalakis NM, ”Path planning of autonomous underwater vehicles for adaptive sampling using mixed integer linear programming,” IEEE Journal of Oceanic Engineering,33(4):522-37,2008 Oct.

[16] Toth P, Vigo D, ”Vehicle routing: problems, methods, and applications,” Society for industrial and applied mathematics; 2014 Nov.

[17] Vidal T, Crainic TG, Gendreau M, Prins C,”A hybrid genetic algorithm with adaptive diversity management for a large class of vehicle routing problems with time-windows,” Computers and operations research, 40(1):475–489, 2013.

[18] A.R. Robinson,”Physical Processes, Field estimation and Interdisciplinary Ocean Modeling,” Earth-Science Reviews, 40, 3-54, 1996.

[19] Vitorino J, Larangeiro S, Silva F, Pinto J, Almeida S, ”An operational system for the forecasting of oceanographic conditions in the Nazare Canyon area (W Portugal),” OCEANS'10 IEEE SYDNEY,pp. 1-5, 2010.

[20] Wouda, N.A., L. Lan, and W. Kool, ”PyVRP: a high-performance VRP solver package,” INFORMS Journal on Computing, pp. 943-955, 2024.

[21] Toth, P.; Vigo, D., eds.,”The Vehicle Routing Problem,” 2002.

[22] J. Pinto, P. S. Dias, R. Martins, J. Fortuna, E. Marques, and J. Sousa, “The LSTS toolchain for networked vehicle systems,” OCEANS 2013 MTS/IEEE Bergen, pp.1-9, 2013 June.

[23] R. Martins, P. S. Dias, E. R. Marques, J. Pinto, J. B. Sousa, and F. L. Pereira, “Imc: A communication protocol for networked vehicles and sensors,” Oceans 2009-Europe, pp. 1–6, IEEE, 2009.

[24] Hwang, Jimin, Neil Bose, and Shuangshuang Fan., ”AUV Adaptive Sampling Methods: A Review,” Applied Sciences 9, no. 15: 3145, 2019.

[25] A. F. Duarte, L. Bernacchi, R. Mendes, J. Sousa, L. Azevedo, ”Uncertainty maps as a tool for efficient AUV data collection,” submitted.

[26] A. Soares, “Direct sequential simulation and cosimulation,” Mathematical Geology, vol. 33, pp. 911–926, 2001.

[27] E.U. Copernicus Marine Service Information (CMEMS), “Atlantic-Iberian Biscay Irish Ocean Physics Analysis and Forecast,” Marine Data Store (MDS), Accessed: Oct. 14-25, 2024.