

# Adaptive Sampling of Algal Blooms Using Autonomous Underwater Vehicle and Satellite Imagery: Experimental Validation in the Baltic Sea

Joana Fonseca\*, Sriharsha Bhat, Matthew Lock, Ivan Stenius, Karl H. Johansson

**Abstract**—This paper investigates using satellite data to improve adaptive sampling missions, particularly for front tracking scenarios such as with algal blooms. Our proposed solution to find and track algal bloom fronts uses an Autonomous Underwater Vehicle (AUV) equipped with a sensor that measures the concentration of chlorophyll *a* and satellite data. The proposed method learns the kernel parameters for a Gaussian process (GP) model using satellite images of chlorophyll *a* from the previous days. Then, using the data collected by the AUV, it models chlorophyll *a* concentration online. We take the gradient of this model to obtain the direction of the algal bloom front and feed it to our control algorithm. The performance of this method is evaluated through realistic simulations for an algal bloom front in the Baltic sea, using the models of the AUV and the chlorophyll *a* sensor. We compare the performance of different estimation methods, from GP to curve interpolation using least squares. Sensitivity analysis is performed to evaluate the impact of sensor noise on the methods’ performance. We implement our method on an AUV and run experiments in the Stockholm archipelago in the summer of 2022.

**Index Terms**—Adaptive Control, Marine Robotics, Gaussian Processes, Satellite Data, Algal Blooms.

## I. INTRODUCTION

Researchers worldwide have been using satellites, remote sensing, and buoys to gather information on ocean phenomena and inform forecast models. These methods tend to be expensive, inefficient for spatial coverage, or unable to provide trustworthy data. They always have a human in the loop for decision-making or data post-processing. With this paper, we design and implement a solution and make the open-source software available, which allows for autonomous coastal surveys using autonomous underwater vehicles (AUVs), focusing on front tracking of ocean phenomena such as algal bloom fronts.

### A. Background and Motivation

Fronts are boundaries between water masses that differ significantly in the value of one or more variables, such as temperature, salinity, or substance concentrations. These fronts shape marine ecosystems as their presence indicates the occurrence of several physical and biological processes of interest,

\*Corresponding author.

This work is supported by Knut and Alice Wallenberg Foundation, Swedish Research Council, and Swedish Foundation for Strategic Research.

J. Fonseca, M. Lock, and K. H. Johansson are with the Division of Decision and Control Systems, School of Electrical Engineering and Computer Science, and Digital Futures. S. Bhat and I. Stenius are with the Department of Aeronautical and Vehicle Engineering, School of Engineering Sciences, KTH Royal Institute of Technology, SE-100 44 Stockholm, Sweden. {jfgf, svbhat, mwlock, stenius, kallej}@kth.se.

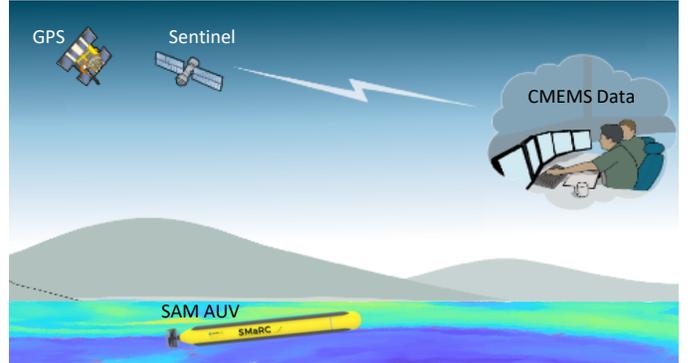


Fig. 1: System overview with the SAM AUV, the GPS, the Sentinel satellite, and the CMEMS data.

including transition zones, jets, eddies, and phytoplankton blooms [1], [2]. Among these frontal phenomena, harmful algal blooms (HABs) are the motivating scenario of this paper. According to [3], “HABs cause human illness, large-scale mortality of fish, shellfish, mammals, and birds, and deteriorating water quality”. HABs occur when algae colonies experience abnormal growth, which results in the production of harmful toxins [4]. These toxins can cause significant harm to marine ecosystems and pose a danger to human activities in the Baltic Sea, such as tourism and aquaculture. Accurate information about the location and movement patterns of algal blooms is crucial to monitor and mitigate these detrimental effects. Traditional methods for observation, such as satellite imaging or ship-towed sensors, are generally unable to provide measurements at the spatial and temporal resolutions required to understand dynamic ocean phenomena [5]. While remote sensing with satellites can offer a broad overview, such data is weather-dependent and prone to false positives in coastal areas. Thus, there is a significant scientific and societal interest in obtaining *in situ* measurements and developing systems for automated monitoring.

This paper proposes an approach to adaptive sampling of algal bloom fronts using an AUV informed by satellite imagery. The proposed algorithm enables tracking multiple water parameters, including chlorophyll *a* concentrations, turbidity, dissolved oxygen, and salinity. Out of these broad sampling possibilities, we are particularly interested in chlorophyll fronts due to their connection with the occurrence of harmful algal blooms (HABs) in the Baltic [6]. In Fig. 1, we illustrate this cyber-physical system, consisting of the AUV, the GPS signal that it uses for localization, the Sentinel satellite, which provides raw imagery of the region, and the Copernicus Marine Environment Monitoring Service (CMEMS), which

reanalyses the imagery from the Sentinel satellite into more accurate datasets that can then be used to inform our AUV.

### B. Related work

There has been significant work in developing solutions for autonomous ocean sampling. These solutions involve satellites, research vessels, floats, gliders, or AUVs. Satellite imagery offers broad datasets with varying resolutions but falls short for shallow waters. Despite the shortcomings of satellite imagery for coastal data, alternate solutions for near-shore regions have been the topic of extensive research for the past 50 years in efforts to bridge that gap [7]. For example, in [8], the authors propose a novel wavelet analysis with satellite data from repeating paths for coastal coverage. On the other hand, several solutions for *in situ* sampling, such as research vessels, generally use towed platforms integrating cameras, sensors, acoustic devices, and sidescan data [9]. In some cases, they also use underwater electronic holographic cameras for studying marine organisms such as plankton [10]. Focusing now on autonomous *in situ* sampling, one of the most widely used platforms is the profiling float, which moves with underwater currents and is sometimes referred to in the literature as a Lagrangian float. Profiling floats are more controllable in regions of more diverse currents, but controlling the horizontal motion of a profiling float remains challenging [11]. Despite their limited controllability, profiling floats such as in the Argo Program are perhaps the greatest international collaborative effort in the history of oceanography and provide researchers with open access to comprehensive data sets [12]. Underwater gliders are more controllable than floats; in [13], the authors who built the glider "Spray" define gliders as *autonomous profiling floats that use a buoyancy engine to move vertically and wings to glide horizontally*. The creators of the glider "Slocum" stated that their glider originated from the idea of adding horizontal propulsion to floats [14]. The "Seaglider" [15] is another example of one of the first gliders to be developed. With the development of gliders, the problem of under-actuated controllability has been a subject of analysis [16]. Gliders fall short in deterministic controllability, usually have a very limited sensor payload, and require a minimum operating depth of approximately 50m. These make them unsuitable for coastal applications in shallow water. AUVs, on the other hand, can carry different payload sensors, have better controllability, and can operate at multiple depths.

AUVs have been considered for solving the environmental sampling problem. One of the most common solutions in the literature is the open-loop scenario with a fixed sampling pattern. The most widely used sampling pattern is, undoubtedly, the lawn-mower [17], which has been used for both single-agent [18] and multi-agent scenarios [19]. However, other relevant methods, such as the spiral and circular patterns in [20], aim to improve efficiency and robustness. Or the oval spiral coverage strategy to plan coverage paths that better suit oval-shaped areas of interest [21]. While these open-loop strategies enable and even guarantee coverage of survey areas, they are inherently not designed to react or respond to changes in the observed features. In such cases, there is great motivation to close the loop.

Over the past two decades, there has been significant effort in closing the loop using adaptive sampling strategies. Adaptive sampling is a closed-loop control architecture in which an agent autonomously makes decisions during a mission in response to environmental changes. As reviewed in [22], adaptive sampling can be divided into three distinct objectives: source localization, front determination, tracking, and mapping. These objectives can be realized for different types of targets: thermoclines, algal blooms, oils spills, etc., using different vehicle configurations: single-agent, multi-agent with leader-follower, cooperative multi-agent, etc. In [23], the focus is on covering multiple thermoclines as they evolve in time and space in a dynamic water column. In [24] and [25], the problem of source localization using a multi-agent system is approached as a gradient climb with optimal formation to minimize the gradient estimate error. An example of front determination is [26], where the authors use a single AUV to find and track a salinity and temperature front while zig-zagging around it. Considering the mapping problem, in [27], the authors evaluate how to find the optimal path to maximize the accuracy of the field estimates for single and multi-agent scenarios. Also, in [28], the front determination problem is considered with a single AUV zig-zagging the front, but for bathymetric contours. In [29], a deep Chlorophyll maximum layer is tracked and mapped using three agents moving adaptively - one on the surface, one tracking the layer, and one mapping the area around the layer. There are other examples similar to our objective in this paper, in which an adaptive sampling algorithm is proposed that augments a standard Gaussian process (GP) with a nearest neighbor prior [30]. Unlike our approach, this paper does not use external data to aid the vehicle's decisions, while being similar to our method of building a GP model from measurements.

In this field, one of the rising problems considers integrating external data (satellite imagery, numerical models and predictions, etc.) with local measurements taken by autonomous agents such as AUVs. Specifically, the problem is how to aid ocean sampling missions using external data. An early example uses a predictive ocean model to assist in motion planning for steering an AUV to a high-valued location [31]. Here, it is assumed that there is a predicted model for the day of the mission, which is not available in this paper's scenario. Other related results include using knowledge from previous missions to create a model [32] and using hydrodynamic and biological model systems as prior information [33]. Such adaptive sampling is closely linked to data assimilation. Among the first works in data assimilation for oceanic applications are [34], [35]. In [36], the data collected by the AUV is augmented with remote sensing, buoys, a ship, particle imaging systems, and discrete water samples. In [37], the authors use a genetic algorithm to optimize the deployment, measurements, and information gain of a team of AUVs, ASVs, and mooring platforms. Such measurements are then included in an estimation framework to estimate and forecast environmental parameters given a dynamic ocean model. In [38], the authors propose using generic environmental models that are updated with data collected by a team of AUVs to update the map used by these vessels and, with it, perform mission-specific goals. And in

[39], a sparse and variational Gaussian process is trained with datasets of different seafloor textures and then used for seafloor texture classification. Our approach, in contrast, uses satellite data as prior in a Gaussian Process regression framework that is similarly updated with local measurements from AUVs to create high-resolution estimates of the environment around the autonomous agents.

### C. Contribution

The main contributions of this paper are an algorithm containing a control law and a GP estimator, a sensitivity analysis with estimation method comparison, and an experimental setup and experimental demonstration in the Baltic sea. In greater detail, this includes:

We implement a path-planning guidance law for adaptive sampling for the AUV heading. This control law receives information on the position of the AUV, the chlorophyll  $a$  concentration measurement it took, and the gradient of the chlorophyll  $a$  concentration gradient. We analyze the performance of the control law for a chlorophyll  $a$  concentration sensor uncertainty of  $10^{-3}\text{mg/m}^3$ , with full AUV and chlorophyll  $a$  concentration sensor model. In this scenario, the front tracking error concerning the chlorophyll  $a$  concentration front reference is always below  $0.02\text{mg/m}^3$ .

We propose a gradient estimator using GPs that estimates the gradient of chlorophyll  $a$  concentration. As it moves, the AUV records its position and chlorophyll  $a$  concentration. The estimation is updated at each timestep by fitting a GP model with the latest data points collected by the AUV. We analyze the performance of the GP gradient estimator for a chlorophyll  $a$  concentration sensor uncertainty of  $10^{-3}\text{mg/m}^3$ , with full AUV and chlorophyll  $a$  concentration sensor model. In this scenario, the gradient estimation error concerning the chlorophyll  $a$  concentration gradient field is always below  $0.4$  rad, corresponding to less than  $22$  degrees.

We design an experimental setup consisting of a cyber-physical system integrating the AUV software, the AUV hardware, the user interface, and a realistic simulator. The AUV software includes numerous packages that can be divided into the behavior tree, the algal bloom front tracking, the onboard controllers, and the dead-reckoning. The algal bloom front tracking library has been developed for the present work and includes the control law and two implemented estimation methods. It is available as an open-source contribution in <https://github.com/JoanaFonsec/gp4aes>. The algal bloom front tracking package was also developed for the present work and is responsible for interfacing with the other AUV packages, simulating the chlorophyll  $a$  sensor, and plotting. It is available as an open-source contribution in <https://github.com/JoanaFonsec/algalbloom-tracking>.

We run a sensitivity analysis in which we vary the standard deviation of the chlorophyll  $a$  sensor and measure the impact on the performance of the control law and the estimation method, considering the full AUV and chlorophyll  $a$  sensor model. The chlorophyll  $a$  sensor uncertainty we consider is between  $10^{-3}\text{mg/m}^3$  and  $10^{-1}\text{mg/m}^3$ . This is in line with the chlorophyll  $a$  concentration sensors in the market, which have

a resolution of  $10^{-2}\text{mg/m}^3$ . For this analysis, we consider two estimation methods, one based on Gaussian Process regression and the other based on Least Squares regression. The analysis confirms that the control law tracking error and the gradient estimation error increase as the standard deviation of the chlorophyll  $a$  sensor increases. Particularly, for a standard deviation equal and above  $10^{-2}\text{mg/m}^3$ , the gradient estimation error increases exponentially for the LSQ estimator and linearly for the GP estimator.

We provide experimental results from two surveys in the Stockholm archipelago in the Baltic Sea. In these experiments, we demonstrate that the proposed algorithm performs well in the real-time real-world scenario and compare them to a simulation under experiment conditions. Our results indicate that the front tracking error never exceeds 10 meters of overshoot and undershoot and that the gradient estimation error never exceeds 90 degrees. We also examine the sources of error, namely surface waves that influence the AUV's movement but also partially occlude the GPS receiver, which introduces Gaussian noise on the GPS-measured position of the AUV.

### D. Organization

The paper is organized as follows. In Section II, we introduce the proposed front tracking algorithm. This includes the high-level system architecture, the dataset we use, the GP model for the chlorophyll- $a$  concentration in the Baltic sea, and the path planning guidance law. An overview of the hardware and software components used in the AUV implementation is given in Section III. In Section IV, we provide realistic simulation results and a sensitivity analysis on the impact of sensor noise on algorithm performance with a comparison and evaluation of gradient estimation methods. Experimental results follow in Section V, from field trials in the Baltic sea in the summer of 2022. Concluding remarks, discussion, and future directions follow in Section VI.

## II. ALGAL BLOOM FRONT TRACKING

This paper considers algal bloom front tracking as an adaptive environmental sampling problem. The objective of front tracking is to find and track a front with limited global information on its location or shape and only use local information collected by the AUV as it moves to explore the map. This limited global information consists of satellite imagery from previous days to inform our model. Then, the AUV has to decide where to explore next, given the information it has collected so far. We approach this problem using an AUV with a chlorophyll  $a$  sensor and remote satellite data from CMEMS. Our solution consists of a novel system architecture containing three main components, as seen in the three green blocks of Fig. 2. They are a GP model estimator, a gradient estimator, and a motion controller. In the following subsections, we present the system architecture and its components.

### A. System Architecture

The system architecture is summarised in Fig. 2. Here, we illustrate the main components of the proposed system,

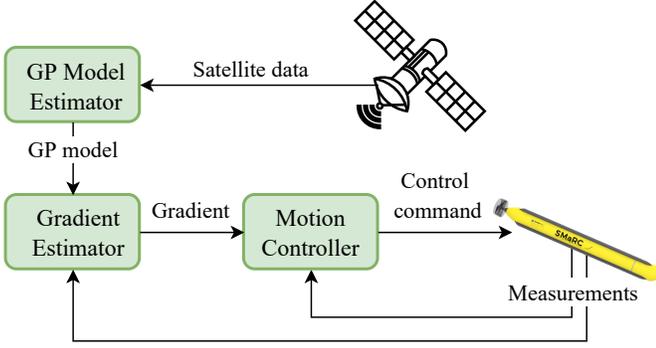


Fig. 2: System architecture with its main components: satellite data, GP model estimator, gradient estimator, motion controller, and AUV.

from the AUV to the motion controller, gradient estimator, GP model estimator, and satellite data.

The AUV has a chlorophyll  $a$  sensor that measures the chlorophyll  $a$  concentration at a set frequency as it moves in the field. The AUV movement is dictated by the control command received from the motion controller. The motion controller uses the AUV's past measurements and an estimation of the field gradient to calculate the control command, closing the adaptive sampling loop. The gradient estimator uses the past measurements taken by the AUV and a model of the chlorophyll  $a$  concentration field to estimate the chlorophyll  $a$  concentration field gradient. The GP model estimator uses the previous days of satellite data to train kernel parameters of a GP model that represents the field we want to estimate. Finally, the satellite data consists of remote measurements of the chlorophyll  $a$  concentration field from a few days preceding the mission and is used in the GP model estimator to generate the GP model estimate of the chlorophyll  $a$  concentration field.

### B. Satellite Data

The satellite data concerns chlorophyll  $a$  concentration for a given region. We denote this chlorophyll  $a$  concentration field by  $\delta(\mathbf{p})$ , where  $\delta$  denotes the chlorophyll  $a$  concentration at position  $\mathbf{p}$ . In this paper, we consider surface data only. Fig. 3 shows a plot of sample chlorophyll  $a$  concentration data, where high regions of high concentration are highlighted in yellow and regions of low concentration are highlighted in blue. The dark grey area represents the land. This data has a spatial resolution of 2 km by 2 km and is obtained from CMEMS [40]. The location is on the west coast of Finland, near the coastal city *Pori*. We chose this location because a clear chlorophyll  $a$  bloom front can be observed here. We hypothesize this is due to the nutrients that the river *Kokemäenjoki* carries into the Baltic sea [41]. This paper will focus on the region marked by the red square taken on the 17th of April 2021. We chose April because it is the spring season of algal blooms.

### C. GP Model Estimator

The GP model estimator models the chlorophyll  $a$  concentration field  $\delta(\mathbf{p})$  for the given region and time, exploiting prior information from satellite data of the previous days and measurements taken by the AUV in real-time. Given this

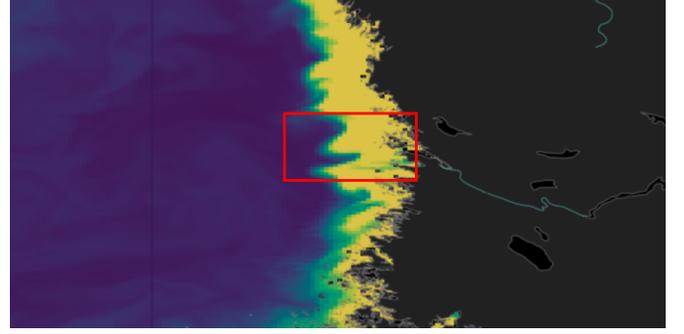


Fig. 3: CMEMS data of chlorophyll  $a$  in the Baltic Sea (blue-yellow) and land (dark grey).

application's slow time scale, we assume that the chlorophyll  $a$  concentration fields at different days have identical distributions, which simplifies model fitting and reduces the computational complexity.

To obtain the chlorophyll  $a$  concentration model, we must define the type of kernel that will accurately depict the process. The kernel represents *a priori* knowledge about the process by specifying how the chlorophyll  $a$  concentration data is related to the corresponding spatial locations. Among the multitude of kernels described in the literature, only some fit the characteristics of the biogeochemical data we consider. We use the Matérn kernel [42], which proves to be capable of modeling different degrees of smoothness across both vertical and horizontal length scales [43]. The covariance matrix  $K \in \mathbb{R}^{N \times N}$  is defined concerning two points in the field map,  $x_i$  and  $x_j$ . Each matrix element is computed by the kernel function  $k(x_i, x_j)$  for which  $1 \leq i, j \leq N$ . The kernel is defined as

$$K_{i,j} = k(x_i, x_j) = \sigma_k^2 (1 + r) e^{-r}, \quad (1)$$

where  $r^2 = (x_i - x_j)^\top M (x_i - x_j)$ , with

$$M = \begin{bmatrix} \left(\frac{\sqrt{3}}{l_0}\right)^2 & 0 \\ 0 & \left(\frac{\sqrt{3}}{l_1}\right)^2 \end{bmatrix}. \quad (2)$$

The kernel hyper-parameters are  $(\sigma_k^2, l_0, l_1)$ , where  $\sigma_k^2$  is the variance of the chlorophyll  $a$  concentration process, and  $(l_0, l_1)$  define the length scale in each dimension.

The kernel hyper-parameters,  $(\sigma_k^2, l_0, l_1)$ , are estimated by maximizing the log marginal likelihood of the prior distribution - using only the available satellite data from previous days. This data is called the training set and consists of a vector of size  $N$  containing positions in the chlorophyll  $a$  concentration field  $\mathbf{X} = [\mathbf{p}_1, \dots, \mathbf{p}_N]$ , and their respective chlorophyll  $a$  concentration values  $\mathbf{y} = [\delta_1, \dots, \delta_N]$ . The log marginal likelihood to maximize is

$$\log p(\mathbf{y}|\mathbf{X}) = -\frac{1}{2} \mathbf{y}^\top (K + \sigma^2 I)^{-1} \mathbf{y} - \frac{1}{2} \log |K + \sigma^2 I| - \frac{N}{2} \log 2\pi \quad (3)$$

where  $K$  is the  $N \times N$  covariance matrix in which each value is created as in (1), and  $\sigma$  is the noise variance of each data point.

Using the trained kernel, the GP model can be fitted with the collected data, using the standard conditioning formulae [42] to obtain the model for the chlorophyll  $a$  concentration field, which we define as  $\bar{\delta}(\mathbf{p})$ . To do so, we consider the  $n$  most recent measurements taken by the AUV. It contains the AUV's positions  $\mathbf{P} = [\mathbf{p}_1, \dots, \mathbf{p}_n]$ , and its chlorophyll  $a$  concentration measurements  $\Delta = [\delta_1, \dots, \delta_n]$ . Then, the mean of the model  $\delta(\mathbf{p})$  is denoted by  $\bar{\delta}(\mathbf{p})$  and the covariance by  $\text{cov}(\delta(\mathbf{p}))$ . We can compute the mean and covariance at some point  $\mathbf{p}_*$  as

$$\bar{\delta}(\mathbf{p}_*) = K_* (K + \sigma^2 I)^{-1} \Delta \quad (4)$$

$$\text{cov}(\delta(\mathbf{p}_*)) = K_{**} - K_* [K + \sigma^2 I]^{-1} K_*^T \quad (5)$$

where  $K \in \mathbb{R}^{n \times n}$  corresponds to the covariance between the data in points  $\mathbf{P}$ ,  $K_* \in \mathbb{R}^{1 \times n}$  corresponds to the covariance between the data in points  $\mathbf{p}_*$  and  $\mathbf{P}$ ,  $K_{**} = \sigma^2$  corresponds to the variance at the point  $\mathbf{p}_*$ , and  $\sigma^2$  is the variance of the measurement noise. Since the set of the  $n$  most recent measurements is changing in time, then our estimate  $\bar{\delta}(\mathbf{p}_*)$  is also changing in time.

To evaluate the accuracy of the GP model applied to chlorophyll  $a$  concentration fields, the kernel is trained using both the low and high-resolution datasets. Then, the goodness of fit of both sets of parameters is evaluated by comparing the respective predictions of the chlorophyll  $a$  concentration to the ground truth. Using past satellite data, we construct the set  $X$  and  $y$  in (3) from data from multiple process realizations. Such division of the training dataset prevents overfitting. Then,  $X$  and  $y$  are composed of data from 3 days before the prediction date, selecting non-overlapping random scattered sub-datasets of the same size each day. The optimization algorithm for the maximization of (3) is L-BFGS-B, which is implemented in the `scipy` library, and the resulting parameters follow in Table I.

	$\sigma^2$	$l_0$	$l_1$
Low res	44.2959	0.5465	0.2890
High res	18.2106	0.0559	0.0245

TABLE I: Kernel hyperparameters obtained through maximum likelihood estimation, using the low-resolution and high-resolution datasets.

The results of this comparison are presented in Figures 4a and 4b. The observations and test datasets are approximately of size 1500 and 13500, respectively, having in consideration a ratio of 10/90% between both, where the former is a set of scattered samples from the ground truth data in figure 4c, having a standard deviation of  $\sigma_n = 10^{-3}$ .

Visually, the results are very similar. The average relative error of the prediction compared to the ground truth data in figure 4c was approximately 12% and 11% using low and high-resolution data, respectively. Based on the similarity between results, we conclude that the proposed GP model accurately represents the statistical properties of the chlorophyll  $a$  concentration in the operations scenario, even when the training dataset is different from the ground truth data.

#### D. Gradient Estimator

The gradient estimator derives the previously obtained model of the chlorophyll  $a$  concentration to estimate the

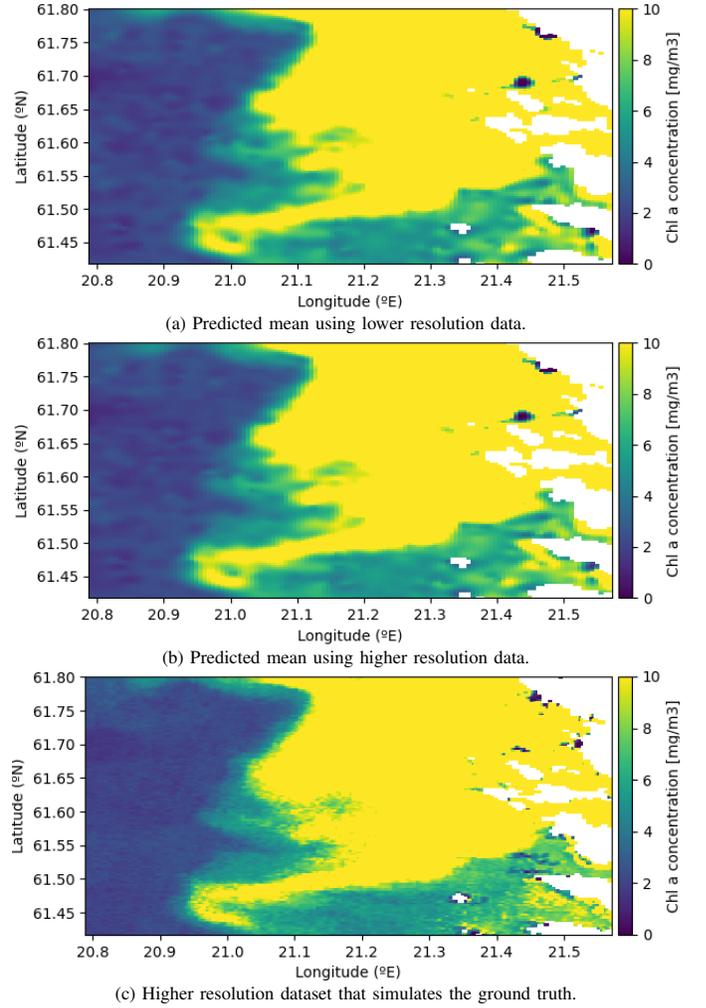


Fig. 4: Predicted mean from scattered measurements compared to the ground truth (high-resolution data).

chlorophyll  $a$  concentration gradient field. From the equation (4), the gradient  $\nabla \bar{\delta}(\mathbf{p}_*)$  is obtained by computing the derivative of the predicted chlorophyll  $a$  concentration with respect to position  $\mathbf{p}_*$ ,

$$\nabla \bar{\delta}(\mathbf{p}_*) = \nabla_{\mathbf{p}_*} \left[ K_* (K + \sigma^2 I)^{-1} \Delta \right]. \quad (6)$$

Since the second and third terms inside the gradient in (6) are constant relative to  $\mathbf{p}_*$ , we only need to compute  $\nabla_{\mathbf{p}_*} K_*$ . Each element of the  $K_*$  matrix is given by (1), in which  $x_i$  corresponds to  $\mathbf{p}_*$  and  $x_j$  corresponds to  $\mathbf{p}_j \in \mathbf{P}$ . So we take the derivative of  $k(\mathbf{p}_*, \mathbf{p}_j)$  with respect to  $\mathbf{p}_*$ ,

$$\nabla_{\mathbf{p}_*} k(\mathbf{p}_*, \mathbf{p}_i) = -\sigma_k^2 e^{-r} M(\mathbf{p}_* - \mathbf{p}_j),$$

where  $M$  and  $r$  are as in subsection II-C. Note that the gradient of the kernel equation is not defined when the test point in  $\mathbf{P}$  is equal to the current position  $\mathbf{p}_*$ . To account for this, the current position  $\mathbf{p}_*$  is not included in  $\mathbf{P}$  when computing (6). Then the gradient estimate at position  $\mathbf{p}_*$  is

$$\nabla \bar{\delta}(\mathbf{p}_*) = \nabla_{\mathbf{p}_*} K_* (K + \sigma^2 I)^{-1} \Delta. \quad (7)$$

#### E. AUV

The AUV receives the control command  $\mathbf{u}$  from the motion controller, which is the reference for direction and velocity.

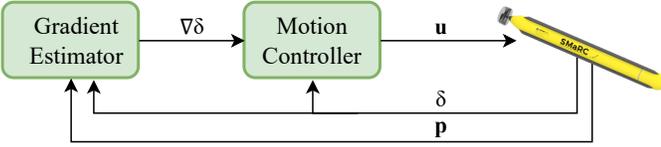


Fig. 5: Control architecture with the motion controller, the gradient estimator, and the AUV.

Then, using its internal lower-level controller, the AUV turns this reference  $\mathbf{u}$  into thrust commands to its thrusters  $\tau_C$ . For this AUV, we consider the 6DOF model in which the state is the velocity vector given by  $\boldsymbol{\nu} = [u \ v \ w \ p \ q \ r]^T$  containing the translational and rotational velocities. Note that these velocity vector elements directly influence the AUV's position  $\mathbf{p}$ .

The dynamics of the AUV can be formulated as a nonlinear system represented by a vectorial notation presented by Fossen [44] as follows:

$$(\mathbf{M}_{RB} + \mathbf{M}_A)\dot{\boldsymbol{\nu}} + (\mathbf{C}_{RB}(\boldsymbol{\nu}) + \mathbf{C}_A(\boldsymbol{\nu}))\boldsymbol{\nu} + \mathbf{D}(\boldsymbol{\nu})\boldsymbol{\nu} + \mathbf{g}(\boldsymbol{\eta}) = \boldsymbol{\tau}_C, \quad (8)$$

where  $\mathbf{M}_{RB}$  is the rigid body mass and inertia matrix and  $\mathbf{C}_{RB}$  is the matrix of Coriolis and centripetal terms on the left-hand side.  $\mathbf{M}_A$  and  $\mathbf{C}_A(\boldsymbol{\nu})$  represent the effect of added mass,  $\mathbf{D}(\boldsymbol{\nu})$  represents the damping matrix, and  $\mathbf{g}(\boldsymbol{\eta})$  is the vector of gravitational and buoyancy forces and moments.  $\boldsymbol{\tau}_C$  is a vector of external control forces based on the AUV's actuator configuration. The damping matrix  $\mathbf{D}(\boldsymbol{\nu})$ , in particular, has a significant effect on the nonlinear hydrodynamics of the AUV [45] and is a key simulation parameter.

#### F. Motion Controller

The control law we propose is summarised in Fig. 5. It relies on the chlorophyll  $a$  gradient  $\nabla\delta$  and the latest chlorophyll  $a$  concentration measurement  $\delta$  to produce a control command  $\mathbf{u}$ . First, we define a front as a level set of a time-varying scalar field  $\delta: \mathbb{R} \times \mathbb{R}^2 \rightarrow \mathbb{R}$ :

$$F(t) = \{\mathbf{p} \in \mathbb{R}^2 : \delta(t, \mathbf{p}) = \delta_{\text{ref}}\}, \quad (9)$$

where  $\delta_{\text{ref}}$  is some reference value,  $\mathbf{p}$  the position and  $t$  time.

Assuming that the reference value  $\delta_{\text{ref}}$  is known, we used the previously developed control law as in [46]. There, we define the control law as

$$\begin{aligned} \mathbf{u}(t, \mathbf{p}) &= \mathbf{u}_{\text{seek}}(t, \mathbf{p}) + \mathbf{u}_{\text{follow}}(t, \mathbf{p}) \\ \mathbf{u}_{\text{seek}}(t, \mathbf{p}) &= -\alpha_{\text{seek}}(\delta(t, \mathbf{p}) - \delta_{\text{ref}})\nabla\delta(t, \mathbf{p}) \\ \mathbf{u}_{\text{follow}}(t, \mathbf{p}) &= \alpha_{\text{follow}}R_{\pi/2}\nabla\delta(t, \mathbf{p}), \end{aligned} \quad (10)$$

where  $\nabla\delta$  is the gradient of  $\delta$  with respect to  $\mathbf{p}$ ,  $R_{\pi/2}$  is a mapping which rotates vectors by 90 degrees, and  $\alpha_{\text{seek}}$  and  $\alpha_{\text{follow}}$  are tunable parameters.

As seen in Fig. 6, the control law consists of two components:  $\mathbf{u}_{\text{seek}}$ , which controls the AUV towards the front by following the gradient field, and  $\mathbf{u}_{\text{follow}}$  which controls the AUV to move along the front, perpendicularly to the gradient field. By designing the control law with these two components, we ensure convergence to the front [47]. Namely, note that if

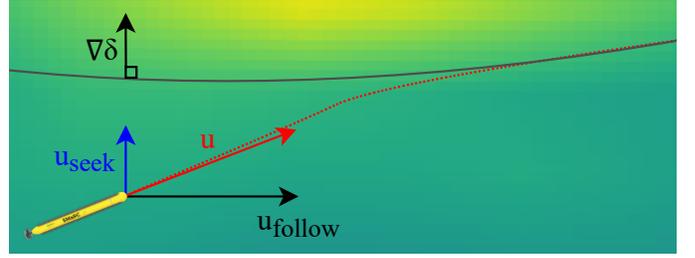


Fig. 6: Seek and follow components of the control law and gradient.

$\delta(t, \mathbf{p}) \neq \delta_{\text{ref}}$ , the  $\mathbf{u}_{\text{seek}}$  component grows proportionally to this difference, making seeking the front a priority, comparing to following the front. On the other hand, if the AUV is on the front, the most prominent component becomes the  $\mathbf{u}_{\text{follow}}$ .

### III. EXPERIMENTAL SETUP

The test platform used in these experiments is the SAM AUV, a research platform developed at the Swedish Maritime Robotics Centre (SMaRC) <sup>1</sup>. The algal bloom front tracking algorithm from the previous section is integrated into the SAM AUV software to be deployed in the field and validated experimentally.

The experimental setup is a cyber-physical system (CPS) whose architecture is summarised in Fig. 7. The user interacts with the AUV software system through an interface that enables them to send mission plans and monitor the current status. A behavior tree monitors the mission status and delegates actions to an algal bloom front tracker. The algal bloom front tracker reads payload measurements of chlorophyll  $a$  data and sends waypoints to onboard feedback controllers. These onboard controllers, in turn, send commands to actuators considering state feedback. Such state feedback is provided by dead-reckoning based on onboard navigation sensors on the AUV hardware. The entire software system can also be validated in simulation via the Stonefish simulator [48]. Each component of the CPS will be further described in the following subsections.

#### A. AUV Hardware

We begin by describing the SAM AUV hardware in the upper right corner of Fig. 7. SAM, short for Small and Affordable Maritime robot (as seen in Fig. 8), is a torpedo-shaped, under-actuated AUV that serves as an agile research platform for testing new sensing, perception, and control strategies [49], [50].

Five key actuator subsystems enable SAM to be highly maneuverable and hydrobatic (these are depicted in Fig. 9):

- 1) The *Longitudinal Center of Gravity (LCG)* system uses a movable battery pack to change the center of gravity position longitudinally and enable static pitch control.
- 2) The *Transversal Center of Gravity (TCG)* system contains rotating counterweights that enable static roll control or changes to the AUV's stability margin.
- 3) The *Variable Buoyancy System (VBS)* facilitates buoyancy regulation and static depth control by pumping water in and out of a tank.

<sup>1</sup><https://www.smarc.se>

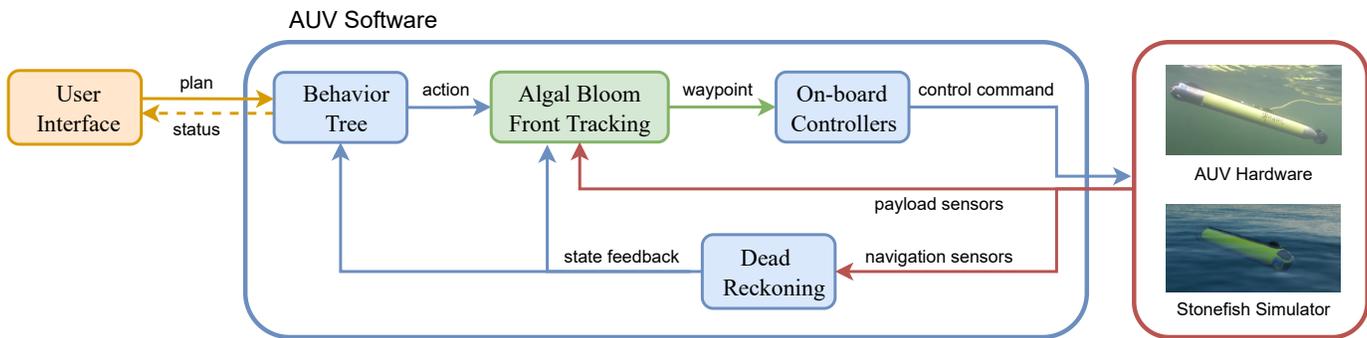


Fig. 7: The cyber-physical system architecture integrates the user interface, AUV software, AUV hardware, and simulator.



Fig. 8: The SAM AUV developed by SMaRC.

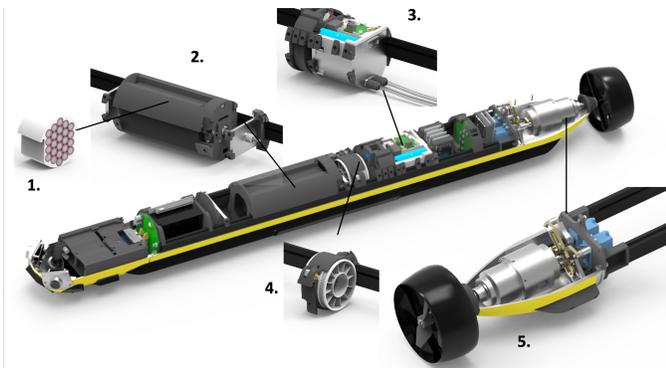


Fig. 9: AUV subsystems: 1. Battery pack, 2. Longitudinal Center of Gravity Trim System, 3. Variable Buoyancy System, 4. Transversal Center of Gravity System, 5. Thrust Vectoring System with Counter-rotating Propellers.

- 4) *Counter-rotating propellers* provide propulsion while compensating for propeller-induced roll.
- 5) The *Thrust Vectoring* system contains a servo-actuated nozzle for steering in the horizontal and vertical planes. The thrust vectoring nozzle enables rapid maneuvering in the horizontal and vertical planes.

Sensors have been integrated into SAM for navigation and environmental sensing. Navigation sensors include an Inertia Measurement Unit, a compass, a GPS receiver, a Doppler Velocity Logger (DVL) for bottom tracking, and pressure sensors for depth measurements. Payload sensors include cameras and sidescan sonar for inspection and surveying, a forward-looking sonar for obstacle avoidance, and a Conductivity-Temperature-Depth (CTD) probe for water-column monitoring. For our algal bloom front tracking application, we use a chlorophyll  $a$ -turbidity-phycochlorophyll fluorometer for phytoplankton sensing.

### B. Stonefish Simulator

Before running autonomy software on the SAM AUV, simulations of mission scenarios can be performed within the

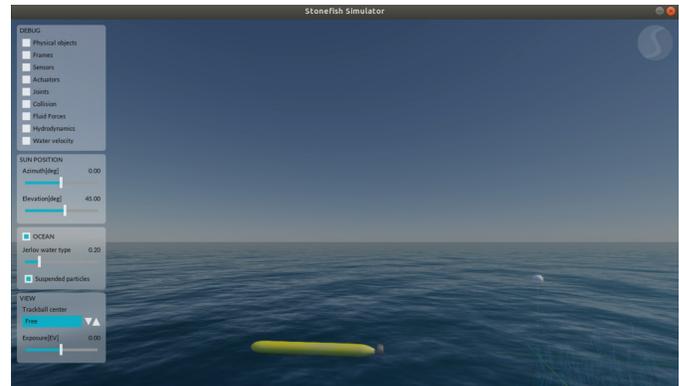


Fig. 10: Print screen of the Stonefish simulator.

Stonefish simulator, represented in the bottom right corner of Fig. 7. Stonefish offers a photo-realistic simulation environment where entire missions can be modeled and rehearsed. The AUV's dynamics are modeled within the simulator together with models of the sensors. Objects, environmental features, and bathymetry can be imported into the simulator to create mockups of planned environments. Within Stonefish, perception and planning software can be validated before deployment on the hardware.

For the application presented in this paper, the satellite data for algal blooms is modeled in Stonefish as a lookup table of chlorophyll  $a$  values over a grid encompassing the entire simulation environment. A simulated chlorophyll sampler interpolates the relevant chlorophyll measurement from this grid based on the AUV's current position. The software interfaces within the Stonefish simulator and the real SAM AUV are identical, thus enabling virtual validation of a full mission sequence for algal bloom tracking.

### C. AUV Software

The four AUV software components presented in the center of Fig. 7 are further detailed in this section. The autonomy software of SAM runs on the Robot Operating System (ROS)<sup>2</sup> environment. Its sub-components include a *behavior tree* for decision-making and mission execution, a path planner for *algal bloom front tracking*, *onboard controllers* for path following, and a *dead-reckoning* package for navigation. These software packages are also available in an open-source repository in <https://github.com/smarc-project>.

<sup>2</sup><https://www.ros.org>

1) *Behavior Tree*: SAM uses a Behavior Tree (BT) to ensure safe and transparent mission execution. A BT is a reactive decision-making structure that is comprised of Sequences, Fallbacks, Actions, and Conditions. The main objective of the BT is to receive a mission plan and delegate actions (e.g., waypoints) to lower-level systems (e.g., motion planners and controllers) while checking the safety criteria of the AUV simultaneously and continuously. In an unsafe situation, the BT executes emergency actions to bring the system back to a safe state. A BT ensures safety and compliance requirements during mission execution by disallowing unsafe behaviors autonomously. Further information on designing BTs for underwater robots can be found in [51], [52].

The Gaussian process path planner presented in Section II-F is integrated into the BT with additional conditions and actions considering satellite data ingestion, chlorophyll *a* sampling, and front-tracking. Each algal bloom tracking experiment using SAM follows a set operational sequence. Considering a relevant area with algal blooms, the following workflow is used for sampling and tracking the bloom:

- 1) An area of interest is specified, and satellite imagery is downloaded.
- 2) An initial mission plan is set via a user interface for the AUV to traverse to the algal bloom feature.
- 3) The AUV is launched, and a GPS fix is acquired.
- 4) The AUV starts the mission and follows user-defined waypoints to reach the algal bloom front's vicinity.
- 5) The AUV detects the front through a chlorophyll sensor, and the front-tracking algorithm is engaged.
- 6) The path planner generates new waypoints to track the front upon sampling, and the AUV follows these waypoints to track the front.

This sequence is also rehearsed within the Stonefish simulator. To follow such a mission sequence, the BT for algal bloom tracking is summarised in Fig. 11. A sanity check on chlorophyll measurements is performed in the first subtree. Second, safety conditions are checked. If either fails, the mission is aborted, and emergency actions are performed. Third, if the system is safe and measurements are available, the user-defined waypoint mission is followed. Fourth, if the AUV reaches the front, the algal bloom front following action is performed. This is further detailed below.

Such a BT structure allows the designer to prioritize actions, sequence missions, and ensure the vehicle operates safely while keeping the overall setup modular and easy to understand.

2) *Algal Bloom Front Tracking*: The algorithm of the previous section is incorporated as the algal bloom front following action in the BT. The action ingests payload data on chlorophyll concentration and sends live waypoints to the onboard controllers so that the AUV follows the front.

When the AUV crosses the algae front, the front tracking behavior is enabled, with a higher priority than following the original waypoints (see Fig. 11). A path planner for front tracking sends new waypoints to the AUV based on live measurements. The AUV samples the front and follows the

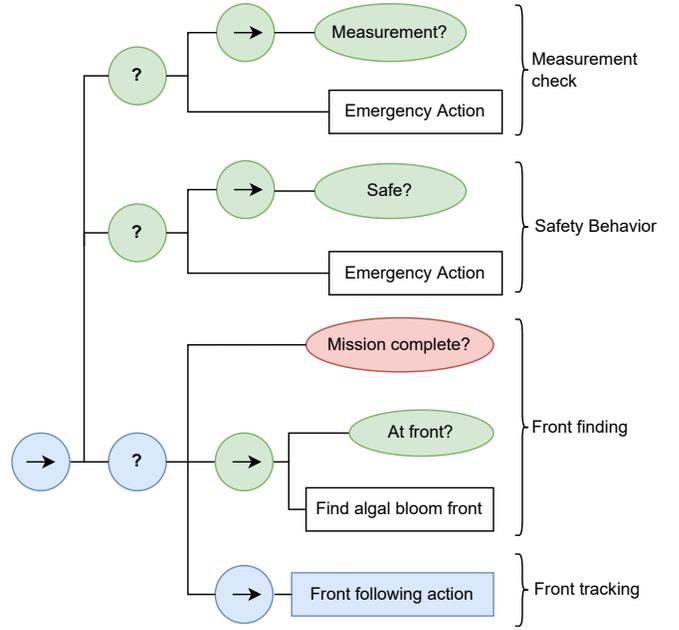


Fig. 11: High-level view of the BT used for algal bloom front tracking. Inner nodes are sequences (arrows) and fallbacks (question marks). Leaf nodes are actions (rectangle) and conditions (ellipse). All nodes can return Success (green), Failure (red), and Running (blue).

edge of the bloom. Once the AUV has exited the front, the vehicle will fall back to following the operator's plan.

3) *Onboard Controllers*: The waypoints sent from the original mission plan and the front-tracking algorithm provide input to the onboard controllers on SAM. These onboard controllers enable the AUV to follow pre-defined waypoints and track the algal front. Given a set of waypoints, a line-of-sight guidance law minimizes cross-track error and ensures the vehicle can approach each waypoint at a set heading and depth. Feedback controllers then control the AUV to follow these set points. Further information on the waypoint following guidance law can be found in [53].

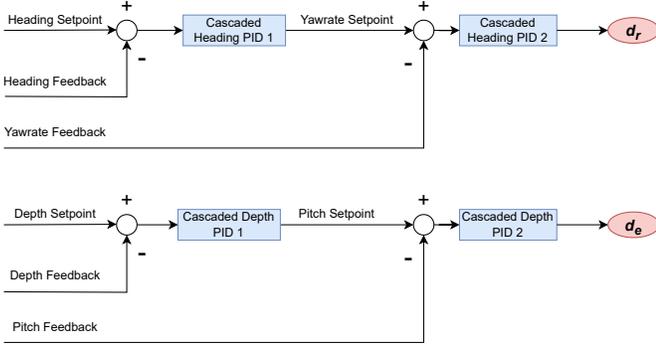
In SAM's case, the control force vector  $\tau_C(c)$  in equation (8) is a function of the available actuator inputs, which are contained in

$$c = [ rpm_1 \quad rpm_2 \quad d_e \quad d_r \quad LCG \quad VBS ], \quad (11)$$

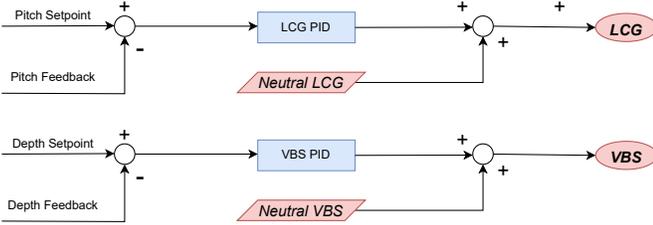
where  $rpm_{1,2}$  represent the propeller speed,  $d_e$  and  $d_r$  are vertical and horizontal thrust vector angles, and  $LCG$  and  $VBS$  specify the position and buoyancy level, respectively.

The feedback control architecture is presented in Fig. 12). First, we consider flight control to regulate the heading and depth where cascaded Proportional-Integral-Derivative (PID) controllers are used to command the thrust vector angles. In the outer loop, the controllers provide a yaw rate and pitch setpoint, which are translated to actuator commands to the thrust vectoring system in the inner loop. These account for couplings between states for flight control. Second, to stabilize the AUV in pitch and depth, additional PID controllers control the trim (LCG) and buoyancy (VBS) subsystems. Finally, coupled roll and velocity control are realized using parallel PIDs to command the counter-rotating propellers. These provide an average propeller rpm to achieve the desired velocity while

### Heading and Depth Control



### Pitch and Buoyancy Control



### Velocity and Roll Control

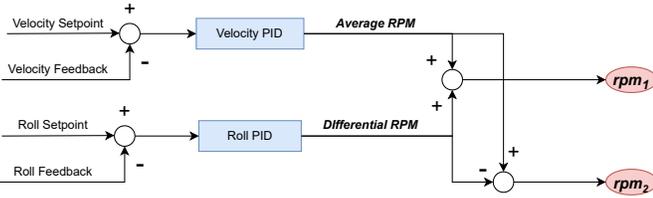


Fig. 12: The feedback control architecture on SAM with cascaded heading and depth PIDs for flight control (top), trim stabilization with pitch and buoyancy control (middle), and coupled velocity and roll control with the counter-rotating propellers (bottom).

also providing a differential rpm between the two propellers that causes the AUV to hold a roll angle. It is also possible to directly command constant rpm values instead of a desired velocity. The combination of flight and trim controllers enables SAM to track the algal front at a specified velocity or propeller rpm but are dependent on reliable state feedback.

4) *Dead Reckoning*: Underwater navigation is challenging because radio waves attenuate rapidly in water. This means that GPS-based positioning and navigation are not available underwater — we need to use inertial and acoustic sensors for estimating the AUV’s position, orientation, and velocities. Dead reckoning is thus used for obtaining state feedback. An Extended Kalman Filter is used to fuse acoustic and inertial measurements collected by the AUV’s onboard sensors to estimate the vehicle’s current position, orientation, and velocity. In particular, the IMU and compass are used to obtain orientations, angular velocities, and accelerations, the DVL is used to obtain linear velocities, and the pressure sensor is used to calculate the depth. These are then fused and integrated with a motion model to obtain a position and velocity estimate.

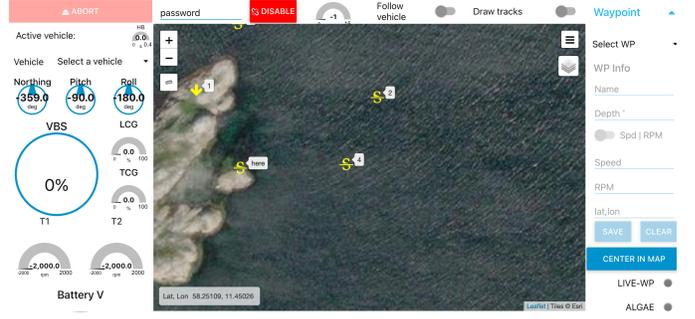


Fig. 13: Print screen of the user interface: Node-Red.

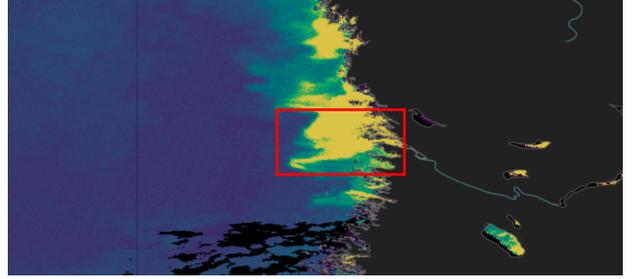


Fig. 14: CMEMS data of chlorophyll *a* concentration in the Baltic Sea (blue-yellow), clouds and cloud coverage (black), and land (dark grey).

### D. User Interface

The user interacts with SAM’s software system via the user interface in the left corner of Fig. 7. As represented in Fig. 13, it consists of a web-based graphical interface based on *Node-RED*. It enables the operator to plan the mission on a world map and monitor the vehicle’s current status during the mission. Different AUV parameters can be tracked, new missions can be run, and the current status of several measurements can be observed. A second graphical user interface allows test engineers to run specific hardware drivers and controllers on SAM for low-level control and validation.

## IV. SIMULATION RESULTS

In this section, the control and estimation components of the proposed system architecture are tested in a front-tracking procedure in the operational area in Fig. 4c. The simulation starts by deploying the vehicle close to the front and providing an initial heading setpoint towards it. When the AUV reaches the front, the gradient estimation is triggered, and the control law receives the estimated value from the GP model as an input. The section is divided into three subsections: simulation setup, numerical results, method analysis, and comparison.

### A. Simulation setup

In this subsection, we introduce the setup for the simulations presented in the following subsections. For this scenario, we consider the environment illustrated in Fig. 14 in which we will deploy the AUV and track the algal bloom front. Note that this environment we consider using this data is for simulation purposes only and differs from the satellite data used to inform the GP model, as described in subsection II-B. Here, the chlorophyll *a* concentration is represented by a map

that goes gradually from a high concentration in yellow to a low concentration in blue; these are the values that will later be measured by the chlorophyll  $a$  concentration sensor mounted on the AUV. Specifically, the simulated mission will occur inside the red square. The data used to simulate this environment has a spatial resolution of 300 m by 300 m [54] from the exact satellite data location considered earlier. This environment is modeled within the Stonefish simulation environment presented in the previous section. Further, the entire chlorophyll  $a$  concentration map from the satellite image is integrated as a lookup table to enable simulated sampling.

The source code that implements the algorithm is available as an open-source contribution on two repositories. The first one is the Gaussian Processes for Adaptive Environmental Sampling (GP4AES) library, which includes the GP model estimator, the gradient estimator, and the motion controller <https://github.com/JoanaFonseca/gp4aes>. The second one is the ROS [55] interface, which uses the GP4AES library and handles the connection with the AUV's software <https://github.com/JoanaFonseca/algabloom-tracking>.

The simulation starts in the Stonefish Simulator by deploying the AUV close to the front and providing an initial estimated gradient. When the AUV is near the front, the gradient estimation is triggered. The AUV travels at a constant speed of  $v = 1$  m/s. For that reason, we are interested only in the ratio of  $\alpha_{\text{seek}}$  and  $\alpha_{\text{follow}}$ , and thus the latter is set to 1. Moreover, based on the available satellite data, we consider  $\delta_{\text{ref}} = 7.45$  mg/m<sup>3</sup>. While tracking the front, the AUV collects measurements at a frequency  $f = 1$  Hz while considering a standard deviation of the measurement noise of  $\sigma = 10^{-3}$  mg/m<sup>3</sup>. The measurements are filtered using a weighted moving average filter of size 3, with  $w = [0.2, 0.3, 0.5]$

$$\delta_{\text{filtered}}(t) = w_{-2}\delta(t-2) + w_{-1}\delta(t-1) + w_0\delta(t). \quad (12)$$

With the same sampling rate, the gradient is estimated as in (6), using data from the last  $n = 200$  measurements. Then we apply a first-order low pass filter, with  $\alpha = 0.97$ ,

$$\nabla\delta_{\text{filtered}}(t) = \alpha\nabla\delta(t-1) + (1-\alpha)\nabla\delta(t). \quad (13)$$

The parameters described are summarised in Table II.

$\sigma$	$\alpha_{\text{seek}}$	$v$	$n$	$\delta_{\text{ref}}$	$\alpha$
$10^{-3}$ mg/m <sup>3</sup>	10	1 m/s	200	7.45 mg/m <sup>3</sup>	0.97

TABLE II: Control algorithm parameters.

## B. Numerical results

In this subsection, we present and analyze the results from simulated missions using two gradient estimation methods.

We illustrate the complete AUV mission in Fig. 15. The AUV follows the front while collecting chlorophyll  $a$  concentration measurements, estimating the chlorophyll  $a$  concentration field and its field gradient, and updating its direction. The complete mission has a duration of approximately 23 hours. The starting position is far from the bloom and represented by the white star, while the final position is on the front and represented by the white square. In this figure, the AUV

closely follows the algal bloom front. This is further analyzed in subsection IV-B, in which we focus and zoom-in in on the area inside of the blue square.

In Fig. 16, we zoom in on a region of the longer mission, previously defined by a blue square, to focus on the performance of the gradient estimation and front tracking algorithm. This region corresponds to about 5 hours of mission time. Here we illustrate the gradient performance through arrows representing the true and estimated gradients along the path. The true gradient refers to the gradient that the AUV would be able to compute if it had access to the global information of the field. We compute it by taking the spatial derivative of the chlorophyll  $a$  concentration field. The estimated gradient refers to the output of the gradient estimator, as in (7). The angle between the true and estimated gradient arrows indicates the gradient error. However, in this scenario, the chlorophyll  $a$  concentration field is non-convex and fast-changing even in small areas. Therefore, the gradient is an abstraction that gives an idea of direction rather than an exact measure of the gradient. If we analyze the straight portions of the path, we would say that the error is very close to zero. In contrast, we could say that the error is larger by analyzing the portions of the path with higher curvature. At the same time, the gradient looks ambiguous and sensitive to small changes in position. As for both the control performance and comparison between gradient estimators, this figure doesn't allow for such analysis so we zoom-in in on the two areas inside the blue squares.

Let us now analyze the performance of the control and its control components in Fig. 17. These figures correspond to the two zoom-in locations in the previous figure; the front is the thin black line, and a thicker red line represents the AUV path. We also plot the *seek* and *follow* components of the control law using arrows along the AUV path. The control law is constructed as in eq. 10, and it is a sum of the seek component, which has the same direction as the estimated gradient, and the follow component, which has a perpendicular direction with respect to the estimated gradient. This sum constitutes the control law corresponding to the AUV's direction of movement. The first zoom-in corresponds to about 15 minutes of mission time. Here, for both estimators in Fig. 17a and Fig. 17b, the AUV always follows the front closely and with minimal error and without visible differences of performance among the estimators. This is expected as the front is smooth in curvature, and the AUV remains on top of the front. The control seek component accounts for small adjustments in the trajectory. The second zoom-in corresponds to about 30 minutes of mission time. In Fig. 17c and Fig. 17d, the AUV remains on top of the front most of the time; thus, the control follow component dominates the control law. On the other hand, once the curvature changes faster, the control follow component is reduced, and the control seek component becomes the dominating component. In this scenario of fast-changing curvature, the AUV seems to have a delay in updating its direction. Two leading causes for this behavior are the AUV's turning radius and the update function with the update rate on the gradient. The gradient's update function in (13) introduces a delay and a cut-off frequency. This cut-off frequency is inversely proportional to the update rate. Hence,

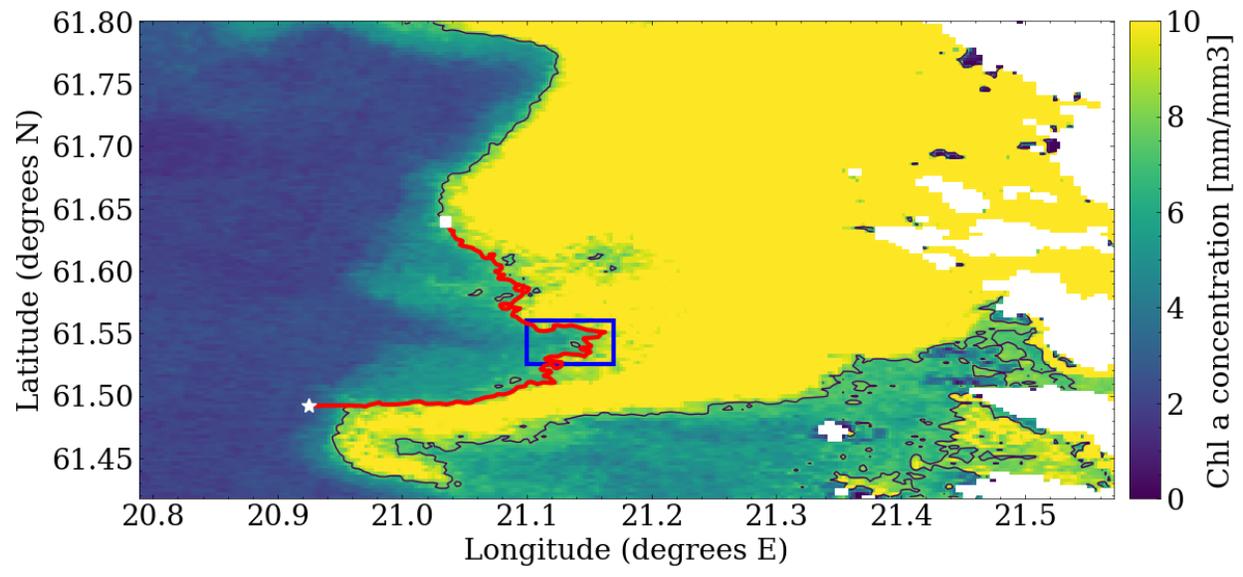


Fig. 15: Overview of the full mission having the trajectory of the AUV (red) tracking the front (black) in the chlorophyll *a* field (blue-yellow). The white star indicates the initial position, and the white square the final position.

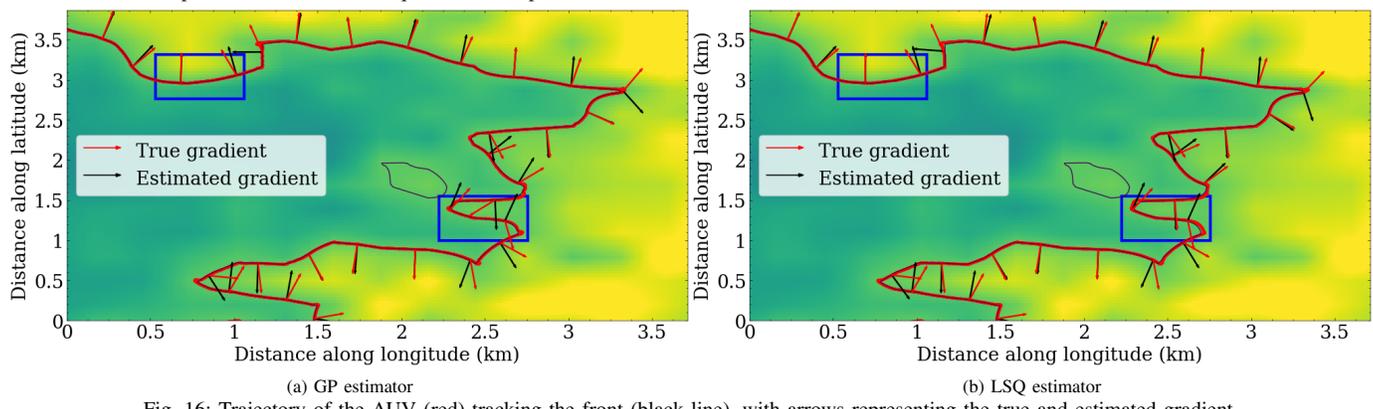


Fig. 16: Trajectory of the AUV (red) tracking the front (black line), with arrows representing the true and estimated gradient.

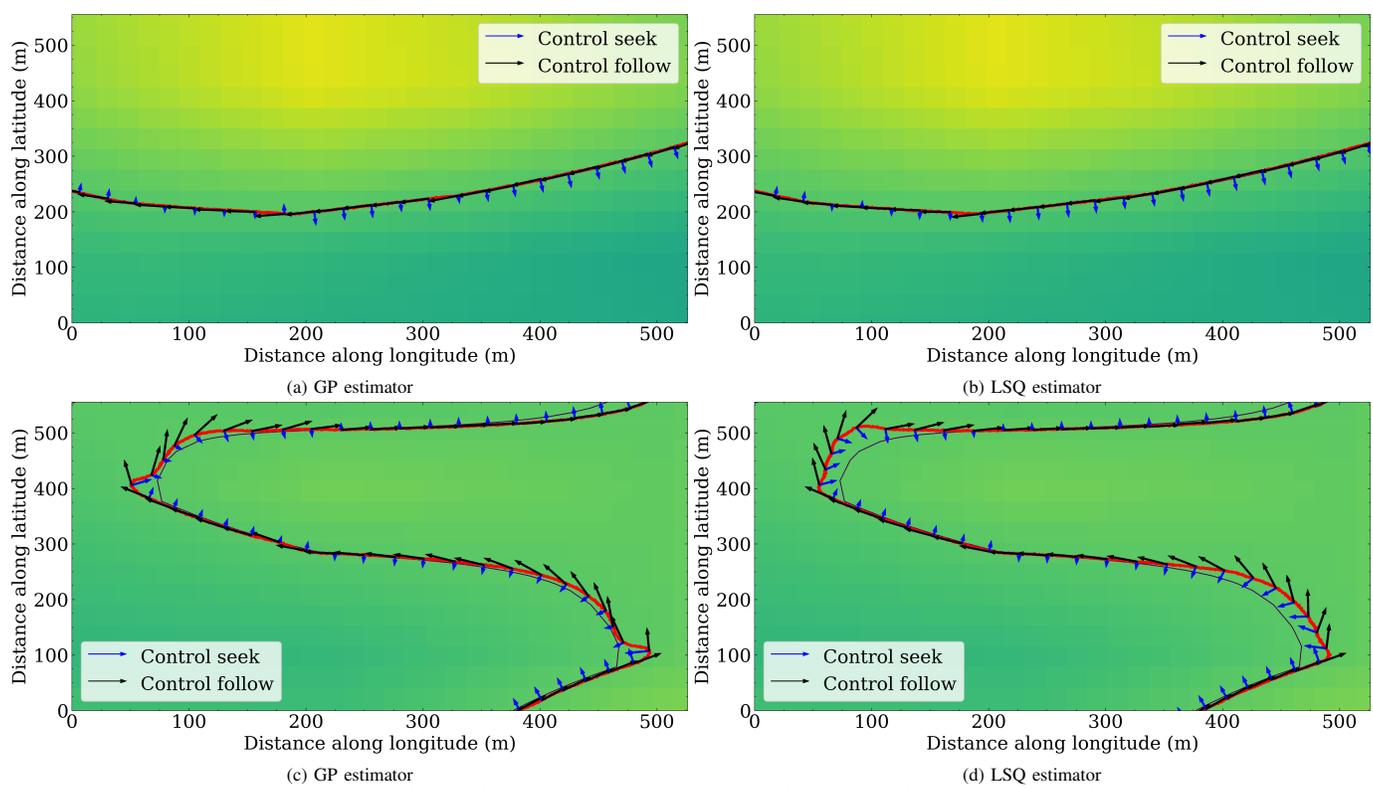
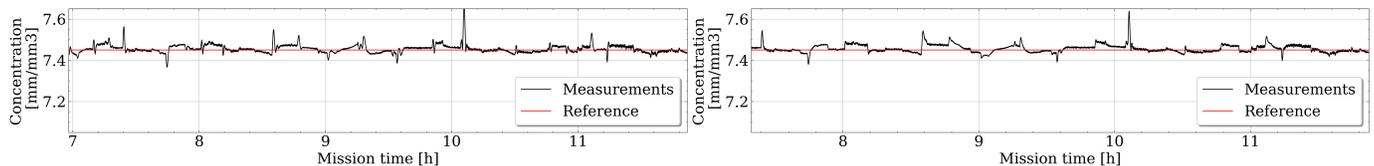
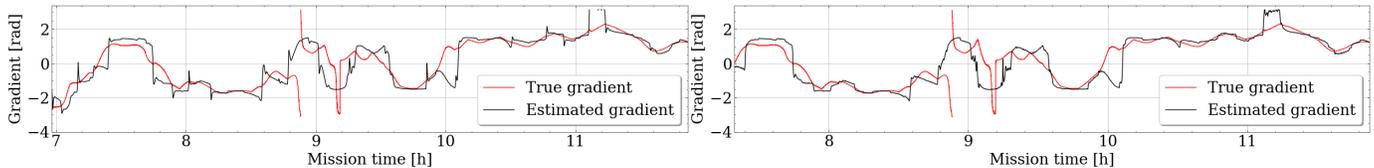


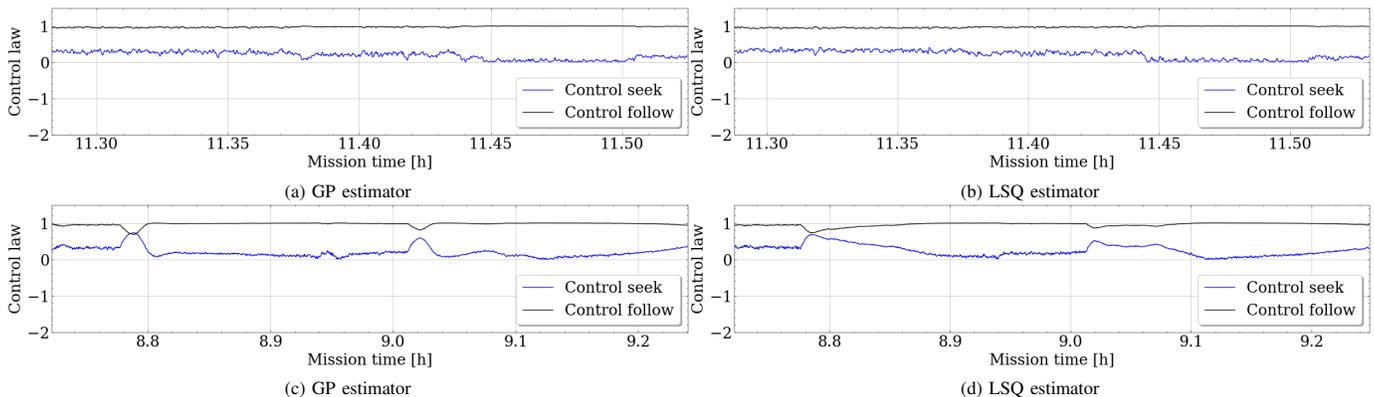
Fig. 17: AUV path (red) tracking the front (black), with arrows representing seek and follow components of the control law.



(a) GP estimator (b) LSQ estimator  
Fig. 18: Concentration of chlorophyll  $a$ : measurements from the AUV, and reference value.



(a) GP estimator (b) LSQ estimator  
Fig. 19: Gradient of chlorophyll  $a$ : AUV estimated gradient, and true gradient.



(a) GP estimator (b) LSQ estimator  
(c) GP estimator (d) LSQ estimator  
Fig. 20: Control law components: seek and follow.

the algorithm's performance becomes a trade-off between the smoothness introduced by the update function with a lower update rate and the delay it introduces. For this scenario, we considered smoothness of movement a more important objective than the apparent delay in the tighter curvature of the front. As for the performance comparison between the two gradient estimation methods, it is apparent from Fig. 17 that the GP gradient estimator allows for closer tracking of the front, most notably at regions with fast-changing curvature.

We further analyze the algorithm's behavior through time series plots in Fig. 18 and Fig. 19 corresponding to the zoom-in area in Fig. 16. First, we consider the chlorophyll  $a$  concentration measurements taken along the path in Fig. 18. The time series indicates that, as seen in the previous figures, the AUV is always on top of the front, oscillating around it and with a minimal error, in this case, lower than  $\pm 0.1 \text{ mg/m}^3$ . Second, we consider the gradient field estimation also taken along the mission, in Fig. 19. This time series also confirms what we saw in the previous figures. Here we can see both the delay of the estimated gradient and its smoothness compared with the true gradient. No major performance differences exist among the gradient estimators for either of the time series plots. Finally, let us further analyze the control law, considering the time series of the two control components in Fig. 20, for the regions defined in Fig. 17. In Fig. 20a and Fig. 20b, we consider the first zoom-in area with an almost linear segment of the front. Here we get a near-constant behavior of both controller components, where the follow component

is almost always 1, and the seek component is near zero most of the time. In Fig. 20c and Fig. 20d, we consider the second zoom-in, which contains two tight curves. Here the follow component also dominates the control law, with exceptions in two instances at  $t = 8.79$  and  $t = 9.02$ , corresponding to the two peaks in trajectory curvature, in Fig. 17c and Fig. 17d. For both examples, the follow component dominates the control law, and the seek component increases when the AUV is far from the front. Here, the difference in performance of the two gradient estimators is not visible. Thus, the following subsection is dedicated to properly evaluating the behavior of the different estimators.

### C. Sensitivity analysis

In this subsection, we analyze the performance of the estimation algorithms for a varying standard deviation of chlorophyll  $a$  concentration sensor noise.

The sensor noise we consider for this analysis varies between  $10^{-3} \text{ mg/m}^3$  and  $10^{-1} \text{ mg/m}^3$ . This is in line with the chlorophyll  $a$  concentration sensors in the market, which have a resolution of  $10^{-2} \text{ mg/m}^3$ . To analyze the impact of the noise introduced by the sensor, we remove the moving average window used on the measurements introduced by eq. 12. We then run multiple simulations using different standard deviations of the sensor noise and estimation methods. For this analysis, we consider two estimation methods: Gaussian Process regression, which is used in this paper, and Least Squares, which is used in the literature. We obtain both the

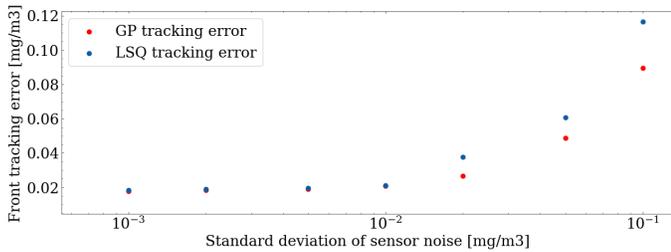


Fig. 21: Influence of sensor noise in the tracking error, for two different estimation algorithms: GP and LSQ.

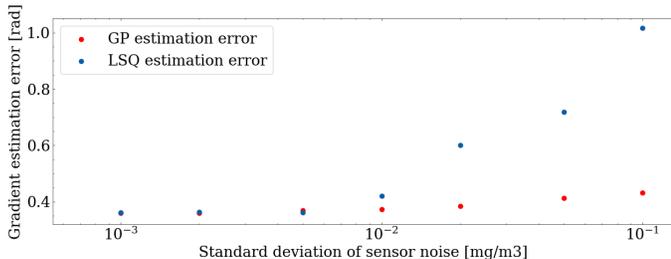


Fig. 22: Influence of sensor noise in the gradient estimation error, for two different estimation algorithms: GP and LSQ.

gradient estimation and tracking errors from each simulation. Each simulation results in a data point in this subsection's results.

The gradient estimation error corresponds to the difference between the true gradient  $\nabla\delta(t)$  and the estimated gradient  $\nabla\bar{\delta}(t)$ . In Fig. 21, we illustrate the impact of sensor noise on this error. Here the tracking error increases exponentially with the sensor noise for both estimators. The difference in performance between estimators becomes exponentially bigger from the standard deviation of sensor noise of 0.02mg/m<sup>3</sup>.

The tracking error corresponds to the difference between the measured chlorophyll *a* concentration  $\delta(t)$  and the chlorophyll *a* reference  $\delta_{\text{ref}}$ . In Fig. 22, we illustrate the impact of sensor noise on this error. Here the tracking error increases exponentially with the sensor noise for both estimators (note that the x-axis is in logarithmic scale). The difference in performance between estimators becomes much bigger from the standard deviation of sensor noise of 0.01mg/m<sup>3</sup>.

From this comparison, we can conclude that the GP estimator performs better than the LSQ-based estimator for gradient estimation and front tracking. We can also conclude that the error grows substantially from a standard deviation of sensor noise of 0.01mg/m<sup>3</sup>.

## V. EXPERIMENTAL RESULTS IN THE BALTIC SEA

This section contains the results obtained from experiments using the components introduced in the previous sections, summarised in Fig. 7. Along with the numerical simulation study, the gradient estimator and motion controller were validated in field experiments. These experiments took place in the Stockholm archipelago in the Baltic Sea, near the island of Djurö, in an area, with slightly different starting positions. We ensured safety by following the AUV by boat as in Fig.23, by reducing the AUV's speed to approximately 0.1m/s, and by constraining the survey area to a small region with low ferry traffic. We also ran the surveys on the surface so that



Fig. 23: The AUV in the water on the mission day, while tethered to the boat.

it is possible to get position feedback using GPS, thereby counteracting the uncertainty from exclusively using dead reckoning. To satisfy the constraints on maximum speed and small survey area, we scaled the data in Fig.14 100 times. This data was used to simulate the chlorophyll *a* concentration sensor as a look-up table, given that there are no algal blooms in this small area.

In Fig. 24, we illustrate two surveys and a simulated scenario. The first survey, in Fig. 24a, was conducted on July 18th, 2022. It corresponds to about 12 minutes of mission time at an average speed of 0.11m/s. The second survey, in Fig. 24b, was conducted on August 11th, 2022. It corresponds to about 10 minutes of mission time at an average speed of 0.11m/s. To better analyze and understand the results, we also present a simulated scenario Fig. 24c in which we set all the conditions to match the conditions of the experiments as closely as possible. These include the scaled data as a look-up table for the simulated sensor, the initial position, GPS noise, and a similar survey length of approximately 10 minutes.

Let us now analyze the results by zooming in on the survey areas and examining the components of our algorithm. In the first zoom-in in Fig. 25, we evaluate the trajectory of the AUV and gradient estimation. The trajectory of the real AUV in both surveys, in Fig. 25a and Fig. 25b, appears very jittery as opposed to previous chapters in which we simulated a mission. This effect is due to two phenomena: the noise in the GPS signal and water currents. Water currents drag the AUV to move in a different direction than the one the algorithm calculated. These currents are also one of the causes for the jittery GPS signal as, in the case of our AUV, the GPS receiver is a few centimeters above the surface, and the signal quality is directly influenced by the existence of waves that can partially cover the signal. In Fig. 25c, we attempted to emulate this phenomenon by introducing Gaussian noise on the GPS receiver.

We can also notice how the front tracking errors appear large for both surveys and the simulated scenario, as the AUV path

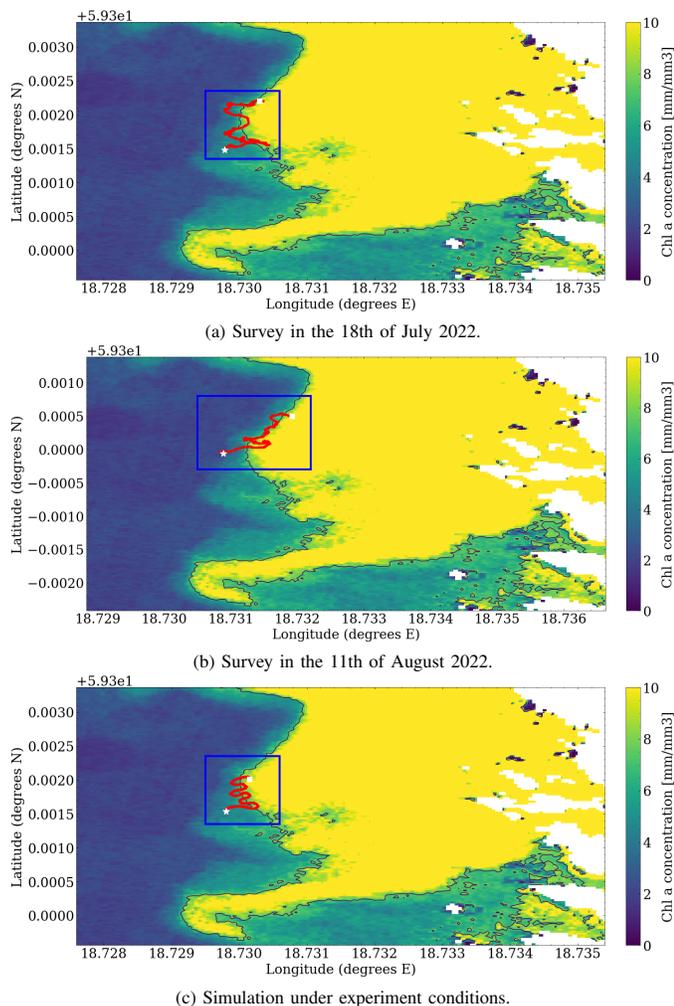


Fig. 24: Two experimental surveys and a simulated scenario under experimental conditions with AUV trajectory (red) tracking the front (black line) in the chlorophyll  $a$  map (blue-yellow).

resembles a sinusoid around the front. In the previous section, we mentioned that the front tracking error in simulations is due to the AUV turning radius and the delay introduced by the update function of the gradient estimator. Here, beyond those factors, we also have introduced a look-up table to simulate the sensor that consists of the chlorophyll  $a$  concentration dataset, which has been scaled 100 fold. Considering that, if we compare the results of the previous section and the simulated scenario, the overshoot always tends to be approximately 10 meters. We can further understand the implications of these overshoots by analyzing the chlorophyll  $a$  concentration tracking error in Fig. 27. Here, the overshoots and undershoots in chlorophyll  $a$  concentration correspond to the overshoots and undershoots in the distance while tracking the front. Bringing our attention to the gradient estimation, even with a jittery trajectory, the gradient error - the difference between the true gradient and estimated gradient - always seems below 90 degrees for all cases. For a more detailed analysis, in Fig. 28, we can see how the estimated gradient resembles a low-pass filtered version of the true gradient with very high frequencies.

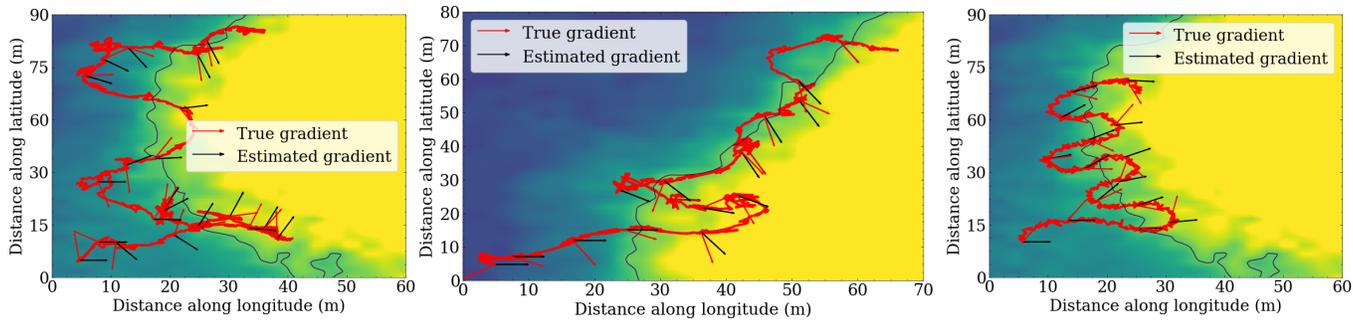
In the second zoom-in in Fig. 26, we evaluate the performance of the control law and its components. Here we note that the control law's seek and follow components match the

jittery trajectory. The follow component tends to be parallel to the front, pointing forward, and the seek component tends towards the front. Analysing Fig. 29, we can see how the over and undershoots correspond to times where seek is the dominating component.

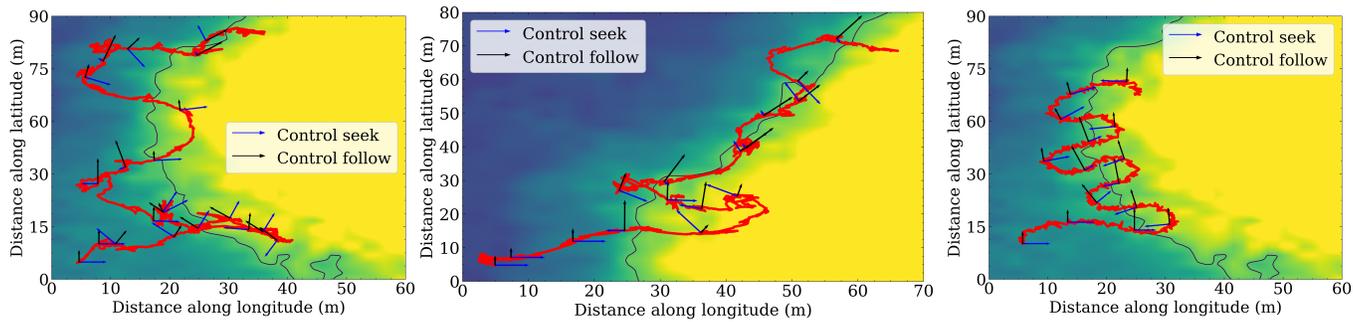
## VI. CONCLUSIONS AND FUTURE WORK

We considered the problem of how to use satellite data to improve adaptive sampling missions of an AUV equipped with a chlorophyll  $a$  sensor and how to design a survey in the Baltic Sea to test our approach. We developed the adaptive sampling algorithm and software packages to build a solution for the environmental sampling problem. Our solution uses GPs to model chlorophyll  $a$  fronts using satellite data and integrates such model into a front tracking algorithm. This integration is done using the estimate of the chlorophyll  $a$  gradient field in the control law. We confirmed the goodness of fit of the GP model by using scattered data points from a higher resolution satellite data and were able to reconstruct the chlorophyll  $a$  field using the GP model. We implemented the developed algorithm in the AUV's software and ran realistic simulations using the model of our AUV and chlorophyll  $a$  sensor model. These simulations resulted in accurate front tracking with low gradient estimation error. We considered the two most important performance metrics for our objective to be gradient estimation errors and front tracking errors. Concerning these metrics, the sensor noise analysis indicated that the gradient estimation using GP results in smaller errors than when using LSQ, mainly when the sensor noise is bigger or equal to  $0.01\text{mg/m}^3$ , which corresponds to most chlorophyll  $a$  concentration sensors in the market. We did not consider other performance metrics, such as time of computation, because both methods appeared fast enough to generate an estimate in real time. We also did not consider the computation time for model fitting prior to the survey as it's not running in real-time and thus not affecting performance. However, for some applications with fast-changing environments, it could be relevant to train the model during the survey. The GP estimator requires a GP model to be fit prior to the mission, which takes a few minutes, as opposed to the LSQ algorithm, which requires no prior fitting. After the development, implementation, and sensitivity analysis, we designed a survey in the Baltic Sea, near Stockholm, next to the island of Djurö. Similarly to the simulations, the experiments confirmed that the algorithm and software package work as desired in a controlled environment.

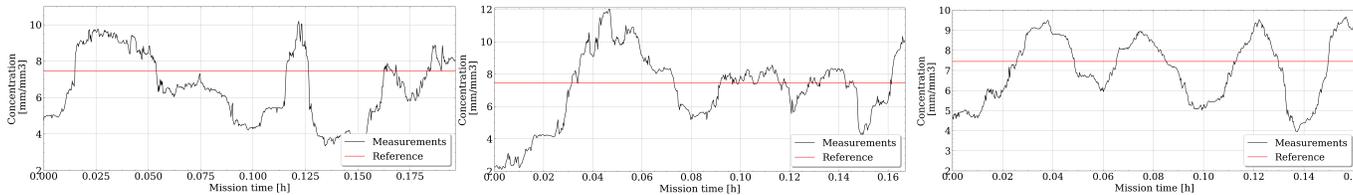
The work in this paper suggests several extensions, from theory to implementation. Starting with more theoretical contributions, it should be noted that our approach was developed for front tracking. Though it cannot directly be used for other applications such as source tracking, zig-zagging around the front, or other movements, the control law can be adapted for such applications. Another interesting extension to consider is the multi-agent scenario. This includes modifying the control law according to the neighbor agents and using the joint variance given by the GP model. Then the agents would be able to make more informed decisions regarding area coverage, knowing the areas with the highest variance. A



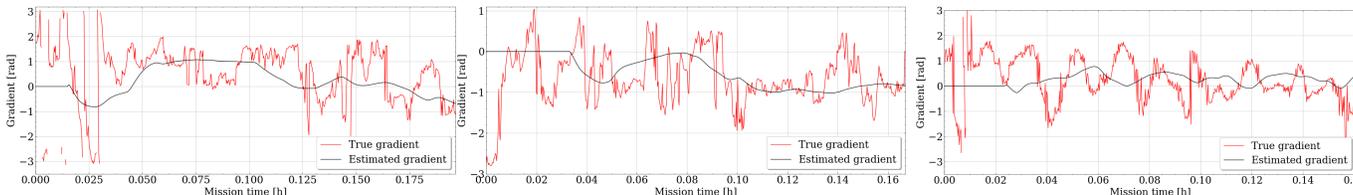
(a) Survey in the 18th of July 2022. (b) Survey in the 11th of August 2022. (c) Simulation under experiment conditions.  
 Fig. 25: Trajectory of the AUV (red) tracking the front (black line), with arrows representing the true and estimated gradient.



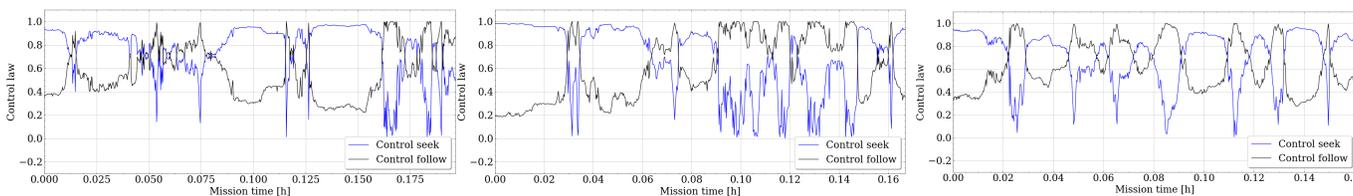
(a) Survey in the 18th of July 2022. (b) Survey in the 11th of August 2022. (c) Simulation under experiment conditions.  
 Fig. 26: Trajectory of the AUV (red) tracking the front (black line), with arrows representing the true and estimated gradient.



(a) Survey in the 18th of July 2022. (b) Survey in the 11th of August 2022. (c) Simulation under experiment conditions.  
 Fig. 27: Concentration of chlorophyll a: measurements from the AUV, and reference value.



(a) Survey in the 18th of July 2022. (b) Survey in the 11th of August 2022. (c) Simulation under experiment conditions.  
 Fig. 28: Gradient of chlorophyll a: AUV estimated gradient, and true gradient.



(a) Survey in the 18th of July 2022. (b) Survey in the 11th of August 2022. (c) Simulation under experiment conditions.  
 Fig. 29: Control law components: seek and follow.

contribution that follows directly from multi-agent systems is data assimilation using the data collected by the multi-agent system. This data could fit a GP model that informs algae population models through physics-informed learning. On the implementation side, future work would include integrating the algae sensor into an AUV, tuning it in a controlled setting, and running experiments with the complete system. One relevant contribution would be to work towards a higher degree of autonomy. There are several steps to reach a level of robustness sufficient for real-world deployment of higher autonomy. Some of them are reliable collision avoidance for islands, boats, and people, robust autonomous dock-in for charging, and a cloud-based data storage solution.

#### ACKNOWLEDGMENTS

We want to thank Josefine Severholt, Carl Ljung, Andreas Lezdins, and Koray Kulbay for their help with AUV hardware and support on the mission day, and Pedro Roque and Aldo Espinoza for their help with the AUV software, particularly with the control packages. We would also like to thank CMEMS for the available satellite data.

#### REFERENCES

- [1] P. Cornillon, "Fronts in the world ocean's large marine ecosystems," vol. 21, 2007.
- [2] C. C. Chapman, M.-A. Lea, A. Meyer, J.-B. Sallée, and M. Hindell, "Defining southern ocean fronts and their influence on biological and physical processes in a changing climate," *Nature Climate Change*, vol. 10, no. 3, pp. 209–219, 2020.
- [3] S. Shumway, J. M. Burkholder, and S. Morton, *Harmful Algal Blooms: A Compendium Desk Reference*, 2018.
- [4] NOAA. What is a harmful algal bloom? [Online]. Available: <https://www.noaa.gov/what-is-harmful-algal-bloom>
- [5] W. Munk, "Testimony to the u.s. commission on ocean policy." 2002. [Online]. Available: [http://govinfo.library.unt.edu/oceancommission/meetings/apr18\\_19\\_02/munk\\_statement.pdf](http://govinfo.library.unt.edu/oceancommission/meetings/apr18_19_02/munk_statement.pdf)
- [6] N. Wasmund, *Harmful Algal Blooms in Coastal Waters of the South-Eastern Baltic Sea*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2002, pp. 93–116.
- [7] M. Ashphaq, P. K. Srivastava, and D. Mitra, "Review of near-shore satellite derived bathymetry: Classification and account of five decades of coastal bathymetry research," *Journal of Ocean Engineering and Science*, vol. 6, no. 4, pp. 340–359, 2021.
- [8] A. K. Liu, C. Y. Peng, and S.-S. Chang, "Wavelet analysis of satellite images for coastal watch," *IEEE Journal of Oceanic Engineering*, vol. 22, no. 1, pp. 9–17, 1997.
- [9] A. Purser, Y. Marcon, S. Dreutter, U. Hoge, B. Sablotny, L. Hehemann, J. Lemburg, B. Dorschel, H. Biebow, and A. Boetius, "Ocean floor observation and bathymetry system (ofobs): a new towed camera/sonar system for deep-sea habitat surveys," *IEEE Journal of Oceanic Engineering*, vol. 44, no. 1, pp. 87–99, 2018.
- [10] H. Sun, D. C. Hendry, M. A. Player, and J. Watson, "In situ underwater electronic holographic camera for studies of plankton," *IEEE Journal of Oceanic Engineering*, vol. 32, no. 2, pp. 373–382, 2007.
- [11] R. N. Smith and V. T. Huynh, "Controlling buoyancy-driven profiling floats for applications in ocean observation," *IEEE Journal of Oceanic Engineering*, vol. 39, no. 3, pp. 571–586, 2013.
- [12] D. Roemmich, G. C. Johnson, S. Riser, R. Davis, J. Gilson, W. B. Owens, S. L. Garzoli, C. Schmid, and M. Ignaszewski, "The argo program: Observing the global ocean with profiling floats," *Oceanography*, vol. 22, no. 2, pp. 34–43, 2009.
- [13] J. Sherman, R. E. Davis, W. Owens, and J. Valdes, "The autonomous underwater glider "spray"," *IEEE Journal of Oceanic Engineering*, vol. 26, no. 4, pp. 437–446, 2001.
- [14] D. C. Webb, P. J. Simonetti, and C. P. Jones, "Slocum: An underwater glider propelled by environmental energy," *IEEE Journal of oceanic engineering*, vol. 26, no. 4, pp. 447–452, 2001.
- [15] C. C. Eriksen, T. J. Osse, R. D. Light, T. Wen, T. W. Lehman, P. L. Sabin, J. W. Ballard, and A. M. Chiodi, "Seaglider: A long-range autonomous underwater vehicle for oceanographic research," *IEEE Journal of Oceanic Engineering*, vol. 26, no. 4, pp. 424–436, 2001.
- [16] N. E. Leonard and J. G. Graver, "Model-based feedback control of autonomous underwater gliders," *IEEE Journal of oceanic engineering*, vol. 26, no. 4, pp. 633–645, 2001.
- [17] F. P. Chavez, J. T. Pennington, R. Herlien, H. Jannasch, G. Thurmond, and G. E. Friederich, "Moorings and drifters for real-time interdisciplinary oceanography," *Journal of Atmospheric and Oceanic Technology*, vol. 14, no. 5, pp. 1199–1211, 1997.
- [18] J. S. Willcox, J. G. Bellingham, Y. Zhang, and A. B. Baggeroer, "Performance metrics for oceanographic surveys with autonomous underwater vehicles," *IEEE Journal of Oceanic Engineering*, vol. 26, no. 4, pp. 711–725, 2001.
- [19] J. Das, F. Py, T. Maughan, T. O'reilly, M. Messié, J. Ryan, G. S. Sukhatme, and K. Rajan, "Coordinated sampling of dynamic oceanographic features with underwater vehicles and drifters," *The International Journal of Robotics Research*, vol. 31, no. 5, pp. 626–646, 2012.
- [20] Ö. Özkahraman and P. Ögren, "Efficient navigation aware seabed coverage using auvs," in *2021 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*, 2021, pp. 63–70.
- [21] P. Yao, L. Qiu, J. Qi, and R. Yang, "Auv path planning for coverage search of static target in ocean environment," *Ocean Engineering*, vol. 241, p. 110050, 2021.
- [22] J. Hwang, N. Bose, and S. Fan, "Auv adaptive sampling methods: A review," *Applied Sciences*, vol. 9, no. 15, p. 3145, 2019.
- [23] H. Feng, J. Yu, Y. Huang, J. Qiao, Z. Wang, Z. Xie, and K. Liu, "Adaptive coverage sampling of thermocline with an autonomous underwater vehicle," *Ocean Engineering*, vol. 233, p. 109151, 2021.
- [24] P. Ögren, E. Fiorelli, and N. E. Leonard, "Cooperative control of mobile sensor networks: Adaptive gradient climbing in a distributed environment," *IEEE Transactions on Automatic Control*, vol. 49, no. 8, pp. 1292–1302, 2004.
- [25] E. Fiorelli, N. E. Leonard, P. Bhatta, D. A. Paley, R. Bachmayer, and D. M. Fratantoni, "Multi-auv control and adaptive sampling in monterey bay," *IEEE Journal of Oceanic Engineering*, vol. 31, no. 4, pp. 935–948, 2006.
- [26] T. O. Fossum, P. Norgren, I. Fer, F. Nilsen, Z. C. Koenig, and M. Ludvigsen, "Adaptive sampling of surface fronts in the arctic using an autonomous underwater vehicle," *IEEE Journal of Oceanic Engineering*, vol. 46, no. 4, pp. 1155–1164, 2021.
- [27] N. K. Yilmaz, C. Evangelinos, P. F. J. Lermusiaux, and N. M. Patrikalakis, "Path planning of autonomous underwater vehicles for adaptive sampling using mixed integer linear programming," *IEEE Journal of Oceanic Engineering*, vol. 33, no. 4, pp. 522–537, 2008.
- [28] A. Bennett and J. Leonard, "A behavior-based approach to adaptive feature detection and following with autonomous underwater vehicles," *IEEE Journal of Oceanic Engineering*, vol. 25, no. 2, pp. 213–226, 2000.
- [29] Y. Zhang, B. Kieft, B. W. Hobson, J. P. Ryan, B. Barone, C. M. Preston, B. Roman, B.-Y. Raanan, R. Marin III, T. C. O'Reilly, C. A. Rueda, D. Pargett, K. M. Yamahara, S. Poulos, A. Romano, G. Foreman, H. Ramm, S. T. Wilson, E. F. DeLong, D. M. Karl, J. M. Birch, J. G. Bellingham, and C. A. Scholin, "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*, vol. 45, no. 4, pp. 1308–1321, 2020.
- [30] S. McCammon, G. Marcon dos Santos, M. Frantz, T. P. Welch, G. Best, R. K. Shearman, J. D. Nash, J. A. Barth, J. A. Adams, and G. A. Hollinger, "Ocean front detection and tracking using a team of heterogeneous marine vehicles," *Journal of Field Robotics*, vol. 38, no. 6, pp. 854–881, 2021.
- [31] R. N. Smith, Y. Chao, P. P. Li, D. A. Caron, B. H. Jones, and G. S. Sukhatme, "Planning and implementing trajectories for autonomous underwater vehicles to track evolving ocean processes based on predictions from a regional ocean model," *The International Journal of Robotics Research*, vol. 29, no. 12, pp. 1475–1497, 2010.
- [32] J. Das, F. Py, J. B. Harvey, J. P. Ryan, A. Gellene, R. Graham, D. A. Caron, K. Rajan, and G. S. Sukhatme, "Data-driven robotic sampling for marine ecosystem monitoring," *The International Journal of Robotics Research*, vol. 34, no. 12, pp. 1435–1452, 2015.
- [33] T. O. Fossum, J. Eidsvik, I. Ellingsen, M. O. Alver, G. M. Fragosó, G. Johnsen, R. Mendes, M. Ludvigsen, and K. Rajan, "Information-driven robotic sampling in the coastal ocean," *Journal of Field Robotics*, vol. 35, no. 7, pp. 1101–1121, 2018.

- [34] T. D. Dickey, "Emerging ocean observations for interdisciplinary data assimilation systems," *Journal of Marine Systems*, vol. 40, pp. 5–48, 2003.
- [35] O. Schofield, T. Bergmann, P. Bissett, J. F. Grassle, D. B. Haidvogel, J. Kohut, M. Moline, and S. M. Glenn, "The long-term ecosystem observatory: an integrated coastal observatory," *IEEE Journal of Oceanic Engineering*, vol. 27, no. 2, pp. 146–154, 2002.
- [36] T. O. Fossum, G. M. Fragoso, E. J. Davies, J. E. Ullgren, R. Mendes, G. Johnsen, I. Ellingsen, J. Eidsvik, M. Ludvigsen, and K. Rajan, "Toward adaptive robotic sampling of phytoplankton in the coastal ocean," *Science Robotics*, vol. 4, no. 27, p. eaav3041, 2019.
- [37] K. D. Heaney, G. Gawarkiewicz, T. F. Duda, and P. F. Lermusiaux, "Non-linear optimization of autonomous undersea vehicle sampling strategies for oceanographic data-assimilation," *Journal of Field Robotics*, vol. 24, no. 6, pp. 437–448, 2007.
- [38] D. Chang, C. R. Edwards, F. Zhang, and J. Sun, "A data assimilation framework for data-driven flow models enabled by motion tomography," *International Journal of Intelligent Robotics and Applications*, vol. 3, no. 2, pp. 158–177, 2019.
- [39] B. Gips, "Texture-based seafloor characterization using gaussian process classification," *IEEE Journal of Oceanic Engineering*, vol. 47, no. 4, pp. 1058–1068, 2022.
- [40] [Online]. Available: [https://resources.marine.copernicus.eu/product-detail/BALTICSEA\\_ANALYSISFORECAST\\_BGC\\_003\\_007/INFORMATION](https://resources.marine.copernicus.eu/product-detail/BALTICSEA_ANALYSISFORECAST_BGC_003_007/INFORMATION)
- [41] Å. Bilaletdin, H. Kaipainen, and T. Frisk, "Dynamic nutrient modelling of a large river basin in finland," vol. 111, 2008, pp. 53–61.
- [42] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*. Cambridge, MA: MIT Press, 2006.
- [43] P. Erickson, M. Cline, N. Tirpankar, and T. Henderson, "Gaussian processes for multi-sensor environmental monitoring," in *2015 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI)*. IEEE, 2015, pp. 208–213.
- [44] T. Fossen, *Handbook of Marine Craft Hydrodynamics and Motion Control*. John Wiley & Sons Ltd., 2011.
- [45] S. Bhat, I. Stenius, and T. Miao, "Real-time flight simulation of hydrobatic auvs over the full  $0^{\circ}$ – $360^{\circ}$  envelope," *IEEE Journal of Oceanic Engineering*, vol. 46, no. 4, pp. 1114–1131, 2021.
- [46] J. Fonseca, M. Aguiar, J. B. d. Sousa, and K. H. Johansson, "Algal bloom front tracking using an unmanned surface vehicle: Numerical experiments based on baltic sea data," in *OCEANS 2021: San Diego – Porto*, 2021, pp. 1–7.
- [47] J. Fonseca, J. Wei, T. A. Johansen, and K. H. Johansson, "Cooperative circumnavigation for a mobile target using adaptive estimation," in *CONTROLO 2020*. Springer International Publishing, 2021, pp. 33–48.
- [48] P. Cieślak, "Stonefish: An Advanced Open-Source Simulation Tool Designed for Marine Robotics, With a ROS Interface," in *OCEANS 2019 - Marseille*, 2019.
- [49] S. Bhat, I. Stenius, N. Bore, J. Severholt, C. Ljung, and I. Torroba Balmori, "Towards a cyber-physical system for hydrobatic auvs," in *OCEANS 2019 - Marseille*, 2019, pp. 1–7.
- [50] S. Bhat, I. Torroba, Ö. Özkahraman, N. Bore, C. I. Sprague, Y. Xie, I. Stenius, J. S. J., C. Ljung, J. Folkesson, and P. Ögren, "A cyber-physical system for hydrobatic auvs: System integration and field demonstration," in *IEEE OES AUV2020 Symposium*, St. John's, NL, Canada, 2020.
- [51] C. I. Sprague, Ö. Özkahraman, A. Munafo, R. Marlow, A. Phillips, and P. Ögren, "Improving the modularity of auv control systems using behaviour trees," in *2018 IEEE/OES Autonomous Underwater Vehicle Workshop (AUV)*. IEEE, 2018, pp. 1–6.
- [52] Ö. Özkahraman and P. Ögren, "Combining control barrier functions and behavior trees for multi-agent underwater coverage missions," in *IEEE 59th Conference on Decision and Control*. IEEE, 2020.
- [53] I. Stenius, J. Folkesson, S. Bhat, C. I. Sprague, L. Ling, Ö. Özkahraman, N. Bore, Z. Cong, J. Severholt, C. Ljung, A. Arnwald, I. Torroba, F. Gröndahl, and J.-B. Thomas, "A system for autonomous seaweed farm inspection with an underwater robot," *Sensors*, vol. 22, no. 13, 2022.
- [54] [Online]. Available: [https://resources.marine.copernicus.eu/product-detail/OCEANCOLOUR\\_BAL\\_CHL\\_L3\\_NRT\\_OBSERVATIONS\\_009\\_049/INFORMATION](https://resources.marine.copernicus.eu/product-detail/OCEANCOLOUR_BAL_CHL_L3_NRT_OBSERVATIONS_009_049/INFORMATION)
- [55] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, A. Y. Ng *et al.*, "Ros: an open-source robot operating system," in *ICRA workshop on open source software*, vol. 3, no. 3.2. Kobe, Japan, 2009, p. 5.