Optimal AUV trajectory planning and execution control for maritime pollution incident response

Alexandre Rocha*, Miguel Aguiar†, Keila Lima*, Renato Mendes*‡, João M. Dias§, Magda C. Sousa§, João Borges de Sousa*

*LSTS, Underwater Systems and Technology Laboratory, Faculty of Engineering, University of Porto, Porto, Portugal
†Division of Decision and Control Systems, KTH Royal Institute of Technology, Stockholm, Sweden
‡Collaborative Laboratory +Atlantic, Matosinhos, Portugal
§CESAM, Centre for Environmental and Marine Studies, Physics Department, University of Aveiro, Aveiro, Portugal

***Abstract*—Marine pollution incidents can have a huge impact on different ecosystems, with unpredictable short- and long-term consequences. Once the pollutant is detected, it is critical to quickly understand its characteristics so that authorities can lay out an adequate response. In parallel to the time- and cost-constrained traditional operational means, this paper suggests the use of AUVs for the sampling procedures of marine pollution incidents, to increase the speed and efficiency of operations. A new software architecture is developed, integrating trajectory optimization for AUVs into a software toolchain that allows human operators to plan, simulate, and control multiple vehicles deployments. A method to optimize AUVs deployment position and time is also presented. The overall architecture is simulated using high-resolution hydrodynamic data from the Ria de Aveiro lagoon and the adjacent coastal area, in Aveiro, Portugal.**

***Index Terms*—trajectory optimization, networked vehicle system, marine pollution, ocean sampling**

I. Introduction

It is well-known that marine pollution presents a threat to our planet and society, endangering ecosystems, disrupting economies and affecting public health, in the short- and long-term. A variety of factors are at the origin of this problem, involving different types of pollutants, with varying quantities and impacts on the surrounding environments. Nevertheless, in situations where the pollutant has an easily identifiable source, such as oil spills, ship polluting discharges, and effluents from wastewater treatment plants, marine authorities can play a crucial role in counteracting pollution spread.

Once the pollutant enters the marine environment, it is extremely important to identify and sample it as soon as possible, so that an appropriate response action can be planned. However, currently, monitoring pollution events can be expensive and time-consuming, due to the spatiotemporal limitation of the traditional means (buoys equipped with sensors, vessels, and other nonautonomous vehicles). Here we suggest the use of autonomous underwater vehicles (AUVs) to facilitate the identification of polluting sources, while increasing the speed of operations and reducing their inherent costs.

In recent years, AUVs have taken large steps towards attaining technological maturity for scientific exploration, mainly in autonomy level, robustness, and sensors payload. Nonetheless, when operating in complex scenarios, with an irregular morphology and time-varying water velocities whose magnitude can surpass the vehicle's maximum speed, the safety of AUVs can be compromised. To overcome these obstacles, hydrodynamic models can be used to plan AUVs trajectories, not only to guarantee reachability but also to benefit from water velocities flow to minimize mission time.

In this paper, we address the problem of planning optimal trajectories for an AUV travelling from a deployment position to one or more pollution focuses, integrated into a newly developed software architecture that provides authorities with a rapid and efficient mechanism to respond to point source pollution alerts.

This work builds upon the results in Aguiar et al. [1], who presented an efficient method and implementation for computing time-optimal trajectories for AUVs using high-resolution water velocity models. Using the LSTS - Underwater Systems and Technology Laboratory - software toolchain [2], we incorporate the method into a networked vehicle system, optimizing vehicle deployments and enabling the operationalization of marine pollution incident in situ verification. The proposed solution aims to increase the overall response efficacy, enabling onshore human operators to plan, control and supervise every mission stage, while automating the trajectory planning process for AUVs deployments. We present an illustration of the proposed concept in figure 1.

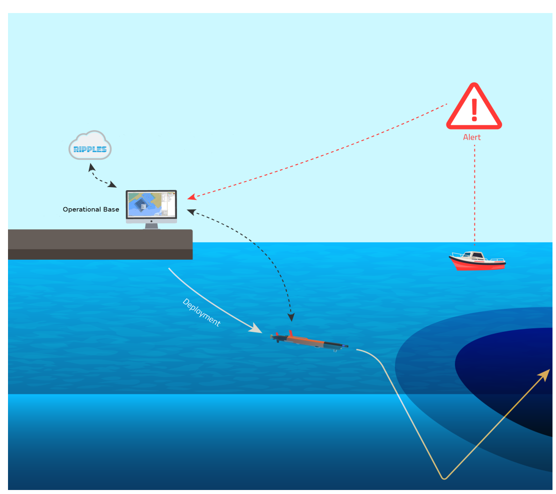


Figure 1: Response Mechanism Concept

Here we apply the response mechanism to an operations scenario in the Aveiro harbor, in the northwestern coast of Portugal. The harbor is located within a coastal lagoon (Ria de Aveiro) with a single artificial connection to the sea. The hydrodynamic of the Ria de Aveiro coastal lagoon is mostly tidal-driven and its intricate bathymetry and morphology and the water velocities constitute a challenge for AUVs operations. Thus, we base our approach on a high-resolution hydrodynamic numerical model, which is a custom implementation of the Delft3D model to the lagoon and adjacent coastal region. The model generates 4-D solutions for the water velocities of the study area [3].

The rest of the paper is organized as follows. In Section II we discuss other works that address marine pollution situations using AUVs. Section III briefly describes the mathematical trajectory problem formulation. In Section IV we present the software architecture components and give a step-by-step description of the flow of computation. We discuss the results in Section V and state our conclusions in the Section VI.

II. Related Work

The improvement of AUVs capabilities in the last years have enhanced their utility in pollution monitoring.

Vasilijević et al. [4] present a networked vehicle system to monitor subsurface pollution, based on the same software toolchain as our solution does. The system is composed by an AUV, an autonomous surface vehicle (ASV) and a support vessel. While the AUV takes underwater measurements of the pollutant, the ASV acts as the communications hub between the AUV and the support vessel. This system enables human operators to access near real-time data of the pollutant.

In [5], Bonin-Font et al. carried six missions with an AUV to gather visual information of the regression of an underwater flora species, as a result of sewage discharges. After the missions' completion, divers were sent to precise locations to take samples of interesting features captured by the AUV's camera. This presented an important gain in mission efficiency, reducing costs and the duration of the sampling tasks.

In [6], the authors developed MARES, an AUV aimed for environmental sampling in coastal waters. It is built in a modular fashion, so that those different sensors could be coupled according to the mission's objectives. Apart from the vehicle, the developed system includes a laptop as the control station, and requires the deployment of buoys equipped with acoustic navigation beacons, both for vehicle navigation and communication support. An overall one-vehicle-only control infrastructure was developed, with both ongoing mission feedback and post-mission analysis capacities.

The authors in [7] developed a guidance system for an AUV for water quality monitoring navigation, in the scope of aquaculture systems. They propose a behavior-based mission planning and guidance system, operating vehicles with low energy consumption, automatic and intelligent measurement and logging. The overall system was successfully tested in a mission involving Oxygen and Nitrate in situ measurements.

In all the referred papers an AUV-based system is presented for sampling environmental phenomena. Nonetheless, none of these gather the set of characteristics our solution does: operations with multiple vehicles, data centralization, and mission execution simulation and control. Moreover, the integration of trajectory optimization for AUVs is a feature that every AUV-based system would benefit from.

III. Trajectory Optimization

Prior to describing the overall software architecture, we start by briefly stating the trajectory optimization problem this paper addresses.

From an operational perspective, the research question that arises is the following: given a forecast of the water velocities, what is the optimal trajectory for an AUV to travel from some deployment position and time to a target located in a polluted area?

To solve this problem, we describe it in an optimal control setting. We consider only planar motions based on the assumption that the vehicle travels at a constant depth, yet extending this problem to three dimensions would be straightforward. Let $\boldsymbol{x}(t) \in \mathbb{R}^2$ denote the horizontal position of the vehicle at the time instant $t \in \mathbb{R}$, and consider the following vehicle's kinematics:

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{u}(t) + \boldsymbol{\nu}(\boldsymbol{x}(t), t) \tag{1}$$

where $\boldsymbol{\nu}(x(t), t)$ is the water velocity flow and $\boldsymbol{u}(t) \in \mathbf{R}^2$ is the control constrained to $|\boldsymbol{u}(t)| \leq \boldsymbol{u}_{max}$, $\boldsymbol{u}_{max}$ being the vehicle's maximum velocity.

There are multiple different feasible trajectories between the initial position $\boldsymbol{x}(t_0)$ and the predefined target. Therefore, in order to obtain the optimal solution, we minimize the following cost function:

$$\begin{aligned} J(\boldsymbol{x}(t_0), t_0, \boldsymbol{u}(t_0)) = q(\boldsymbol{x}(T), T, \boldsymbol{u}(T)) \\ + \int_{t_0}^{T} g(\boldsymbol{x}(\tau), \tau, \boldsymbol{u}(\tau))\, d\tau \end{aligned} \tag{2}$$

where q is a nonnegative function representing the arrival cost, g is a positive function representing the running cost, and T is the arrival time. By setting $q(\cdot) \equiv 0$ and $g(\cdot) \equiv 1$, we obtain a minimum-time control problem.

To take into account common geographic time-varying constraints, such as no-go zones and dynamical obstacles, we add these areas' coordinates to the target set, and we attribute them a substantially greater arrival cost than the one on safer locations. This guarantees that the cost of travelling through these constrained environments is always greater than the other alternatives.

At this point, the minimization is computed with the numerical solver presented in [1] and results in a value function, $V(\boldsymbol{x}(t), t)$, representing the cost of the optimal trajectory starting from every $(\boldsymbol{x}(t), t)$ among the given set of deployment positions and times, i.e., the optimal trajectory duration starting from every point of that set. Then, as shown in the same work, the control input depends only on the gradient of $V(\boldsymbol{x}(t), t)$ and the vehicle's maximum forward speed.

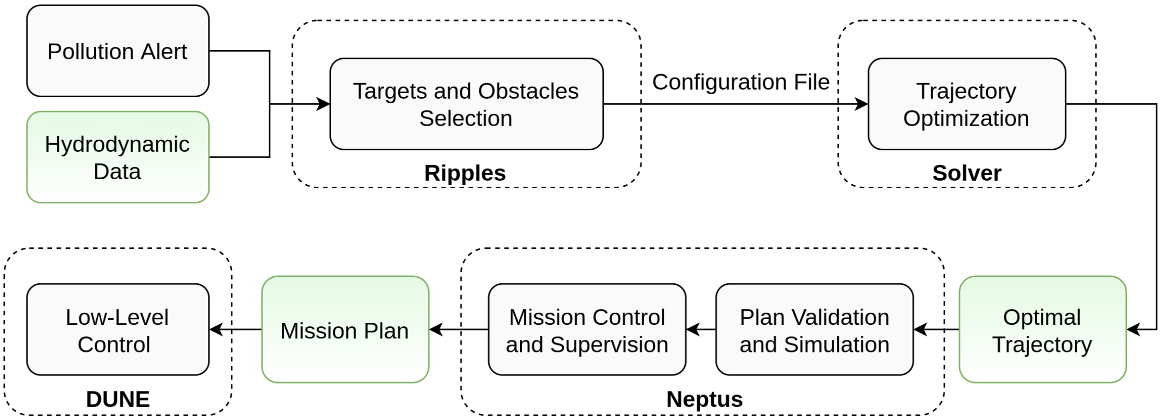


Figure 2: Software architecture block diagram

Therefore, having $V(\boldsymbol{x}(t), t)$, the trajectory can be obtained by integrating (1) with the initial condition $(\boldsymbol{x}(t_0), t_0)$, until $\boldsymbol{x}(T)$ is reached.

One important remark is that the value function provides a very useful auxiliary tool for mission planning and validation. It enables human operators to evaluate reachability, starting at different temporal and spatial locations, as well as estimating the time needed to accomplish the planned missions, including those involving the sampling of multiple pollution focuses. This is of course of great utility in operating AUVs on applications like ours. This is exemplified in the numerical examples we present below.

IV. Software Architecture

The proposed software architecture aims to enable the operationalization of trajectory optimization for AUVs, integrating the numerical solver in [1] into the LSTS software toolchain [8]. This toolchain, being used for operations of networked heterogeneous unmanned vehicles, includes human operators in the control and planning loops of the missions. These tools enable a more thorough analysis regarding vehicles deployments and operations which can be crucial in the decision making and coordination of in situ samplings.

As an additional point, the LSTS toolchain allows the monitoring of the execution of the planned trajectories, giving feedback throughout the command chain and allowing the assessment of the mission progress. From the multiple off- and on-board tools this toolchain offers, we focus on the ones where the high-level planning and monitoring of the mission goals is performed: Neptus and Ripples [8].

The numerical solver in [1] was adapted so that it could be integrated in an intermediate computational stage, and application-specific modifications were made, mainly to facilitate user experience. A block diagram of the overall architecture is presented figure 2.

We now explain the different used components and their roles in the system. Then, we describe the overall architecture and the flow of computation.

A. Neptus

Neptus is a command and control infrastructure which allows human operators to command and supervise a networked vehicle system. It supports all phases of a typical mission's life cycle, ranging from planning, simulation and execution to review and post-mission analysis.

Among its wide spectrum of features, some are particularly interesting to our application. In the first place, with this tool, humans operators can validate distinct mission plans using an integrated plan simulator. Secondly, while in the execution phase, Neptus can display real-time data from the vehicles, estimate their state in case of communication loss and give high-level maneuver commands to the vehicle. To end, another useful feature is the possibility to display real-time Automatic Identification System (AIS) data from nearby ships. This can be especially useful in an ongoing mission scenario, where the operational area is located near the coastline, and possibly near some port.

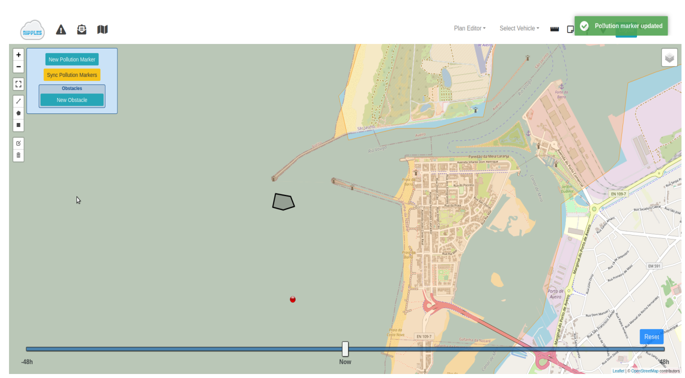


Figure 3: Ripples GUI

B. Ripples

Ripples is a web infrastructure that serves as a communication hub for data dissemination and situation alert between networked systems, aggregating data coming from different communication links: Web, GSM or Iridium. The versatility

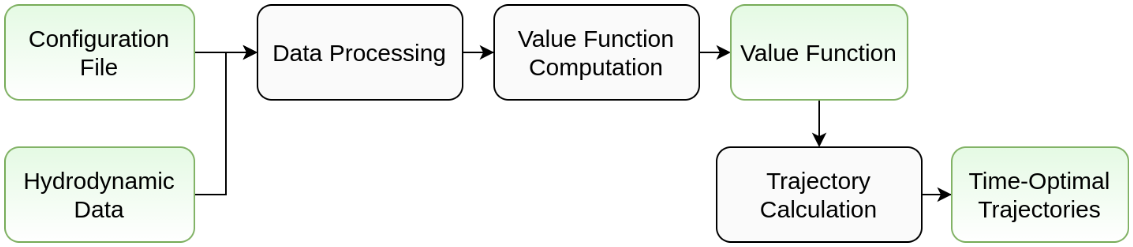


Figure 4: Trajectory optimization module block diagram

of using different links is crucial to guarantee remote communications with nodes in different locations, where different communication means are available, maintaining a global situation awareness.

Besides the aggregation of data from the deployed systems, Ripples can also overlay information from other sources such as marine traffic, oceanic forecasts or bathymetry.

Additionally, Ripples contains a GUI that allows to remotely update information and give feedback on the mission stages. Figure 3 shows the GUI, including one obstacle, represented by the polygon in black, and a target point, represented by the red dot.

C. Numerical solver integration

The numerical solver in [1] presents an efficient method and implementation for computing time-optimal trajectories for AUVs using high-resolution water velocity models. In order to use it integrated into this software toolchain, there was the need to develop additional features for data processing, so that the different inputs fulfilled the solver requirements, and its outputs matched the other software components' characteristics. A block diagram describing the sequence of computations of this software module is presented in 4.

We start by describing the new features that take place prior to the computations and we finish by explaining how are the outputs provided.

The first added feature was a configurable JSON file that acts as the entry point of the software, where different mathematical and computational parameters are specified. These are critical parameters to the value function and trajectories calculation, and thus enabling human operators to change them in a simple manner is critical. Some examples of these parameters are the minimum water level considered safe for vehicle's operations, the vehicle's maximum forward speed, and the spatiotemporal resolution with which the hydrodynamic model is interpolated.

Depending on the scenario, it may be necessary to visit more than one pollution focus, possibly to collect measurements of the pollutant at different times and locations. This requires consecutive value functions computations, resulting in one function for each target and the multiple trajectories from the deployment position to the first focus, between focuses, and back to the initial position. To do that, the ordered sampling sites' coordinates are now introduced as parameters as well. The solver will then iterate through these coordinates and output the respective value functions and trajectories.

Moreover, human operators may need to include obstacles in the operational area which might not be considered in the hydrodynamic model, e.g. an anchored ship and buoys. Hence, insertion of obstacles represented by polygons was implemented, their coordinates being inputted once again as parameters of the entry point file.

In a nutshell, by centralizing information we are simplifying the actuation of the response mechanism, making mission preparation an uncomplicated process.

Apart from this file, the other input of the numerical solver is the hydrodynamic model. Since the solver expects the data to be represented on a regular grid, we interpolate it both in space and time, according to the spatial and temporal resolutions previously inserted in the configurable file. This input will be available in the HDF5 file format.

There is one additional feature that was implemented - the obstacle insertion mentioned previously. Using the polygons vertices coordinates inserted in the configuration file, we use the Generic Mapping Tools[1] to obtain an obstacle binary mask of the operational area, with the same spatial and temporal resolutions as the ones used in the model interpolation. This mask will set the points inside the polygon as 1 and the points outside it to 0. With this information, we proceed as described in Section III to include this in the mathematical problem formulation.

After completing all the computations, this module will provide both outputs - the trajectories and the value functions - to Ripples. Regarding the trajectories, it will discretize them into heterogeneously distanced waypoints depending on trajectory curvature, and save them into a JSON file. The value function is stored using the HDF5 file format.

To better understand how the overall system works, we now present a step-by-step description of the flow of computation.

D. Flow of computation

We consider a scenario where one vehicle is deployed to sample multiple focuses of pollution.

First, a pollution alert is reported to marine authorities, that will decide which locations should be sampled. Then, a human operator inserts the focuses' coordinates in Ripples, alongside existing obstacles. The respective parameters are automatically

[1] https://www.generic-mapping-tools.org/

added to the configurable JSON file of the numerical solver. At this stage, this file must be filled with the rest of the necessary parameters for the numerical computation. Nevertheless, it is important to highlight that part of these parameters can be set in advance for a large operational time window.

Ripples will then be responsible for requesting the corresponding trajectories to the numerical solver, giving it access to the most recent output of the water velocity prediction model. Given the decentralized architecture, the required computations can be performed in onshore machines, with the adequate computational power.

Afterwards, the outputs from the numerical solver are centralized in the Ripples hub. In the end, Ripples communicates with Neptus and shares these results, enabling human operators to define missions that will later be allocated to the AUVs. Recall that Neptus is the console used by the operators in the field to command the mission plans generated from the trajectories that resulted from the numerical solver, and offers a platform for their simulation and plan validation.

Vehicle's low-level control is assured by DUNE, the onboard control software of LSTS toolchain [8].

V. Results and Discussion

This section is divided in two parts. First, we describe mission preparation, including different examples of the different stages of computation. Next, we discuss how to use the value function in this paper scope and how to interpret its data.

A. Mission preparation

The operations scenario in both examples is located in the area of Aveiro, in Portugal, and includes both the coastal lagoon and the coastal waters of the Atlantic Ocean. Using the data from the high-resolution hydrodynamic model in [3], we consider an operational area of approximately 10 km by 13 km in size and a time window of 12 hours, starting at $t_0 = 0.0$ h on the 11th of February, 2021. The data is linearly interpolated with a spatial and temporal resolution of 50 meters and 10 minutes, respectively. The vehicle travels with $u_{\max} = 1$ m/s at the surface, and the minimum depth value is set to 4 meters.

We simulate a round-trip trajectory in the referred area, visiting multiple pollution focuses. We consider a 10 minute interval between trajectories to simulate the duration of the sampling task.

The deployment position is fixed at (40.638º N, 8.768º W) and, in figures 6 and 7, we represent each focus point by a red dot, and the deployment location by a blue dot. We also include an obstacle between the deployment position and the first focus.

With all the aforementioned details, we present below an example of a filled configuration file. The operational area is defined using the coordinates of its lower left and upper right corners - parameters "ll_corner" and "ur_corner" - and the operational time window is limited by the minimum and maximum time instants in UNIX time - parameters "t_min" and "t_max". The minimum acceptable water level and the vehicle's maximum forward speed are set using the "z_min" and "speed" parameters, respectively. The rest of the parameters are easily understandable.

```
{
"ll_corner" = [40.59571, -8.86626],
"ur_corner" = [40.69155, -8.70172],
"t_min": 1613001600,
"t_max": 1613044800,
"space_resolution" = 50,
"temporal_resolution" = 10,
"speed" = 1,
"z_min" = 4,
"target" = [40.638, -8.768],
"focuses" = [[40.620, -8.790],
             [40.640, -8.822],
             [40.650, -8.785]],
"obstacles" = [[[40.6351, -8.7804],
                [40.6308, -8.7740],
                [40.6253, -8.7783],
                [40.6253, -8.7840],
                [40.6327, -8.7840]]]
}
```

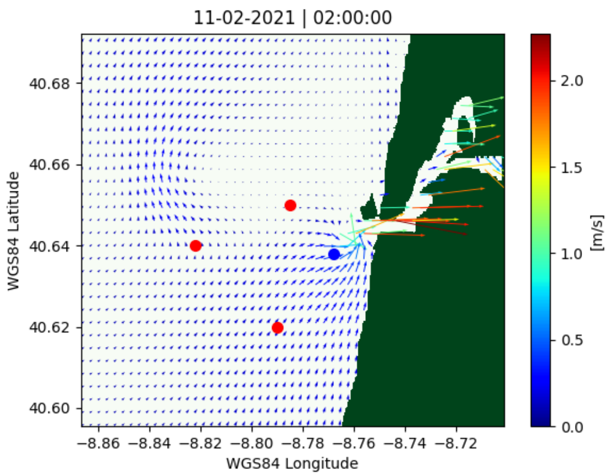
Figure 5: Configuration file

Recall the water velocity flow in the operational area is mainly tidal-driven. Therefore, even before proceeding to the computations and the value function analysis, it can be useful to just simply analyze the hydrodynamic model in the area, and how is the flow in the targets' surroundings. Looking at figure 6, it is interesting to see that the vehicle will face different water velocities depending on the deployment time, mainly due to the tidal effect. Note also the high magnitude of the velocities in some areas.

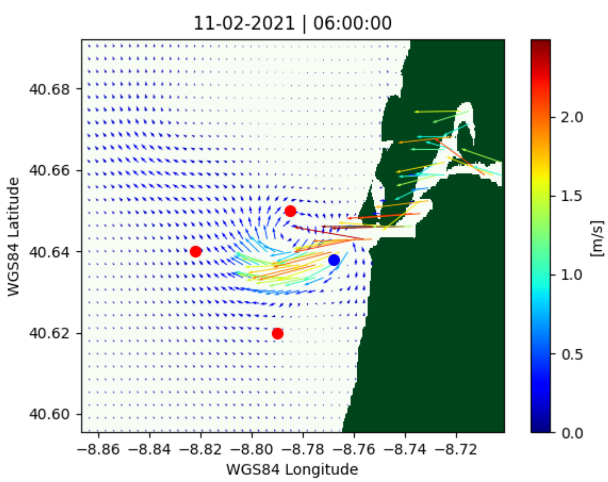
Having the configuration file and the most recent hydrodynamic model data, the software calculates the value functions, and consequently the time-optimal trajectories. We present the result for this scenario in figure 7. The black arrows indicate the water velocity along the trajectory.

Comparing the water velocity with the trajectory's nontrivial shape, we conclude that the latter is extremely influenced by the magnitude and direction of the former, and we can observe how the algorithm of [1] was able to derive a trajectory that takes advantage of the water flow to minimize the cost of travelling.

To end, this information will be provided to Ripples, which will then redirect it to Neptus so that the mission can be validated. We present an example of the Neptus interface in figure 8, including the selected trajectory waypoints. Note the higher concentration of waypoints in the curvilinear parts. This is crucial for the vehicle to execute the planned trajectory as accurately as possible.



(a)



(b)

Figure 6: Water velocity flow in the operational area at a) $t_0 = 2.0$ h and b) $t_0 = 6.0$ h

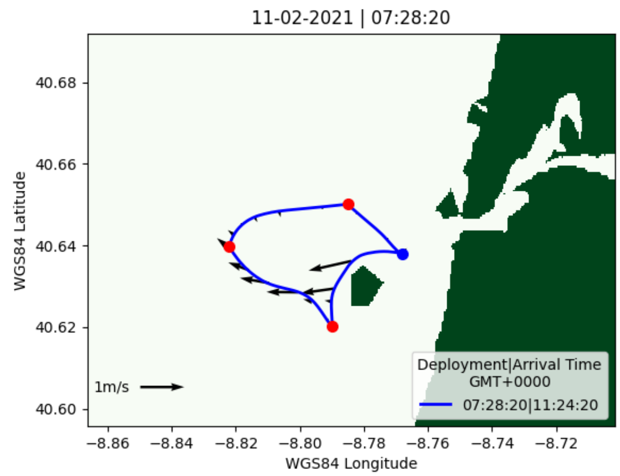


Figure 7: Round trip trajectory visiting multiple pollution centers

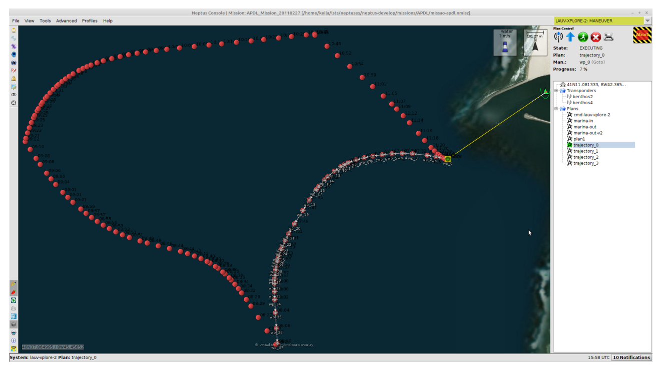


Figure 8: Trajectory on Neptus interface

B. *Using the value function as an operational auxiliary tool*

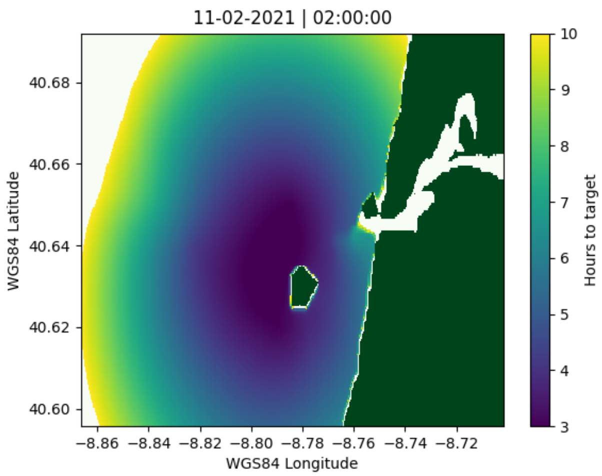
Following the discussion on Section III, we now analyze the value function as an operational management auxiliary tool.

As it is stated in the mentioned section, for each pollution spot, a new value function must be computed, according to the current mathematical problem formulation. However, in a multiple spots scenario, we are interested to evaluate the expected total mission duration. To do so, we proceed as follows. First, recall that in this problem setting $V(\boldsymbol{x}(t), t)$ is the duration of the time-optimal trajectory between $(\boldsymbol{x}(t), t)$ and the target set. We start by choosing an initial time and position, and we compute the time of arrival at the first spot, $t_1 = V(\boldsymbol{x}(t_0), t_0) + t_0$. Then, we sum t_1 with the planned sampling time for that spot. The result of this operation and the coordinates of the spot form the initial conditions for computing the time of arrival to the next spot. By iterating this process through the different spots, finishing at the deployment position, we obtain the mission duration when starting at the chosen initial conditions. We can then repeat the entire process for the set of possible initial conditions and we obtain the mission duration as a function of that same set.

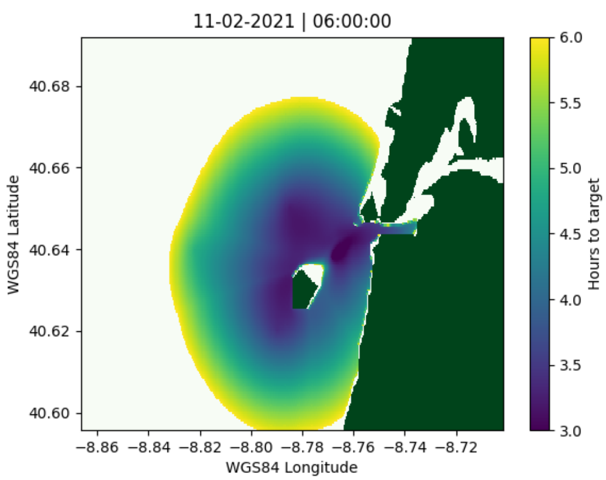
In figure 9 it is possible to see the final result considering the same mission scenario in the first example. Note that in some points no value is plotted. This indicates that the mission cannot be completed within the operational time window when starting from those points on the respective deployment times, $t_0 = 2.0$ h and $t_0 = 6.0$ h. Additionally, it is interesting to conclude that when considering deployment positions near, or inside the lagoon, it is preferable to deploy the vehicle later at $t_0 = 6.0$ h due to the tidal effect. This, of course, highlights the importance of this tool on mission planning.

VI. Conclusion and Future Work

This paper proposes a new software architecture that exploits the advantages of trajectory optimization for AUVs to efficiently and quickly respond to marine point source pollution incidents.



(a)



(b)

Figure 9: Multiple spots mission expected duration departing at a) $t_0 = 2.0$ h and b) $t_0 = 6.0$ h

We integrate time-optimal trajectories computation into a complex, versatile, and modular software toolchain, enabling authorities to plan, control and preview every mission stage. The toolchain offers a framework upon which more specific solutions can be implemented, removing cumbersome implementations of the whole control chain each time one of these solutions comes into play.

We give a practical example of a multiple focuses sampling mission, and we demonstrate how to use the numerical solver output, the value function, during mission planning.

In the future, we plan to test the overall architecture at the simulated area, the coastal lagoon (Ria de Aveiro) in Aveiro, Portugal.

Acknowledgment

This work was partially supported by: Project Ocean-Tech - Sistema de Gestão de Operações com base em Veículos Robóticos Inteligentes para a Exploração do Mar Global a partir de Portugal, POCI-01-0247-FEDER-024508 - funded by FEDER funds through COMPETE2020 - Programa Operacional Competitividade e Internacionalização (POCI) and Project RAMP - Rapid-response to Marine Pollution, FA_06_2017_077 - funded by Direção Geral da Política do Mar (DGPM) through the Fundo Azul program. MCS is funded by national funds (OE), through FCT, I.P., in the scope of the framework contract foreseen in the numbers 4, 5 and 6 of the article 23, of the Decree-Law 57/2016, of August 29, changed by Law 57/2017, of July 19.

References

[1] M. Aguiar, J. B. de Sousa, J. M. Dias, J. E. da Silva, R. Mendes and A. S. Ribeiro, "Optimizing autonomous underwater vehicle routes with the aid of high resolution ocean models," OCEANS 2019 MTS/IEEE SEATTLE, 2019, pp. 1-7, doi: 10.23919/OCEANS40490.2019.8962569.

[2] A. S. Ferreira, J. Pinto, P. Dias and J. B. de Sousa, "The LSTS software toolchain for persistent maritime operations applied through vehicular ad-hoc networks," 2017 International Conference on Unmanned Aircraft Systems (ICUAS), 2017, pp. 609-616, doi: 10.1109/ICUAS.2017.7991471.

[3] Sousa M.C., Ribeiro A.S., Des M., Mendes R., Alvarez I., Gomez-Gesteira M., Dias J.M. (2018) Integrated High-resolution Numerical Model for the NW Iberian Peninsula Coast and Main Estuarine Systems. JOURNAL OF COASTAL RESEARCH, 66-70.

[4] A. Vasilijević, Đ. Nađ, F. Mandić, N. Mišković and Z. Vukić, "Coordinated Navigation of Surface and Underwater Marine Robotic Vehicles for Ocean Sampling and Environmental Monitoring," in IEEE/ASME Transactions on Mechatronics, vol. 22, no. 3, pp. 1174-1184, June 2017, doi: 10.1109/TMECH.2017.2684423.

[5] Bonin-Font, Francisco & Lalucat, Jordi & Oliver, Gabriel & Massot-Campos, Miquel & Font, Eric & Carrasco, Pep Lluis. (2018). Evaluating the impact of sewage discharges on the marine environment with a lightweight AUV. Marine Pollution Bulletin. 135. 714-722. 10.1016/j.marpolbul.2018.07.071.

[6] N. A. Cruz and A. C. Matos, "The MARES AUV, a Modular Autonomous Robot for Environment Sampling," OCEANS 2008, 2008, pp. 1-6, doi: 10.1109/OCEANS.2008.5152096.

[7] Karimanzira, Divas & Jacobi, Marco & Pfützenreuter, Torsten & Rauschenbach, Thomas & Eichhorn, Mike & Taubert, Ralf & Ament, Christoph. (2014). First testing of an AUV mission planning and guidance system for water quality monitoring and fish behavior observation in net cage fish farming. Information Processing in Agriculture. 1. 131-140. 10.1016/j.inpa.2014.12.001.

[8] J. Pinto, P. S. Dias, R. Martins, J. Fortuna, E. Marques and J. Sousa, "The LSTS toolchain for networked vehicle systems," 2013 MTS/IEEE OCEANS - Bergen, 2013, pp. 1-9, doi: 10.1109/OCEANS-Bergen.2013.6608148.