FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO



Tracking and Mapping of the Douro River Plume with an AUV

Diogo Teixeira

Mestrado Integrado em Engenharia Eletrotécnica e de Computadores

Supervisor: Professor João Sousa

External supervisor: Renato Mendes

External supervisor: Joana Fonseca

July 27, 2021

© Diogo Teixeira, 2021

Resumo

Nesta tese, apresentamos três algoritmos para um AUV (*Autonomous Underwater Vehicle*) mapear e seguir de forma autónoma a frente da pluma do Rio Douro (Porto, Portugal). A frente da pluma é caracterizada pelo um elevado gradiente no campo de salinidade, e é seguida em tempo real com recurso a medições de salinidade com um sensor CTD (*Conductivity, Temperature, Depth*).

A descarga do rio flui para o oceano transportando sedimentos, nutrientes e até poluentes que irão interagir e dispersar na água oceânica (com maior salinidade e densidade). Estes elementos propagam-se com a pluma, uma camada de água menos densa, fluindo sobre a água oceânica. Por este motivo, é importante estudar e mapear a dinâmica da pluma, que ajudará a prever o seu efeito nas áreas costeiras.

O estudo de plumas é, geralmente, realizado com a ajuda de navios ou sensores remotos, facilitados por imagens aéreas ou de satélite. Estes métodos são logisticamente difíceis, caros, não possuem as resoluções espaciais e temporais necessárias e tendem a não dar uma imagem sinóptica do fenômeno.

Propomos três estratégias principais para um AUV seguir e mapear de forma autónoma uma pluma dinâmica. O veículo adapta a sua trajetória em tempo-real estimando a orientação da frente. O mesmo atravessa perpendicularmente a frente, de forma a recolher dados em condições ideais para posterior estudo oceanográfico. Esta abordagem foi validada em ambiente de simulação. Implementamos o algoritmo de controlo na LSTS *toolchain* que irá apoiar o lançamento do AUV em campo.

Esta abordagem é inspirada em trabalhos anteriores realizados pelo Laboratório de Sistemas e Tecnologia Subaquática (LSTS) na pluma do rio Douro.

Informações necessárias sobre modelos de plumas e a pluma do rio Douro são apresentadas em preparação para discutir o estado da arte em amostragem adaptativa de diferentes fenómenos marinhos usando veículos autónomos. A formulação do problema, organizado em vários subproblemas, é apresentada, juntamente com uma discussão sobre a abordagem proposta. A realização dos algoritmos de seguimento é documentada, bem como o desenvolvimento de um ambiente de simulação para validar os resultados, que utiliza um modelo numérico que simula com precisão o comportamento da pluma. Seguidamente, é descrita a implementação do algoritmo de controlo na *toolchain* do LSTS. Finalmente, discutimos os resultados e as perspectivas futuras do sistema.

Abstract

In this thesis, we present three algorithms for an AUV (*Autonomous Underwater Vehicle*) to autonomously track and map the front of the Douro river plume (Porto, Portugal). The plume's front is characterized by a steep gradient of the salinity field and is trackable in real-time by measuring salinity with a CTD (*Conductivity, Temperature, Depth*) sensor.

The river discharge flows into the ocean carrying sediments, nutrients, and even pollutants that will interact and disperse in the ocean water (with higher salinity and density). These elements propagate with the plume, a less dense water layer flowing on top of the ocean water. For this reason, it is important to study and map the dynamics of the plume, which will help predict its influence on the coastal areas.

The study of river plumes has been typically done with the help of research vessels or remote sensing products, provided by aerial or satellite imagery. These methods are logistically difficult, expensive, lack the required spatial and temporal resolutions and do not give a synoptic image of the phenomena.

We propose three main strategies for an AUV to autonomously track and map a dynamic plume. The vehicle adapts its trajectory in real-time by estimating the front's orientation and crosses it in a perpendicular direction to retrieve the best possible data for posterior oceanographic study. This approach was validated in a simulation environment. We implemented the control algorithm on the LSTS toolchain that will support the deployment of the AUV in a field setting.

This method is inspired by previous work done by the Laboratório de Sistemas e Tecnologia Subaquática (LSTS) on the Douro river.

Background information on plume models and the Douro river plume is presented in preparation for discussing the state of the art on adaptive sampling of different marine features using autonomous vehicles. The formulation of the problem, organized into several sub-problems, is presented next along with a discussion of the proposed approach. The approach for the tracking algorithms is documented, as well as the development of a simulation environment to validate the results using a numerical model that accurately simulates the plume. Following the implementation of the control algorithm in the LSTS toolchain. Finally, we discuss the results and prospects of the system.

Agradecimentos

Primeiramente, um agradecimento especial aos meus orientadores.

Ao Professor João Sousa, pela orientação constante ao meu trabalho. Por uma revisão crítica e exigente que, sem dúvida, elevou a qualidade desta tese. Agradeço a confiança e a motivação dada ao longo deste semestre.

Ao Renato Mendes, pelos seus conselhos sempre assertivos e generosidade na partilha de conhecimento. Pelo esclarecimento e explicação dos conceitos oceanográficos fulcrais para a compreensão desta temática, totalmente nova para mim. Pelo trabalho realizado na Pluma do Douro, crucial para o desenvolvimento desta tese. Por facultar os dados do modelo da pluma, essenciais para o teste e validação dos algoritmos. Finalmente, pelo tempo que despendeu na revisão e leitura deste documento, nomeadamente do seu conteúdo e do inglês.

À Joana Fonseca, pela total disponibilidade e *feedback* atempado nas diversas fases deste trabalho, nomeadamente pela ajuda no desenvolvimento do algoritmo e do ambiente de simulação, e pela leitura e críticas pertinentes à dissertação.

Ao José Pinto e Alexandre Rocha, pela ajuda na implementação dos algoritmos na LSTS *toolchain*.

Um especial agradecimento ao Miguel Aguiar por partilhar o seu trabalho, crucial para a implementação do algoritmo de controlo no DUNE.

A toda equipa do LSTS que apoiou a realização desta dissertação, permitindo com as suas contribuições e desenvolvimentos passados, a concretização deste projeto.

E um agradecimento especial à minha família, namorada e amigos pelo apoio incondicional.

Diogo Teixeira

Contents

1 Introduction **1**
- 1.1 Context . . . 1
- 1.2 Motivation . . . 1
- 1.3 Objectives . . . 3
- 1.4 Report structure . . . 4

2 Background **5**
- 2.1 Douro river plume . . . 5
  - 2.1.1 Introduction . . . 5
  - 2.1.2 Plume characterization . . . 5
  - 2.1.3 Influence of external parameters on the plume shape . . . 7
  - 2.1.4 DELFT3D Ocean Model . . . 8
- 2.2 Plume measurement and detection . . . 9
- 2.3 Autonomous underwater vehicles . . . 10
  - 2.3.1 Underwater navigation . . . 10
  - 2.3.2 Ocean models . . . 11
  - 2.3.3 Kinematic model . . . 11
- 2.4 Hybrid automata . . . 15

3 State of the art **19**
- 3.1 Introduction to adaptive sampling . . . 19
- 3.2 Sampling of Biological and Chemical features with autonomous vehicles . . . 20
- 3.3 Front tracking and mapping of physical features . . . 23
- 3.4 Performance metrics for oceanographic surveys . . . 26

4 Problem Description **29**
- 4.1 Definitions . . . 29
- 4.2 Static tracking . . . 30
- 4.3 Dynamic tracking . . . 31
- 4.4 Dynamic tracking and full 3-dimensional mapping of the plume . . . 32

5 Approach **35**
- 5.1 AUV trajectory adaptation . . . 35
- 5.2 Front detection and prediction . . . 37
- 5.3 Front tracking at surface level . . . 40
  - 5.3.1 Static tracking . . . 40
  - 5.3.2 Dynamic tracking . . . 47
  - 5.3.3 Measuring plume propagation . . . 47

5.4 Calculation of the plume parameters . . . 49
5.5 Mapping of the plume in 3-dimensional space . . . 53
5.6 Cooperative tracking . . . 55
5.7 External and system parameters . . . 56
5.8 Implementation on the LSTS toolchain . . . 57

6 Results **63**
6.1 Simulation environment . . . 63
6.2 Test plan . . . 64
6.2.1 Performance metrics . . . 67
6.3 Surface tracking of a static plume . . . 69
6.4 Surface tracking of a dynamic plume . . . 70
6.4.1 Effect of a dynamic plume . . . 70
6.4.2 Effect of the AUV velocity . . . 71
6.5 3D tracking . . . 73
6.5.1 Effect of the 3-dimensional trajectory . . . 73
6.5.2 Effect of the distance between surface locations . . . 74
6.6 Effect of different trajectory parameters . . . 75
6.7 Trajectory deviation caused by ocean currents . . . 82
6.8 Plume mapping . . . 87

7 Conclusions and future research **91**

A Test Results **93**
A.1 Surface tracking of a static plume . . . 94
A.2 Surface tracking of a dynamic plume . . . 95
A.3 Tracking of a 3-dimensional plume . . . 97
A.4 Results for different trajectory parameters from set D.1 . . . 99
A.5 Results for different trajectory parameters from set D.2 . . . 104
A.6 Simulation with ocean currents . . . 109

B Plume mapping **111**
B.1 Sampled salinity . . . 111

References **117**

List of Figures

1.1 Schematic of a riverine plume (adapted from [1], Fig. 1). 2
1.2 Light Autonomous Underwater Vehicle (LAUV) designed by LSTS and commercialized by OceanScan-MST (https://www.oceanscan-mst.com/). 2
1.3 Douro river plume. True-color image process from Sentinel 2A satellite data acquired on 15th January 2020. 3

2.1 Imagery obtained by Unmanned aerial vehicle (UAV) video camera on Douro's outfall ([2], Fig. 6). 6
2.2 Model surface salinity in the coastal region of the Douro River. Salinity data from the DELFT3D ocean model. 6
2.3 Salinity contourmap and vertical movement of the vehicle ([2], Fig. 10). 7
2.4 Water leaving radiance at 555 nm in the Douro region. *nLw(555)* composites for low (a) and high (b) discharges calculated from daily scenes when river input is below the 25 percentile and above the 75 percentile, respectively ([3], Fig. 2.3). . 8
2.5 Schematic representation of the Douro estuarine plume under influence of northerly (red), southerly (dark blue), easterly (green) and westerly winds (grey). The locations of secondary freshwater outflows are also marked (Lm-Lima River; Cv-Cávado River; Av - Ave River; RA - Ria de Aveiro Lagoon) ([3], Fig. 5.11). . . . 9
2.6 Representation of the fixed frame *xyz* and body frame of the vehicle. Figure adapted from the LSTS toolchain website [4]. 12
2.7 Vector representation of the vehicle body speed, $v_{v_{bH}}$, water velocity, v_c, and resulting vehicle velocity relative to the ocean floor, v_a, respectively the black, blue and red colored vectors. The vehicle trajectory is depicted with a color gradient representing the vehicle's depth. 15
2.8 Graphical representation of a finite 2-state machine. 16
2.9 Vehicle's behavior (b) according to the control state machine (a). 17

3.1 Process of adaptive sampling (Adapted from Figure 1, [5]). 20
3.2 Desired AUV behavior for a west-ward river plume ([2], Fig. 3). 27

4.1 Representation of the δ-band defined by the salinity values S_{low} and S_{high}. 31

5.1 Schematic representation of the *Zigzag* trajectory and a true-color image process from Sentinel 2A satellite data acquired on 16 th March 2017 (background) of the Douro River Plume with a possible trajectory of the AUV (adapted from [2], Fig. 1). 36
5.2 Schematic representation of the *Alt. Zigzag* (a) and *Lawn Mower* (b) trajectories. 37
5.3 Schematic representation of the typical zigzag trajectory and the first front estimation, γ, after three crossings of the front. 38

5.4 Tracking a static plume with two different methods of computing the front direction. (a) uses n and $n-2$ crossing points and (b) uses n and $n-1$. 39
5.5 Representation of the $\delta - band$ described by the values S_{low} and S_{high} (a). Influence of the D_{min} and D_{max} conditions on the tracking behavior (b). 41
5.6 State machine for the *Zigzag* tracking strategy. 41
5.7 *Zigzag* approach applied to a static plume. 42
5.8 State machine for the *Alternative ZigZag* tracking strategy. 43
5.9 State machine for the *Lawn Mower* tracking strategy. 44
5.10 *Alt. Zigzag* and *Lawn Mower* strategies applied to a static plume. 44
5.11 Sea surface elevation on the Douro estuary between May 7th and 13th in 2016. In red is plotted the mission window on May 9th during the ebb tide, starting at 4:30 and finishing at 10:30. 47
5.12 Algorithm applied to a dynamic plume with an AUV velocity of $1m/s$. Salinity data from May 9th 2016 from 4:30 to 10:30 (ebb tide). 48
5.13 Vehicle's trajectory passing the region of the front two times. *Lawn Mower* strategy must have D_{in} or D_{out} set to zero. 49
5.14 Salinity field of the Douro river plume represented by contour lines (a). Gradient of the salinity field (b). 50
5.15 Initial survey to compute the reference salinity value s in the southern region of the plume. 51
5.16 Initial survey to compute the reference salinity value s in the northern region of the plume. 52
5.17 Survey to compute the reference salinity value s with a vertical profile. 52
5.18 Schematic of the yo-yo trajectory taken by the vehicle. 53
5.19 Vertical profile of the AUV yo-yo trajectory (black dashed line) while crossing the front. The estimated crossing point is the middle point between the surface locations. 53
5.20 Simulation results for a two-vehicle mission. Situation (a) has the vehicle's start positions at the southern and northern edges of the plume. In (b), the AUVs have similar start positions and directions. 55
5.21 Possible shapes of the plume caused by wind direction. 56
5.22 LSTS toolchain structure. 58
5.23 Description of DUNE structure. Image from the Laboratório de Sistemas e Tecnologia Subaquática. 58
5.24 Structure of the Control task interactions. Received and dispatched IMC messages transmitted to/by other DUNE tasks. 59
5.25 Neptus interface describing the vehicle's trajectory and current waypoint (in yellow). 62
5.26 Salinity (in blue) and depth (in red) communicated by the vehicle to Neptus. Data is observed in the Neptus software. 62

6.1 Salinity fields of the plumes representing the three main variations caused by a Southern, Northern, East and/or light wind, respectively, (a), (b) and (c). 64
6.2 Front progression, v_{fp}, (a) and average error, ε_{avg}, (b) results for a surface tracking of a static plume. 69
6.3 Crossings distribution (a,b,c) and efficiency, μ, (d) results for a surface tracking of a static plume. 70
6.4 Front progression,v_{fp}, (a) and average error, ε_{avg}, (b) results for a surface tracking of a dynamic plume. 71
6.5 Crossings distribution for a surface tracking of a dynamic plume. 71
6.6 Average error, ε_{avg}, for different vehicle velocities, v. 72

6.7 Front progression, v_{fp}, (a) and average error, ε_{avg}, (b) results for the 3D tracking and mapping of a dynamic plume. . . . 73
6.8 Crossings distribution for the 3D tracking and mapping of a dynamic plume. . . . 74
6.9 Effect of the distance between surface points, $d_{surface}$, on the average error, ε_{avg}, while the vehicle performs the *Zigzag* trajectory. . . . 75
6.10 Effects of the trajectory angle, α, on the average error, ε_{avg}, and front progression, v_{fp}, with the vehicle performing the *Zigzag* and *Alt. Zigzag* strategies. . . . 76
6.11 Effects of D_{out} on the average error, ε_{avg}, and front progression, v_{fp}, with the vehicle performing the *Alt. Zigzag* and *Lawn Mower* strategies. . . . 76
6.12 Effects of D_{in} on the average error, ε_{avg}, and front progression, v_{fp}, with the vehicle performing the *Lawn Mower* strategy. . . . 77
6.13 Effects of the trajectory angle, α, on the average error, ε_{avg} with the vehicle performing the *Zigzag* and *Alt. Zigzag* strategies on a 3D and dynamic plume. . . . 77
6.14 Effects of the distances, D_{in} and D_{out}, on the average error, ε_{avg} with the vehicle performing the *Alt. Zigzag* and *Lawn Mower* strategies on a 3D and dynamic plume. 78
6.15 Effects of the ratio between the parallel transects to the front outside and inside the plume, $\frac{D_{out}}{D_{in}}$, on the average error, ε_{avg}, with the vehicle performing the *Alt. Zigzag* strategies on a 3D and dynamic plume. . . . 79
6.16 Schematic of one cycle of each trajectory. The front is represented in red. . . . 80
6.17 Front progression, v_{fp}, calculated for the default parameters and compared to the experimental results. . . . 80
6.18 Effect of the angle α in the front progression, v_{fp}. . . . 81
6.19 Effect of D_{out} in the front progression, v_{fp}. . . . 82
6.20 Direction and velocity of the ocean currents at 04:30 May 9th 2016 (*S plume*). The data is shown at three depths: 0 m, 5 m and 10 m. . . . 83
6.21 Direction and velocity of the ocean currents at 13:00 May 15th 2016 (*N plume*) and 21:30 May 2nd 2017 (*LW plume*), respectively, (a,b) and (c,d). The data is shown at surface level. . . . 84
6.22 Effect on the tracking performance in the *N* plume compared to the results where water velocity is assumed to be zero. . . . 85
6.23 Effect on the tracking performance in the *S* plume compared to the results where water velocity is assumed to be zero. . . . 85
6.24 Effect on the tracking performance in the *LW* plume compared to the results where water velocity is assumed to be zero. . . . 86
6.25 Results of tracking a dynamic plume with ocean currents and the path correction method. From simulation set **E**. . . . 87
6.26 Front progression, v_{fp}, (a) and average error, ε_{avg}, (b) results for the 3D tracking and mapping of a dynamic plume with active ocean currents. . . . 87
6.27 Crossings distribution for the 3D tracking and mapping of a dynamic plume with active ocean currents. . . . 88
6.28 Measured surface salinity with the *Zigzag* strategy in the *S plume*. . . . 88
6.29 Sampled salinity data in a front crossing transect. . . . 89
6.30 Sampled salinity data for the entirety of the mission in the *S* plume. . . . 90
6.31 Sampled salinity data for the entirety of the mission in the *N* and *LW* plumes with the *Zigzag* strategy. . . . 90

A.1 Test results for the surface tracking of a static plume. Simulation set **A**. . . . 94
A.2 Test results for the surface tracking of a dynamic plume. Simulation set **B.1**. . . . 95

A.3 Test results for the surface tracking of a dynamic plume for different AUV velocities, v. Simulation set **B.2**. 96
A.4 Test results for tracking a dynamic and 3-dimensional plume. Simulation set **C.1**. 97
A.5 Test results for tracking a 3-dimensional static plume for different distances between surface locations, $D_{surface}$. Simulation set **C.2**. 98
A.6 Results for different values of α using the *Zigzag* strategy. Conditions: (a,b,c) - #00011, (d,e,f) - #00021 and (g,h,i) - #00031. From simulation set **D.1**. 99
A.7 Results for different values of α using the *Alt. Zigzag* strategy. Conditions: (a,b,c) - #00012, (d,e,f) - #00022 and (g,h,i) - #00032. From simulation set **D.1**. 100
A.8 Results for different values of D_{out} using the *Alt. Zigzag* strategy. Conditions: (a,b,c) - #00012, (d,e,f) - #00022 and (g,h,i) - #00032. From simulation set **D.1**. 101
A.9 Results for different values of D_{out} using the *Lawn Mower* strategy. Conditions: (a,b,c) - #00013, (d,e,f) - #00023 and (g,h,i) - #00033. From simulation set **D.1**. 102
A.10 Results for different values of D_{in} using the *Lawn Mower* strategy. Conditions: (a,b,c) - #00013, (d,e,f) - #00023 and (g,h,i) - #00033. From simulation set **D.1**. 103
A.11 Results for different values of α using the *Zigzag* strategy. Conditions: (a,b,c) - #01111, (d,e,f) - #01121 and (g,h,i) - #01131. From simulation set **D.2**. 104
A.12 Results for different values of α using the *Alt. Zigzag* strategy. Conditions: (a,b,c) - #01112, (d,e,f) - #01122 and (g,h,i) - #01132. From simulation set **D.2**. 105
A.13 Results for different values of D_{out} using the *Alt. Zigzag* strategy. Conditions: (a,b,c) - #01112, (d,e,f) - #01122 and (g,h,i) - #01132. From simulation set **D.2**. 106
A.14 Results for different values of D_{out} using the *Lawn Mower* strategy. Conditions: (a,b,c) - #01113, (d,e,f) - #01123 and (g,h,i) - #01133. From simulation set **D.2**. 107
A.15 Results for different values of D_{in} using the *Lawn Mower* strategy. Conditions: (a,b,c) - #01113, (d,e,f) - #01123 and (g,h,i) - #01133. From simulation set **D.2**. 108
A.16 Test results for a 3D tracking of a dynamic plume with ocean currents. From simulation set **E**. 109
A.17 Test results for a 3D tracking of a dynamic plume with ocean currents and path correction method. From simulation set **E**. 110

B.1 Sampled surface salinity for all simulations from set **D** with the path correction method. 112
B.2 Sampled salinity of the *S plume*. 113
B.3 Sampled salinity of the *N plume*. 114
B.4 Sampled salinity of the *LW plume*. 115

List of Tables

2.1 Transition function for a finite 2-state machine. 16

3.1 Categories of adaptive sampling in respect to the type of target feature and mission objective (Adapted from Table 1, [5]). 20

5.1 Strategy parameters. 56
5.2 Mapping parameters required for the tracking and mapping mission on the Douro river plume. 57
5.3 Vehicle parameters for tracking and mapping the Douro river plume. 57

6.1 Description of the parameters that define the test conditions and respective number used to identify them. 65
6.2 Description of all performed simulation sets. 66
6.3 Default parameters for the simulations. 66
6.4 Velocity, v, values taken by the AUV in the test set B.2. 66
6.5 Yo-yo trajectory configurations executed in the test set C.2. 67
6.6 Angle, α, taken by the AUV strategy in the test sets D.1 and D.2. 67
6.7 Distances, D_{out}, taken by the AUV strategy in the test set D.1 and D.2. 67
6.8 Distances, D_{in}, taken by the AUV strategy in the test set D.1 and D.2. 67
6.9 Performance metrics. 68
6.10 Classification of the perpendicular crossings to the front in respect to the error between the plume's front and its prediction, ε. 68
6.11 Proposed parameters for a field mission on the Douro River plume. 78
6.12 Estimation of the total length of mapped front, D_{mapped}, for the default parameters defined in Table 6.3. 82

List of abbreviations and symbols

ASV Autonomous Surface Vehicle. 24

ATBN Augmented Terrain Based Navigation. 11

AUV *Autonomous Underwater Vehicle*. i, iii, ix, xii, 1, 2, 3, 7, 10, 11, 13, 14, 15, 19, 20, 21, 23, 24, 26, 27, 32, 33, 35, 36, 40, 41, 47, 48, 50, 51, 52, 53, 55, 56, 57, 59, 61, 64, 65, 66, 72, 82, 86, 91, 92, 96

BBP Behavior-based planning. 22

CTD *Conductivity, Temperature, Depth*. i, iii, 9, 10, 49, 57, 91

DUNE Unified Navigation Environment. x, 57, 58, 59, 63, 67, 92

GPS Global Positioning System. 10, 11

IMC Inter-Module Communication. x, 57, 58, 59

IMU Inertial Measurement Unit. 10

LAUV Light Autonomous Underwater Vehicle. ix, 2, 4

LSTS Laboratório de Sistemas e Tecnologia Subaquática. i, iii, ix, x, 1, 2, 3, 4, 12, 19, 35, 57, 58, 63, 91, 92

MODIS MODerate Resolution Imaging Spectroradiometer. 9

POSIX Portable Operating System Interface. 58

ROV Remotely Operated Vehicles. 10

STF Subtropical Front. 23

UAV Unmanned aerial vehicle. ix, 6, 25

URV Underwater Robotic Vehicles. 10

USV Unmanned Surface Vehicle. 22, 23, 24

UUV Unmanned Underwater Vehicles. 10

Chapter 1

Introduction

1.1 Context

The main goal of this thesis is to develop algorithms for single or multiple Autonomous Underwater Vehicles (AUVs) to track a river plume autonomously. By expanding on previous work on river plume tracking [2] we will develop and study different methods to successfully track the plume's front. This front is trackable by measuring and processing salinity and temperature variables in real-time to guide the AUV along the edges of the plume.

This work will be developed at the Laboratório de Sistemas e Tecnologia Subaquática – LSTS (Underwater Systems and Technologies Laboratory). The LSTS (https://www.lsts.pt/) is specialized in the design, construction, and deployment of networked air, underwater, and surface vehicles for applications in ocean sciences, security, and defence. The LSTS has expertise in planning and execution control for networked vehicle systems, and in the design of autonomous vehicles and open-source software toolchain for networked vehicle systems. LSTS also has extensive operational expertise with operations taking place in the Atlantic, Arctic, and Pacific oceans, as well as in the Mediterranean and Adriatic seas.

The proponent and supervisor of this thesis is professor João Sousa with Renato Mendes and Joana Fonseca as external supervisors.

1.2 Motivation

A plume is defined as a body of fluid that moves through another one with different properties. In this work, we are interested in the processes by which fresh water (lower salinity) from the Douro river (Porto, Portugal) rich in nutrients, sediments and even pollutants interact with denser ocean water with higher salinity (Figure 1.1). For this reason, it is important to study and map the dynamics of the plume, which will help predict its influence on the coastal areas surrounding the river mouth [6].

It is worth noting that the behavior of the plume is heavily influenced by the meteo and oceanographic setting conditions. The plume will be trackable with an AUV (Figure 1.2) by sampling

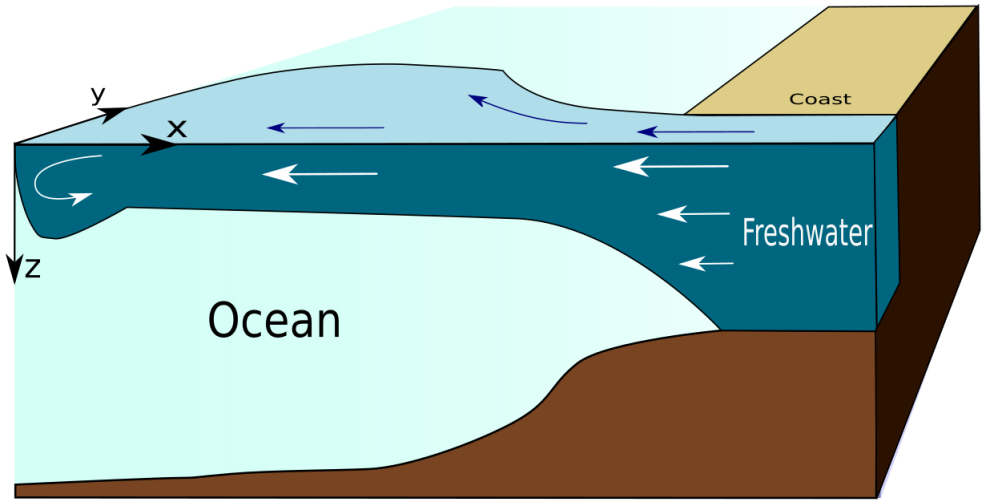


Figure 1.1: Schematic of a riverine plume (adapted from [1], Fig. 1).

salinity values of the water. The ideal conditions for the deployment of the AUV are mainly in spring/summer during ebb tide, where the plume presents a steep salinity gradient. Therefore, we have a 6-hour window to perform the survey.

Typically, the study of river plumes is done by using numerical models and satellite imagery (the front is noticeable since the plume usually has a distinct colour or surface temperature from the ocean water as shown in Figure 1.3), by taking water samples or by doing CTD casts from a manned vessel. Most remote sensing techniques used to observe the plume are affected by clouds. Furthermore, daily satellite imagery has a poor temporal and spatial resolution, thus making it difficult to track the evolution of the front in smaller time scales. Additionally, it only provides a 2D characterization of the plume [2]. This is why we need to complement remote sensing data with in-situ data to be collected by an AUV operating autonomously to obtain a precise 4D model of the plume.

We will implement our algorithms with the LSTS software toolchain, an open-source software that supports the deployment of ocean vehicles [7].



Figure 1.2: Light Autonomous Underwater Vehicle (LAUV) designed by LSTS and commercialized by OceanScan-MST (https://www.oceanscan-mst.com/).

Previous implementations of a front tracking system were applied with success by the LSTS. Pinto *et al.* implemented a system with aerial and underwater vehicles to acquire the structure of the Douro river plume [2]. Building upon this work, we intend to create a more efficient AUV trajectory and further reduce the need for human intervention. A more in-depth discussion of the work done by Pinto *et al.* is carried out in section 3.3.

This work is aligned with the developments of the MASCOT projected jointly undertaken by the Norwegian University of Science and Technology and LSTS, with funding provided by the Norwegian Research Council. MASCOT includes a comparative study of the behaviour of Douro and Nidelven (Trondheim) river plumes. This project builds on the work developed by the LSTS on the Douro river plume.

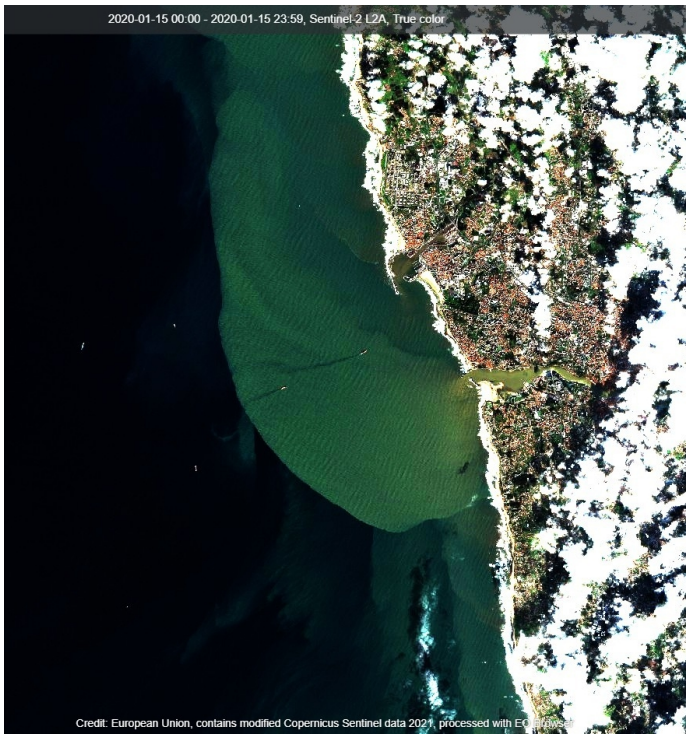


Figure 1.3: Douro river plume. True-color image process from Sentinel 2A satellite data acquired on 15th January 2020.

1.3 Objectives

This thesis has as its main objective to develop and study different strategies to track the plume's front with an AUV. From previous implementations, we want to improve the efficiency and precision of the tracking algorithms. Also, the system should be able to be fully automated by requiring minimal to no human intervention. This will entail the development of an adaptive estimation of the front. The goals are twofold. First, to improve tracking performance. Second, to efficiently sample the plume for detailed oceanographic studies.

Additionally, the front-tracking algorithms should be implemented on the LSTS toolchain, so as to achieve software ready to be implemented on the LAUV and deployed in a future field mission.

1.4 Report structure

This thesis is structured as follows. In Chapter 2 we present a brief overview of relevant oceanographic concepts and methods of detecting and measuring plumes. A brief description of underwater vehicle models and hybrid automata is also provided. In Chapter 3 we discuss related research on adaptive sampling methods for tracking and mapping marine features using autonomous marine vehicles. The problem formulation is done in Chapter 4. Chapter 5 lays out the developed approach for the tracking and mapping algorithms. Chapter 6 presents the simulation results of the tracking system and a consequent discussion. Finally, Chapter 7 gives the final remarks on the dissertation and lays out possible further work to be carried out in continuation to this thesis.

Chapter 2

Background

Relevant background information on oceanographic characteristics of the Douro river plume, marine vehicle dynamics and the mathematical models of hybrid automata is presented in this chapter.

2.1 Douro river plume

2.1.1 Introduction

The Douro river represents one of the main freshwater inputs into the ocean on the Portuguese coast [8]. This water is rich in nutrients, sediments, and even pollutants and will have a great influence on the coastal areas surrounding the river mouth [6], therefore it is important to study and map the interaction between freshwater and the ocean.

River plumes are coastal features created by the discharge of fresh water from rivers [9], in our case the Douro River. A plume is a body of fluid traveling through an ambient substance in which the difference in properties will cause the fluids to not immediately merge (Figure 1.1). The plume will flow from its source, widening along with its travel until it dilutes with the exterior fluid. The propagation of river plumes is difficult to generalize and predict [8]. Mapping and tracking the progression of these plumes with *in situ* monitoring is difficult due to spatial and temporal variability of the external factors that influence the river plume. Therefore, circulation models and remote sensing imagery are used to better understand and predict its behavior [3].

The difference in turbidity caused by the different compositions of the two fluids makes it observable by visible light as shown in Figure 2.1.

2.1.2 Plume characterization

The freshwater outflow near the mouth of the river will usually form a bulge with clockwise rotation since it is located in the northern hemisphere, and then a coastal current is generated that transports water with riverine properties (Figure 1.3). This is the typical behavior of the plume without the influence of external parameters [3].

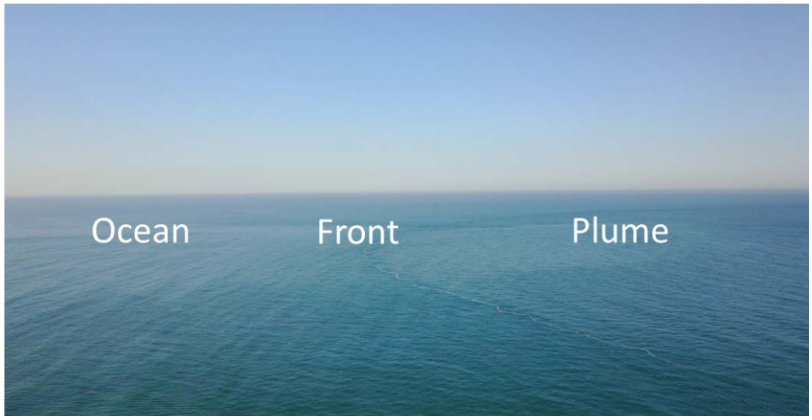


Figure 2.1: Imagery obtained by Unmanned aerial vehicle (UAV) video camera on Douro's outfall ([2], Fig. 6).

Buoyant surface plumes, such as the Douro river plume, can be divided into three areas, source, near and far-field regions. The river mouth is the source area, and the near-field region outlines the expansion of the plume, where the difference between plume and ocean is still significant. The boundary of the near-field is the plume's front where a sharp decrease in contrast in the substances can be observed. The area where the final merge of the waters occurs is the far-field region [3].

Since the main variable used in riverine plume tracking algorithms is salinity, it is imperial to understand the salinity profile of the plume. Salinity is expressed in $\frac{g \text{ salt}}{kg \text{ water}}$. Salinity on the side of the ocean is fairly constant. Then a sharp decrease in salinity is verified at the plume's edge, followed by a continuous decrease of the salinity going into the river mouth. Figure 2.2 shows the surface salinity level where it is possible to see this effect.

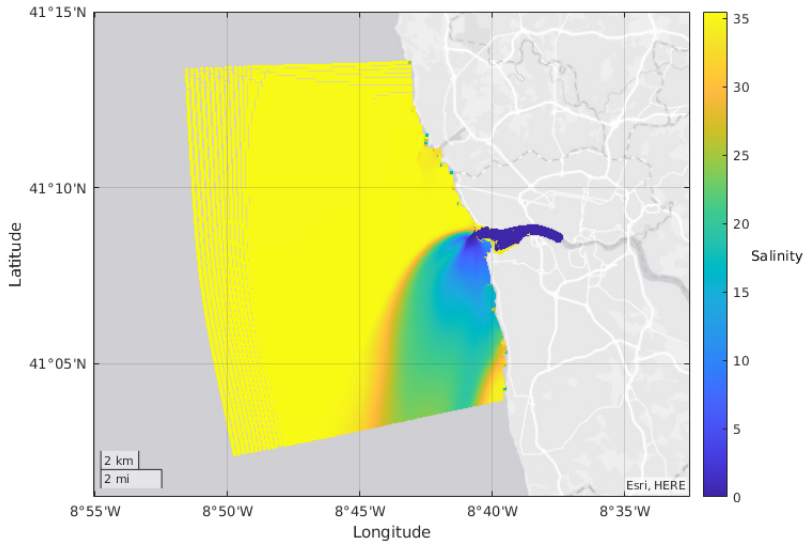


Figure 2.2: Model surface salinity in the coastal region of the Douro River. Salinity data from the DELFT3D ocean model.

However, the river plume is also specified by its depth, Figure 2.3 from a previous tracking study with an AUV on the Douro estuary [2] shows a plume depth in the plotted region of 2 meters. Both the surface and the vertical profile show a clear, sharp gradient between the ambient water and the riverine water, this is classified as the plume's front.

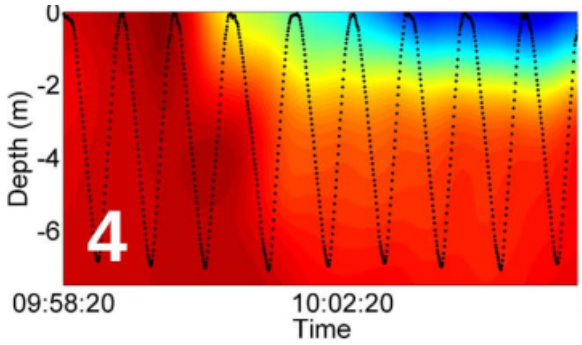


Figure 2.3: Salinity contourmap and vertical movement of the vehicle ([2], Fig. 10).

To perform the best possible characterization of the plume with an AUV (as is exemplified in Figure 2.3), the AUV's path must be perpendicular to the plume's front. Since field transects not perpendicular to along-current direction may cause a distortion in the identification of the plume characteristics such as spreading width, length of hydrodynamic mixing zone and minimum dilution. For this reason, the algorithms to be developed in this work are intended to retrieve the maximum number of possible transects perpendicular to the front. This will provide the optimal sampled data for posterior oceanographic study.

2.1.3 Influence of external parameters on the plume shape

The plume shape and movement are influenced by external parameters. Thus, it is possible to roughly predict its behavior by interpreting forcing factors. The Douro river plume is mainly influenced by the river discharge, wind, and tide [8].

The river discharge of the Douro river averages at 505 $m^3 s^{-1}$ and varies between zero and more than 13000 $m^3 s^{-1}$ [10] [8]. This oscillation occurs with an annual cycle, with peak values during winter and high rain episodes [6]. A higher discharge will result in a larger plume [3]. Figure 2.4 presents a satellite measurement of the turbidity-proxy of the water, higher *nLw(555)* values correlate with where the plume is located.

Since the plume is much larger in winter (annual peak discharge), this makes the tracking of the plume in that period unfeasible with an AUV with the current hardware and software capabilities.

The average tidal range is 2.8 m [11] [9]. The influence of high and low tide decreases with high discharge rates, thus in peak discharge, no significant tidal influence is found in the shape of the plume [3]. In Summer/spring (seasons of low discharge where the tracking system should be deployed) the best period to see a good salinity gradient is during ebb tide, i.e. interval between high to low tide. Therefore, the best window for the deployment of the autonomous agents has a duration of about 6 hours, equivalent to half of the semidiurnal period.

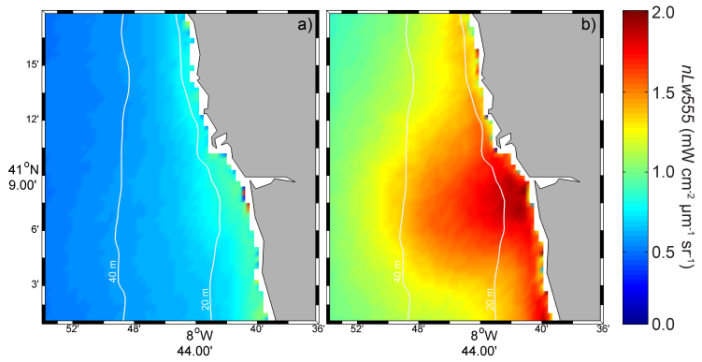


Figure 2.4: Water leaving radiance at 555 nm in the Douro region. *nLw(555)* composites for low (a) and high (b) discharges calculated from daily scenes when river input is below the 25 percentile and above the 75 percentile, respectively ([3], Fig. 2.3).

Wind intensity and direction play a key role in plume propagation and shape [8]. Especially when the freshwater inertia from the river discharge weakens as it distances itself from the source. Northerly winds create an offshore expansion on the plume with a slight inclination southwards and a thinning out of the plume layer as it expands [9] (Figure 2.5 - red). South winds will push the plume against the north coast (Figure 2.5 - blue). Easterly winds will facilitate the creation of a bulge with a northwestern direction and a detachment from the north coast (Figure 2.5 - green). On the other hand, westerly winds will push the plume along the coast in both directions [3] (Figure 2.5 - grey).

Considering the significant influence of the wind on the plume shape this will have a direct effect on the behavior of the front-tracking algorithms, namely the initial position of the vehicle and its direction.

2.1.4 DELFT3D Ocean Model

To test and validate the tracking algorithms a 4D ocean model (longitude, latitude, depth, time) was used to represent the Douro river plume conditions. The DELFT3D model provides multiple oceanographic variables, we will use salinity and ocean current values. These have a temporal resolution of 30 minutes and an average spatial resolution of 0.0004×0.0004 degrees (about 40×40 m). The density of points is not distributed uniformly, the points further away from the river mount have a smaller spatial resolution, as exemplified by the plotted salinity values in Figure 2.2.

The model results follow a previous implementation to the region validated by [12].

This ocean model is the primary tool used in this thesis to validate results, as it provides a reliable simulation of the possible plume conditions. By testing our system in these conditions we can expect a similar behavior to the mission in the field.

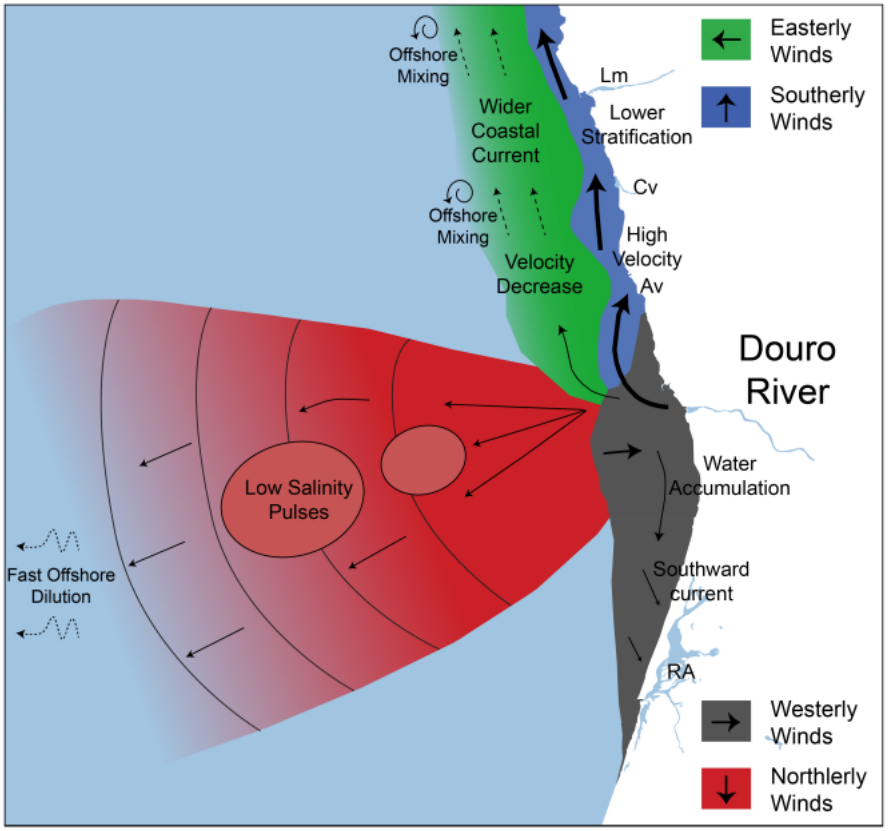


Figure 2.5: Schematic representation of the Douro estuarine plume under influence of northerly (red), southerly (dark blue), easterly (green) and westerly winds (grey). The locations of secondary freshwater outflows are also marked (Lm-Lima River; Cv-Cávado River; Av - Ave River; RA - Ria de Aveiro Lagoon) ([3], Fig. 5.11).

2.2 Plume measurement and detection

Various methods have been typically used to study river plumes. One of these methods is the study of satellite imagery. These sensors distinguish the ocean water from plumes by comparing colour or turbidity. River plumes will present sediments that reduce the water transparency making it especially visible in the green and red wavelengths of the electromagnetic spectrum [13] [3].

One of the most used colour sensors is the MODerate Resolution Imaging Spectroradiometer (MODIS) sensors, operational on the Terra and Aqua satellites. The spatial resolution of these sensors depends on the wavelengths used and can be 1km to 250m [14] [3].

MODIS sensor images have been successfully used to track and study the Douro estuary [8]. However, this measurement has some disadvantages such as having low availability, the satellite completes a full mapping of the Earth within one or two days [14], consequently, it is impossible to measure plume propagation within a small time interval. Furthermore, the plume is mapped in a 2-dimensional space.

The study of river plumes has also been done with the help of research vessels. Typically CTD (*Conductivity, Temperature, Depth*) casts are periodically deployed to obtain vertical profiles of

the plume properties or even moved along with the vehicle effectively mapping the target feature along the path taken by the ship. Another strategy is to place buoys equipped with sensors that continuously retrieve data from that location. These *in-situ* methods are logistically difficult, expensive, lack the required spatial and temporal resolutions, and do not give a synoptic image of the phenomena.

Tracking and mapping plumes with autonomous vehicles addresses these problems by making what can be a laborious and manual operation, with a research vessel, into an automated and much more efficient process with the use of AUVs.

To detect the plume with autonomous vehicles we need access to the variables capable of differentiating the plume from the ambient water. CTD (Conductivity, Temperature, Depth) is an oceanographic instrument used to measure the salinity levels and temperature of the water and the depth of the device. The salinity values are achieved by measuring the electric conductivity of the water which has less resistance with the presence of dissolved salts. To calculate the water temperature a metal is put in contact with the water and measured its resistance by passing a current through the metal. Depth is measured by the effect of the water pressure on a metal membrane that alters its shape, and therefore its resistance, it is possible to correlate this effect with the water pressure. The AUVs used in this thesis will have a CTD aboard sampling these three parameters in real-time.

For surveys in other plumes, different sensors are typically applied. For the detection of oil plumes, sensors that measure the presence of refined oils are used. Chlorophyll sensors detect its concentration in the water and are used, for example, to detect algal blooms. Additionally, a turbidity sensor uses a light receptor to measure the turbidity of the water and can also be a useful tool to detect underwater plumes.

2.3 Autonomous underwater vehicles

Fossen defines an underwater vehicle as a "small vehicle that is capable of propelling itself beneath the water surface as well as on the water's surface. This includes unmanned underwater vehicles (UUV), remotely operated vehicles (ROV), autonomous underwater vehicles (AUV) and underwater robotic vehicles (URV)" [15].

In this section the navigation and kinematic model of underwater vehicles are explored.

2.3.1 Underwater navigation

The AUV must perform a 3-dimensional mapping of the plume, as a result, the vehicle spends most of the mission underwater. Vehicles, when in this situation, are in a GPS-denied environment and ocean currents can alter the AUV trajectory in unpredictable ways. Usually, methods for underwater navigation are based on energy-hungry sensors (Doppler Velocity Loggers and dead reckoning with a compass and Inertial Measurement Unit (IMU)). Such energy consumption reduces mission times. Another alternative for locating underwater vehicles is the use of buoys equipped with GPS that communicate with the vehicle via acoustic signals [16].

Anderson and Smith highlighted the difficulty in the localization of underwater vehicles. The authors developed a method called Augmented Terrain Based Navigation (ATBN) using environmental data, along with depth information to create a terrain map that can aid and be utilized in navigation [17].

In our case, since the vehicle is performing a yo-yo trajectory, it can access a GPS signal when at the surface, this allows performing adjustments to the direction of the vehicle at every cycle of the yo-yo. It is necessary to understand if such methods, like the ones mentioned above, will be necessary. For this, external parameters such as water velocity must be simulated and its influence on the vehicle dynamic tested for the existence of significant alteration of the underwater trajectory. For a successful mission our system does not need a precise position of the vehicle underwater, however, we need to guarantee that the progression of the AUV along its predicted trajectory is not severely impaired as to the point of compromising the mapping and tracking of the plume. Such results are presented in section 6.7 where we see that the influence of the water velocity is manageable and without the need for underwater navigation systems.

2.3.2 Ocean models

To test the capability of the tracking and data sampling of the AUV, a deterministic model was used, the DELFT3D ocean model. Ocean models are now capable of predicting multiple variables (e.g. temperature, salinity, ocean currents, chlorophyll and other biological and physical parameters) with high spatial and temporal resolutions. The use of these models has been proven to increase the effectiveness of marine campaigns [18]. Equally important, their accuracy has also been tested against field results [19]. The authors point to the importance of using ocean models in predicting optimal vehicle movement and increase the effectiveness of scientific data collection. Smith *et al.* studied the error in waypoint navigation of the vehicle using a simulation environment and the field results. Concluding that the model did not produce reliable predictions "along the shelf break region", however, in the majority of other areas the model corresponded to field testing, thus, the areas of more unpredictable currents the deterministic model had less utility on predicting the results.

The use of ocean models in simulations to validate results is a common practice and is exemplified by some of the studies that will be mentioned in the state of the art of tracking and mapping marine features in Chapter 3 [20, 21].

2.3.3 Kinematic model

In order to simulate and validate results, a mathematical model for the movement of the vehicle and ocean currents must be established.

Firstly, frames of reference must be established and be consistent with the existing underwater vehicle literature [22], as such, it is considered an "inertially fixed, non-rotating reference frame xyz", x and y are located in the horizontal plane (orthogonal to gravity) and z has the direction of gravity. We define $z = 0$ at sea level for z to represent the depth of the vehicle.

Additionally, we assign a body-fixed frame to the vehicle with its origin being the center of buoyancy, the frame's axis is aligned with the vehicle's body. Axis 1 has the direction of the vehicle's nose, axis 2 is defined by the plane of the wings and axis 3 is orthogonal to the wings plane.

The coordinates of the vehicle on the inertial frame can be derived by the vectors that express the glider orientation by a rotation matrix $\mathbf{R} \in \mathbb{R}^{3\times3}$. $\mathbf{R}$ is typically parameterized using quaternion parameters or Euler angles. The standard for underwater vehicles is the Euler angle convection (*XYZ*, *yaw*, *pitch*, *roll*) and is the one used in this thesis. The transformation from body to inertial frame is defined by the three angles *yaw* ψ, *pitch* θ and *roll* ϕ. Their directions are defined as illustrated in Figure 2.6 along with the inertial frame.

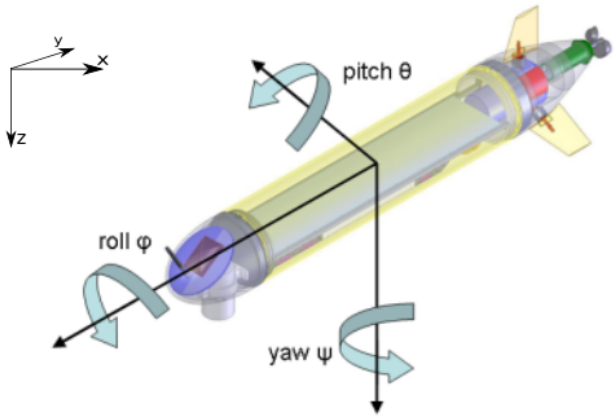


Figure 2.6: Representation of the fixed frame *xyz* and body frame of the vehicle. Figure adapted from the LSTS toolchain website [4].

The Euler angle rotations are represented by the rotational matrices $\mathbf{R}_\psi$, $\mathbf{R}_\theta$ and $\mathbf{R}_\phi$:

$$\begin{aligned}
\mathbf{R}_\psi &= \begin{bmatrix} \cos\psi & -\sin\psi & 0 \\ \sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix}, \\
\mathbf{R}_\theta &= \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix}, \\
\mathbf{R}_\phi &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi & \cos\phi \end{bmatrix}.
\end{aligned} \tag{2.1}$$

The rotation from body to inertial frame is then defined as,

$$\mathbf{R} = \mathbf{R}_\psi \mathbf{R}_\theta \mathbf{R}_\phi. \tag{2.2}$$

Huynh *et al.* present a path planning method to minimize AUV energy consumption [23]. This takes into account the spatial disturbances and temporal variability of ocean currents. The simplified 4-D kinematics defined by the authors are used in this thesis to study the water velocity influence on the vehicle.

To simplify the model, the following assumptions are made – roll and sway movements are inconsequential. Sway is the movement perpendicular to the vehicle's body (i.e. along the pitch axis), this movement can be caused by the water exerting a force in the vehicle. To account for such effects a hydrodynamic model of the vehicle would need to be implemented. This adds a considerable level of complexity that is not required for our simulation studies. Effectively, in our model, the vehicle is treated as a particle.

The vehicle's state, η, is defined as

$$\eta = (x, y, z, \theta, \psi)^T \tag{2.3}$$

with longitude and latitude coordinates being, respectively, x and y, z is the vertical depth and θ and ψ the pitch and yaw angle. Additionally, v is the absolute body vehicle speed where v_{bH} and v_{bV} are its horizontal and vertical components,

$$\begin{aligned} v_{bH} &= v\cos\theta, \\ v_{bV} &= v\sin\theta, \end{aligned} \tag{2.4}$$

the AUV velocity relative to the ocean floor is

$$v_a = (\dot{x}, \dot{y}, \dot{z})^T. \tag{2.5}$$

The water velocity (ocean current) is expressed by

$$v_c = (v_{c1}, v_{c2}, v_{c3})^T. \tag{2.6}$$

Thus, we can derive the following equations of motion:

$$\begin{aligned} \dot{x} &= v_{bH}\cos\psi + v_{c1}, \\ \dot{y} &= v_{bH}\sin\psi + v_{c2}, \\ \dot{z} &= v_{bV} + v_{c3}. \end{aligned} \tag{2.7}$$

The vertical currents in the model have a large error compared to values in real conditions, and since these are currents with small absolute values they will be considered to be zero, i.e. $v_{c3} = 0$.

Therefore, it is possible to write the equation 2.7 as follows,

$$\begin{aligned} \dot{P_H} &= \begin{pmatrix} \cos\psi & 0 & 1 & 0 \\ 0 & \sin\psi & 0 & 1 \end{pmatrix} \begin{pmatrix} v_{bH} \\ v_{bH} \\ v_{c1} \\ v_{c2} \end{pmatrix}, \\ \dot{z} &= v_{bV}. \end{aligned} \tag{2.8}$$

$P_H = (x,y)^T$ represents the position of the AUV in the horizontal plane.

The discrete model used to compute the results with a time step, T, is

$$P_H(k+1) = P_H(k) + \begin{pmatrix} T\cos\psi & 0 \\ 0 & T\sin\psi \end{pmatrix} \begin{pmatrix} v_{bH} \\ v_{bH} \end{pmatrix} + \begin{pmatrix} T & 0 \\ 0 & T \end{pmatrix} \begin{pmatrix} v_{c1} \\ v_{c2} \end{pmatrix}. \tag{2.9}$$

The state-space representation is then given by:

$$\begin{aligned} P_H(k+1) &= f(\mathbf{P_H}(k)\mathbf{U}(k)), \\ \dot{b_3}(k) &= v_{bV}(k), \end{aligned} \tag{2.10}$$

where,

$$\begin{aligned} f(\dot{\mathbf{P_H}}(k), \mathbf{U}(k)) &= \mathbf{A}(k)\dot{\mathbf{P_H}}(k) + \mathbf{B}(k)\mathbf{U}(k), \\ \mathbf{A} &= \mathbf{I}_{2\times 2}, \\ \mathbf{B}(k) &= \begin{pmatrix} T\cos\psi & 0 & T & 0 \\ 0 & T\sin\psi & 0 & T \end{pmatrix}, \\ \mathbf{U}(k) &= (v_{bH}, v_{bH}, v_{c1}, v_{c2})^T. \end{aligned} \tag{2.11}$$

To account for uncertainties present on the model, the authors add some terms to the equations of motion,

$$\begin{aligned} \dot{P_H}(k+1) &= \mathbf{A}(k)\dot{\mathbf{P_H}}(k) + \mathbf{B}(k)\mathbf{U}(k) + \mathbf{W}\omega(k), \\ \mathbf{W} &= \mathbf{I}_{2\times 2} \in \mathbb{R}^{2\times 2}, \end{aligned} \tag{2.12}$$

where $\omega(k) \in \mathbb{R}^{2\times 1}$ is the uncertainty at instant k. Conversely to the authors, we do not have access to an uncertainty value, however, this same principle is applied to add noise and create a larger influence of the ocean currents on the vehicle allowing us to study such implications.

The equations and concepts described here are used to model the vehicle's dynamics in the simulation environment developed to test and validate the tracking algorithms, results are shown in Chapter 6.

Figure 2.7 shows an example of the current influence through a vector representation. Even though the body speed of the vehicle is constant (black colored vector has always the same direction) the trajectory of the agent does not form a horizontal straight line due to the effect of the water velocity (blue vector). The sum of the two vector results in the AUV velocity relative to the ocean floor (red vector).

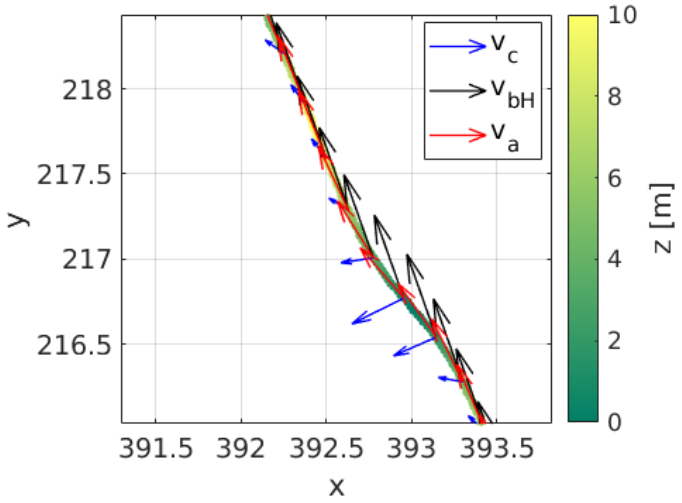


Figure 2.7: Vector representation of the vehicle body speed, $v_{v_{bH}}$, water velocity, v_c, and resulting vehicle velocity relative to the ocean floor, v_a, respectively the black, blue and red colored vectors. The vehicle trajectory is depicted with a color gradient representing the vehicle's depth.

2.4 Hybrid automata

This section presents a mathematical model of hybrid automata, one of the mathematical constructs used in this thesis. A hybrid system displays both a discrete and continuous behavior. Considering a system with finite states, a finite state machine (i.e. a finite automaton) can model the discrete component of the system and a set of differential equations model the continuous component.

A finite automaton M is defined by a finite set of states Q, a finite set of input symbols Σ, also denominated the alphabet of the automaton and a transition function Φ ,

$$M \begin{cases} Q = \{q_0, q_1, \ldots, q_n\} \\ \Sigma = \{a, b, c, \ldots\} \\ \Phi : Q \times \Sigma \rightarrow Q \end{cases}, \tag{2.13}$$

where $q_0 \in Q$ is the initial state.

Figure 2.8 depicts the transition system of a 2-state automaton. In this example, we have a set of states $Q = \{1,2\}$ and a set of inputs $\Sigma = \{a,b\}$. A transition from one state to another is denominated a *jump* or *guard*, the inputs $\{a,b\}$ are guard conditions. The transition function Φ is represented in table 2.1. Note that not all combinations of (q,s) result in a state transition, this is represented by the symbol $\circlearrowleft$ indicating that the current state q dos not change. Therefore, in Figure 2.8 we get single direction arrows from q to $\Phi(q,s)$ with the label s for each pair (q,s) where $\Phi(q,s) \neq \circlearrowleft$.

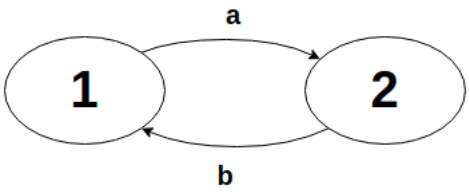


Figure 2.8: Graphical representation of a finite 2-state machine.

$q \in Q$	$s \in \Sigma$	$\Phi(q,s)$
1	a	2
1	b	$\circlearrowleft$
2	a	$\circlearrowleft$
2	b	1

Table 2.1: Transition function for a finite 2-state machine.

An hybrid automaton combines finite state machines and differential equations. To each discrete state there corresponds a differential equation. Considering a state space $x \in \mathbb{R}^n$ and a set of discrete states Q we define the vector field f describing the behavior of the hybrid system,

$$f(q,x) : Q \times \mathbb{R}^n \rightarrow \mathbb{R}^n. \tag{2.14}$$

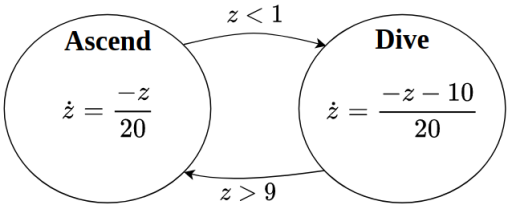
A simple example of such system is an underwater vehicle performing a yo-yo trajectory. This trajectory is characterized by a minimum and maximum depth, $z_{min} = 1$ and $z_{max} = 9$. The vehicle has two states – *Ascend* and *Dive* – and these are controlled by the vehicles depth, z.

The position of the vehicle is described by the following differential equations,

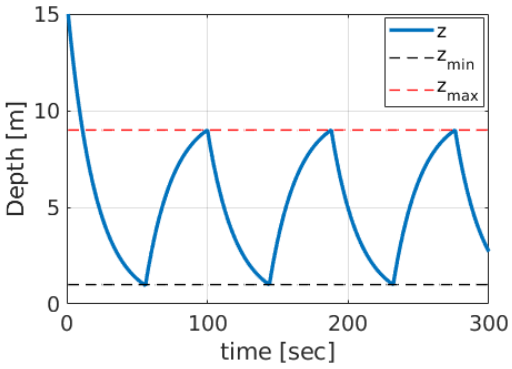
$$\begin{aligned} Ascend : \dot{z} &= \frac{-z}{20}, \\ Dive : \dot{z} &= \frac{-z+10}{20}. \end{aligned} \tag{2.15}$$

The state machine of this system is shown in Figure 2.9a. Defined by the set of states $Q = \{Ascend, Dive\}$ and the guard conditions $\Sigma = \{z < 1, z > 9\}$. Figure 2.9b plots the vehicle's depth, $z(t)$. As a result we have a continuous parameter z that is modeled by a set of discrete

states Q.



(a) State-machine.



(b) Vehicle's position.

Figure 2.9: Vehicle's behavior (b) according to the control state machine (a).

Chapter 3

State of the art

In this section, previous tracking and mapping studies of marine features with autonomous vehicles are discussed to better understand the current state of the art on adaptive sampling of biological/-chemical, and physical plumes.

3.1 Introduction to adaptive sampling

AUVs have now been used in various marine features. As mentioned before, traditional methods for detecting ocean phenomena are logistically difficult, consequently, the use of autonomous systems is required to study complex marine features that are difficult to predict and can be influenced by numerous external conditions (ocean currents, wind, tides, etc.), furthermore, these can have a large temporal variability or could be of very difficult reach with traditional methods. Examples of these situations are an under-ice operation performed on the Arctic [24] and a sampled oil plume located at depths of about 1000m in the ocean [25]. Zhang *et al.* [26] illustrate with multiple examples the utility of autonomous vehicles in "effectively characterizing targeted processes with greater flexibility than what is possible using manned vessels".

These vehicles are programmed to react autonomously to varying environmental conditions allowing them to complete their mission objective in an optimal manner. The autonomy of underwater vehicles can be comprised of three main areas: Energy autonomy, Navigation autonomy and Decision Autonomy [27]. The energy consumption of the AUV is not a major factor since the vehicle is capable of performing the 6-hour mission on the Douro plume. The Navigation autonomy is dealt with by the LSTS toolchain, some of the processes will be further explained in section 5.8. Finally, Decision autonomy is the main contribution of this thesis, and as such we will explore the state of the art in this area.

By collecting *in-situ* data vehicles adapt their trajectories to effectively complete their mission objective without the need for user interference, this process is defined as adaptive sampling. Hwang *et al.* in a review of different AUV adaptive sampling methods divide these into three different categories [5]. The characteristic of the targeted marine feature and the objective of the mission are depicted in Table 3.1. And finally, the composition of the system in a number of

vehicles, these can be classified as a solo, in pair or multi-vehicle approaches. In this chapter, we detail some examples of these methods.

Type of marine feature		
Physical	Biological	Chemical
Pycnocline (Halocline and Thermocline)	Harmful Algal bloom	Natural oil seep/Oil spill
Upwelling	Phytoplankton layer	Waste water
Internal Wave	Zooplankton layer	Hydrothermal vent
Mission objective		
Source localization	Front determination	Tracking and mapping
Gradient Search	Gradient anomaly	Adaptive transect mapping
Bio-mimetic	Isothermal line	Zig-zag feature-based mapping
NMPC scheme	Control-based technique	

Table 3.1: Categories of adaptive sampling in respect to the type of target feature and mission objective (Adapted from Table 1, [5]).

A vehicle in an adaptive sampling mission will modify/optimize its mission plan based on the retrieved data of the vehicle's sensors. The adaptive process of a system can be comprised of two phases: Perceptive and Behavioural. The perceptive phase is where the sensor measurements are sampled and interpreted into meaningful data. Then the Behavioural phase alters the mission plan to best suit the mission objective [5]. This process will run on the vehicle in a closed-loop as depicted in Figure 3.1.

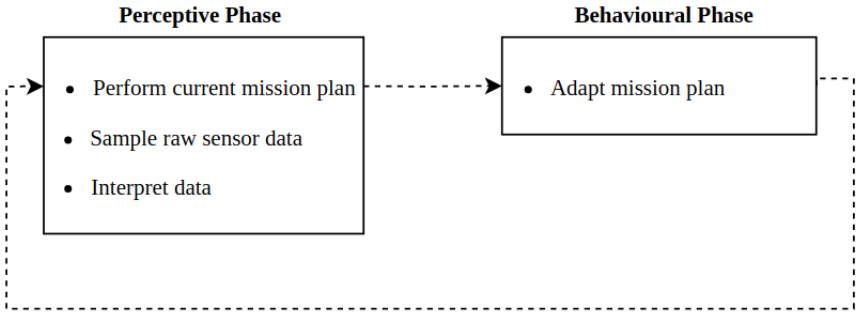


Figure 3.1: Process of adaptive sampling (Adapted from Figure 1, [5]).

3.2 Sampling of Biological and Chemical features with autonomous vehicles

As mentioned before the use of autonomous vehicles is common in biological and chemical features, although this is not the case of the Douro river plume, we will review some of the methods applied to these plumes since these can provide valuable insight on the use of AUVs in tracking and mapping marine features.

The tracking of chemical plumes such as oil spills has been studied on multiple occasions. A multi-agent system for tracking oil spill plumes was simulated on a realistic test environment with success by Fahad *et al.* [20], the system is an estimator-controller that is based on the model of the oil plume. The source of this plume is a single point that causes propagation of the fluid through advection and diffusion, this process can be modeled and used to guide the algorithm [28]. The system developed by Li *et al.* relies on the specific properties of the oil plume and is not suitable for other marine features such as a riverine plumes. Alternatively, a system with a single AUV was developed to track an oil plume without crossing it, by having a sonar sensor it is possible to detect hydrocarbon clouds, when the agent detects the plume it would track its surface delineating the plume shape without needing to cross it [29][30]. In our problem, the crossing of the riverine plume is required and it is not possible to detect it with the use of methods like sonar.

An adaptive triggering method was developed to acquire water samples of a thin phytoplankton layer in Monterey Bay, CA, USA [31]. This method uses an AUV capable of taking water samples. In order to retrieve water samples, the authors needed to develop a triggering method to identify the crossing of the plankton layer. The AUV performs a yo-yo trajectory to cross the layer multiple times. After the first crossing, the peak value is identified and saved as the threshold value. It is assumed that the layer has similar peak values, and thus is possible to identify the subsequent crossings with the defined value. A similar survey on the Douro river plume is made to identify the front value, an analogous assumption is made that this value will define a good isoline to successfully perform the tracking of the front (method detailed in section 5.4).

Zhang *et al.* [25] used a similar approach to their system mentioned above [31] to detect "a horizontally oriented subsurface hydrocarbon plume" caused by the 2010 Deepwater Horizon oil spill in the Gulf of Mexico. This horizontal layer was located between 1100 and 1200m depths. The situation is very similar to the plankton layer and thus the same yo-yo trajectory and adaptive triggering method are used. However, the assumption made for the plankton layer that the same peak value is expected in all crossings is not correct for the hydrocarbon plume, since depending on the direction of the vehicle, moving away or into the plume's source, the peak values will decrease or increase, respectively. At the same time, it is possible to assume that the peak will be at similar depths in the next crossing. The detection of the peak in real-time presents some difficulties. Identifying the peak is done with a basic slope tracking approach, i.e. when the sign of the slope changes a peak is detected. However, this poses the problem of when the peak is detected in real-time the sample should already be taken in that instant and not on the next. Additionally, the authors use a low-pass filter to reduce noise that poses additional delay. To solve this problem, the authors use the yo-yo cycle in their favor. In the first descend dive, the peak value is measured with a delay, and thus the saved depth is not ideal, now the vehicle adjusts the reference depth value for the delay and in the next crossing, the sampling of the water will occur at the correct depth. The identification of the plume's front in our case does not suffer from this problem since there is no change to be made in the sampling process of the vehicle and thus the problem with delays is not significant in this case.

The low passed filter used by the author to reduce noise is achieved by a moving average window,

$$M_{lowpass}(n) = \frac{1}{N}\sum_{i=0}^{N-1} M(n-i), \tag{3.1}$$

where $M_{lowpass}$ is the filtered result, n the current time and M the raw sensor measurements. A moving average window is also used to filter the salinity values in our system.

Source localisation is common when it comes to chemical features since these plumes usually have a single point-source origin (e.g. pipe leakage) [5]. Burian *et al.* [32] implemented a gradient-following method to find hydrothermal vents by using the temperature gradient of the field. A source seeking strategy that estimates a field model using the least-squares method is another example of a system used for source localization [33].

Additionally, Farrel *et al.* present a methodology to find a chemical plume and trace its source with an autonomous vehicle [34]. They implemented a Behavior-based planning (BBP) in which the authors define the different main tasks that the agent needs to perform with 5 behaviors. The deployment of the agent to the mission area and searching for the chemical plume are examples of two implemented tasks. Behavior coordination is achieved by Boolean logic. A similar system is used in the tracking of the Douro plume, behaviors such as traveling to an initial position, performing a salinity profile survey and tracking the front are coordinated with a similar system described by Farrel *et al.*

A system for tracking harmful algal blooms was developed using a set of unmanned surface vehicles (USVs) and satellite information [21]. The algorithm developed by Fonseca *et al.* was studied in more detail to better understand the possible algorithms used to track marine features with multiple autonomous vehicles using adaptive estimation systems.

The algal bloom is estimated as a circumference that the systems of robots will circumnavigate. The center and radius of the shape are estimated, and then a control law is applied to all robots with the goal of converging to the boundary and maintaining an equal distribution throughout the circumference. The shape used in the paper was a circle but the algorithm can be modified to ellipsoid shapes.

n agents are initialized at some initial position $p_i(0)$. The agents have a control input $u_i \in \mathbb{R}^2$, with $p_i = [x_i, y_i]^T$ the vehicle mechanics are described as

$$\dot{p}_i = u_i, \quad i \in I. \tag{3.2}$$

Since the robots must maintain an equal distance between themselves, a angle β_i between consecutive vehicles is calculated as

$$\begin{aligned} \beta_i &= \angle(p_i + 1 - c, p_i - c), \quad i = 1, ..., n-1 \\ \beta_n &= \angle(p_1 - c, p_n - c), \end{aligned} \tag{3.3}$$

$D_i^c(t) = \left\| c(t) - p_i(t) \right\|$ defines the distance of the agent i to the center $c(t)$. The radius $r(t)$ is

estimated and each agent can compute the distance to the boundary,

$$D_i^b(t) = D_i^c(t) - r(t). \tag{3.4}$$

State of circumnavigation is achieved if all agents are moving in the same direction (counter-clockwise) on the boundary of the target and are equally distributed on the circle ($\beta_i = \frac{2\pi}{n}$).

Since in the study case assumes 4 agents the authors use the least-squares method to estimate the circle,

$$\min_{c,r} \sum_{i=1}^{n} (\|p_i - c\| - (r + D_i^b))^2. \tag{3.5}$$

After c and r are updated the direction of each agent to the center is computed as

$$\psi_i(t) = \frac{c(t) - p_i(t)}{\left\|c(t) - p_i(t)\right\|}, \tag{3.6}$$

then it is applied the control law, computed by taking into account the bearing, $\psi_i(t)$ and its perpendicular, $E\psi_i(t)$, being E a rotational matrix (3.7), u_i is defined in 3.8.

$$E = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \tag{3.7}$$

$$u_i = \dot{c} + ((D_i^c - r) - \dot{r})\psi_i + \beta_i D_i^c E \psi_i \tag{3.8}$$

Fonseca *et al.* then prove convergence of the angle to the desired value and the agent's distance to the target boundary.

By extending on the previously discussed paper, a multiple USV and single unmanned aerial vehicle system to circumnavigate a target using adaptive estimation is proposed [35]. As before, the target is approximated as a circle, the additional USV estimates the center and radius of the boundary, the surface vehicles circumnavigate the target using the estimation communicated by the aerial agent. Adaptive estimation of the Douro plume could be used, by defining the target as the plume's front the agents would use this estimation in their control laws. A full shape estimation as is seen in [35] has its problems applied to a riverine plume since the area of the plume would be difficult to estimate as a circle or ellipsoid.

3.3 Front tracking and mapping of physical features

Our problem for the Douro plume is one of tracking and mapping a physical feature. In this section, we study some of the methods used for tracking marine features of this nature.

Ocean isolines of salinity or temperature have been tracked by autonomous marine robots with success. An open-ocean Subtropical Front (STF) in the Pacific was a test subject for a successful tracking algorithm using an AUV [36]. Salinity was the variable tracked in these studies. The salinity profile in ocean fronts can be modeled as a linear function between the boundaries of the

front. Outside of these boundary points, we get a constant value of salinity. Older views would see this profile as a smooth function, with the plume front being located on the point with peak transversal gradient. Conversely, Belkin *et al.* brought their attention to the front edges instead of the front axis. The algorithm uses the values that limit the front's core and not the front axis value, this approach is denominated "front core tracking". The vehicle will follow the front axis by remaining inside the front. This strategy has two main advantages, by traveling less distance since the front is not crossed completely like a typically zigzag approach, the front progression is faster. Furthermore, the risk of the system losing the front is smaller since we maintain the AUV always inside of it. Since we want to fully map the salinity profile, it is not possible to maintain the AUV inside the front, furthermore, the Douro riverine plume has a smaller front width making it not possible to use a similar implementation.

A front-following algorithm was used for tracking a coastal upwelling front in Monterey Bay, CA [37]. The front is distinguished by sampling temperature levels with a single AUV. The algorithm uses a yo-yo vertical trajectory to record temperature values at different depths. By continuously traversing the front it is possible to use the multiple front detection points to understand the front progression with a high resolution. A similar yo-yo trajectory will be used in our implementation to measure salinity values at different depths. Analogous to the described system, multiple passages of the plume's front could be made to determine the velocity of the front.

A front-tracking algorithm for a thermal plume caused by a nuclear power plant outflow was implemented on a USV and successfully proved in field testing [38]. The system uses an adaptive zigzag trajectory based on six different variables, sampled by the agent, that identify the plume. These values are applied to a plume indication function that signals if the vehicle is inside or outside of the plume. The authors conclude that the utilization of this function dependent on six different variables increases the robustness of the tracking system. A similar zigzag like trajectory used in this paper is applied to our system.

Two approaches for static plume tracking and mapping were studied by Cannell and Stilwell [39]. The first approach consists of an AUV performing a fixed path in the study area and then returning a probabilistic distribution of the plume based on the measurements and path of the agent. The second approach is a non-parametric boundary tracking algorithm that first runs a classification phase that identifies the two different regions and proposes a probability distribution of the plume's boundary that then is used to guide the agent along the front. This solution assumes a static phenomenon which is not the case for the Douro riverine plume.

Zigzag like trajectories for tracking marine fronts were validated experimentally with ASVs (Autonomous Surface Vehicles) by Kularatne *et al.* using an indoor setting [40]. The author's tracking system generates an initial path for the vehicle based on a previous characterization of the front using observations from different remote sensing agents. The initial front crossings are determined by the initial zigzag trajectory, after two points, f_n and f_{n-1}, the system estimates the next front crossing location, $\tilde{f}_{n+1}$, using a linear projection. The current destination waypoint, p_n, of the vehicle is adjusted when at the front crossing to limit the travel distance beyond the front. The prediction $\tilde{f}_{n+1}$ is then used to compute the next waypoint, p_{n+1}. The algorithm is presented

as follows:

Algorithm 1: Adaptive Path generation.

if f_n *located before reaching* p_n **then**

$\tilde{f}_{n+1} = f_n + (f_n - f_{n-1}) \times a$;

$p_n = f_n + (f_n - p_{n-1}) \times b$;

$p_{n+1} = \tilde{f}_{n+1} + (\tilde{f}_{n+1} - p_n) \times c$;

else

terminate;

end

a, b and c are constants that define the distance for the projection of $\tilde{f}_{n+1}$, p_n and p_{n+1}. To guarantee that the agent crosses the front in the next crossing, c must have a sufficiently large value, the calculated waypoint will probably overshoot the required distance beyond the front crossing, for this reason, at the next crossing the current waypoint is adjusted. If no front crossing is found, the system is stopped.

A linear projection is used in the Douro plume-tracking system. It is possible to identify some differences, the system described above does not intend to perform perpendicular segments to the front, thus, it is capable of advancing the front at each step by projecting the estimated next crossing of the front. Conversely, in our case, due to the perpendicular crossing requirement, the linear prediction is used to obtain a direction, that then directly influences and controls the vehicle. In fact, the same mathematical process is used in estimating the shape of the front, however, due to the difference in implementation, this leads to different results. Nevertheless, the solution for tracking the Douro plume borrows from this implementation and proves to be a good approach to following river plumes.

The authors tested the algorithm in various test cases, including a varying flow field. It is pointed out the simplicity and robustness of the system to noise measurement. A disadvantage of the algorithm is the incapability of following a circular front witch is a requirement for our problem.

Conversely, a similar approach developed by some of the authors from [40] predicts the front by fitting a cubic spline on the previous front crossing point [41]. Due to the dynamic nature of the front, a precise and more complicated predicting method, that can perform a very accurate front prediction at the time of computation, may not be advantageous. Due to the time variation, there is no guarantee that the prediction will be as accurate when arriving at the next point [40]. Hence, the extra computational effort may not bring any significant improvement in the tracking system. Therefore, the authors used a more simplistic approach (linear projection instead of a cubic spline). This also becomes an apparent consideration when taking into account the propagation of the Douro plume's front, especially in moments of rapid expansion. For this reason, we will also perform a simpler linear projection of the front. If this method proves to be ineffective in field testing, more complex strategies for estimating the plume shape could be studied.

A tracking algorithm for the Douro riverine plume, combining multiple unmanned vehicles (aerial and underwater), was implemented and tested in the field with success [2]. A UAV is

remotely piloted to detect the plume, in addition to multiple AUVs that track the plume's front and map its structure by traversing it multiple times.

Their approach is divided into two phases. Firstly, the surface of the plume and its salinity expression is detected. Then in a second phase, the structure is mapped by the AUVs.

As mentioned before, the difference in the plume turbidity from the ambient water makes it visible by aerial imaging (with visible and/or infrared spectrum). Hence, the aerial drone captures the plume in real-time and the operator is able to distinguish the plume's front. Then the salinity profile is measured by performing a vertical dive with an AUV, this way the salinity expression of the plume is evaluated. Concluded the first phase, a team at sea deploys the AUVs, if the aerial survey was successful, one or more vehicles are positioned close to the front edge, otherwise, the AUVs are deployed at the river mouth and will travel offshore until crossing the plume edge at surface level.

The AUV travels alternating its depth between the surface and a safe depth under the plume, this is called a yo-yo trajectory. This allows the crossing of the plume not only at the surface level but in its full 3D shape, thus making it possible to fully track the plume structure.

When using multiple AUVs they are set on opposite sides of the plume's edge so that the mapping is completed faster.

The AUV starts by moving away from the river at a predetermined angle, α, with the coastline as it continuously ascends and descends in a yo-yo trajectory. Every time the vehicle ascends it checks if it is still inside of the plume, if not, the AUV inverts its direction towards the river mouth. The same process happens when the UAV is inside of the plume, in this case, the inversion is done with the same angle α (see Figure 3.2).

Expanding on this implementation, our system will estimate the front direction and adapt the vehicle's trajectory. The system intends to retrieve the maximum number of perpendicular transects to the front in order to supply the best possible sampling data for posterior oceanographic study. This will result in more accurate and efficient tracking of the front. No remote sensing methods will be used, and the salinity profile of the plume is calculated with the vehicle's first crossing of the front.

3.4 Performance metrics for oceanographic surveys

The development of a front tracking algorithm will have constraints imposed by the vehicle hardware and functional requirements of the survey, metrics should be developed to then quantitatively classify and validate the system. Willcox *et al.* developed a survey error metric for oceanographic surveys with AUVs [42]. One important limitation of autonomous vehicles is energy consumption, which is dependent on the operation area, A, spatial resolution, λ, and total time of the mission, τ. The authors consider a "simple grid survey over a fixed square area", for this case the total traveled distance is $L \approx \frac{A}{2\lambda}$. The speed, V, of the AUV is defined as $V = \frac{A}{2\lambda\tau}$. The total energy

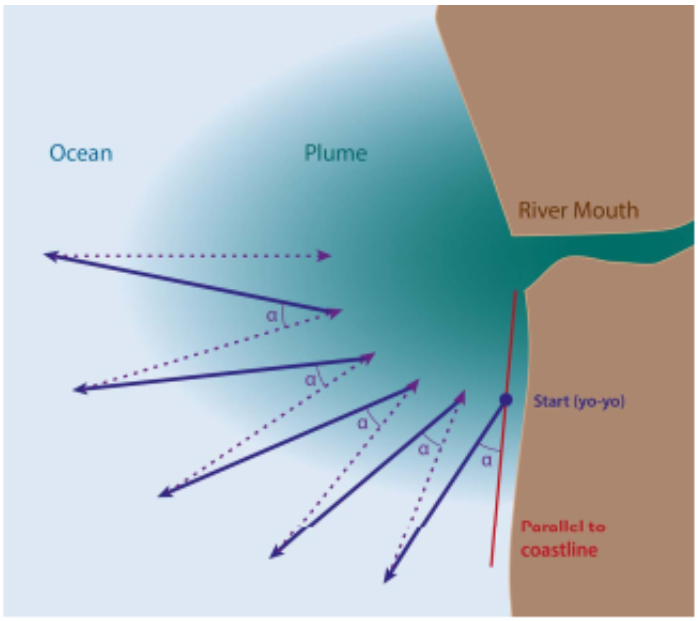


Figure 3.2: Desired AUV behavior for a west-ward river plume ([2], Fig. 3).

consumption is defined as

$$E_{tot} = \left[\frac{\rho C_d S V^3}{2\eta} + H\right]\tau, \tag{3.9}$$

with ρ as water density, D_d as drag coefficient, η as the propulsion efficiency, S as the vehicle surface area that has contact with the water and H as the vehicle hotel load. Dividing E_{tot} by the total distance L and then taking its derivative on the velocity, we get an optimal vehicle velocity,

$$V_{opt} = \left(\frac{H\eta}{\rho C_d S}\right)^{\frac{1}{3}}. \tag{3.10}$$

Thus, minimal energy consumption is achieved when $V = V_{opt}$.

For a yo-yo trajectory with an α angle between straight paths the total travel distance and energy consumption will increase by a factor

$$f = cos(\alpha)^{-3}. \tag{3.11}$$

To classify the system, a combined survey error metric is used that is dependent on the spatial survey error and the temporal survey error. The spacial error considers a field that is independent of time, this error is based on the difference between measurement and real values at some field position. On the other hand, the temporal error takes the error between the true field value at a time t_0 and the sampled value at time $t_0 + t'$ in the same location.

Error metrics such as those developed by Willcox *et al.* are implemented to quantitatively qualify the algorithms designed to track the Douro river plume's front. Since energy consumption is not in focus in this work, the developed metrics will mainly evaluate the performance of the tracking algorithms and the ability to characterize the plume structure in an optimal manner. The

performance metrics are described in section 6.2.1.

Chapter 4

Problem Description

In this chapter, we formally describe the problems presented in this thesis, as well as the respective proposed approaches to be implemented in order to solve them. The approaches are presented in a simplified and conceptual manner. Chapter 5 will describe the approaches and implementation of the solutions in detail. The problems are organized by increasing complexity.

4.1 Definitions

We are interested in the study of the Douro river plume, more specifically in the salinity and temperature maps. Thus, we need to define the plume and vehicle in the context of the tracking and mapping problem.

The river plume is defined as a scalar field *Plume*, evolving with time:

$$Plume : D \subset \mathbb{R}^4 \rightarrow \mathbb{R}^2 \tag{4.1}$$

where D is a closed set. *Plume* has two variables of interest, salinity, S, and temperature, T:

$$Plume(x,y,z,t) = \{T(x,y,z,t), S(x,y,z,t)\}. \tag{4.2}$$

No assumptions on the type of mathematical model that describes these variables are made. The evolution of *Plume* is determined by a set of parameters $Parameters_{Plume} = \{\rho_1, \ldots, \rho_n\}$. The set of parameters can describe wind, tide, salinity profile of the front, velocity of the plume, among others, that directly influence the shape and position of the plume. Some of these parameters are unknown, others are known *a priori* while others are determined when performing the mission.

We do not have direct access to *Plume*, only to point-wise measurements taken by vehicle V_i $(i = 1, \ldots, N)$ along a trajectory $Traj_{V_i} : [t_0, t_f] \rightarrow \mathbb{R}^3$.

The position, p_i, of the vehicle, at time t, is $p_i(t) = [x(t), y(t), z(t)]^T$. The evolution of p_i is governed by the equation of motion:

$$\dot{p}_i(t) = f(p_i(t), u(t), t), \quad u(t) \in U \tag{4.3}$$

where $u(t)$ is the control input of the vehicle's movement and U is a compact set.

The measurements of salinity and temperature taken by V_i at time t are, respectively, $\hat{S}_i(p_i(t),t)$ and $\hat{T}_i(p_i(t),t)$.

The streams of salinity and temperature measurements, taken by vehicle V_i up to time t are respectively

$$StreamS(t) : [t_0,t_f] \rightarrow C, StreamS(t) = \hat{S}_i(Traj_{V_i}(t),t) \tag{4.4}$$

$$StreamT(t) : [t_0,t_f] \rightarrow C, StreamT(t) = \hat{T}_i(Traj_{V_i}(t),t) \tag{4.5}$$

where C is the space of continuous functions $[\mathbb{R} \rightarrow \mathbb{R}]$.

Next, we list a sequence of problems, organized by increasing complexity, to be solved in this work.

4.2 Static tracking

Let us first consider a simplified version of the full tracking and mapping of the Douro river plume. Hence, we have a static plume where the agent movements are restricted to the horizontal plane, this can be thought of as the vehicle always maintaining its position at the surface. Therefore, in this situation, we consider the following assumptions:

Assumption 1: *Plume* is not time-varying and is a continuous field. Thus, we can drop the dependency on time t.

Assumption 2: Vehicle motions are restricted to a given depth z_d.

Assumption 3: The initial position $p_i(t_0)$ provides enough information to start the tracking algorithm.

Problem 1. Given one vehicle V_i, an initial position $p_i(t_0)$ and a set of parameters, $Parameters_{Plume}$, track a $\delta-$band centered on the s isoline of the plume located at depth z_d, $Iso_{(s,z_d)} = \{ (x,y,z) : S((x,y,z_d) = s\}$. More specifically, find a control tracking function $u(t) : StreamS(t) \rightarrow U$.

Observe that $Iso_{(s,z_d)}$ is obtained from the intersection of the s-Isosurface of *Plume* with the horizontal plane $z = z_d$.

In this problem, we are interested in the agent never losing the plume's front, the tracking performance of the proposed solution is measured by the algorithm's ability to follow the plume's front for the s isoline that defines it.

Approach: The proposed method for this problem is the application of a zigzag trajectory, three variations of this trajectory are explored in chapter 5. The vehicle travels in straight lines continuously crossing the plume's front, with inversions in the direction of travel when a certain threshold value of salinity or position is achieved. The last front crossing points will be used to perform a linear prediction of the front (section 5.2), the estimated direction will govern the control function of the vehicle and allow us to follow $Iso_{(s,z_d)}$.

4.3 Dynamic tracking

Following the previous section, we now consider the plume to be time-varying, therefore we have the following assumption in addition to Assumptions 2 and 3.

Assumption 4: *Plume* is a continuous field.

Problem 2. Given one vehicle V_i, an initial position $p_i(t_0)$ and a set of parameters, $Parameters_{Plume}$:

i) track a $\delta-$band centered on the s isoline of the plume located at depth z_d, $Iso_{(s,z_d)} = \{ (x,y,z,t) : \hat{S}(x,y,z_d,t) = s\}$.

ii) draw a map of the plume in this $\delta-$band.

iii) direction of travel, $\psi(p_i(t))$, of the agent at position p_i when crossing the plume front should be perpendicular to $Iso_{(s,z_d)}$ for at least half of the total crossings.

The tracking performance of the algorithm is measured by the difference, error ε, between the perpendicular direction to $Iso_{(s,d)}$ and the vehicle's direction $\psi(p_i(t))$ at some front crossing point, $P_{cross} = [x_c, y_c, z_d]^T$.

$$\varepsilon = \psi(x_c, y_c, z_d, t) - \angle\left[\vec{\nabla} S(x_c, y_c, z_d, t)\right] \tag{4.6}$$

Approach: The same solution, as previously presented for static tracking, is applied for a dynamic plume. By recording the last n front crossing points, $P_{front} = [P_{cross_1}, P_{cross_2}, ..., P_{cross_n}] \in \mathbb{R}^{2\times n}$, we can estimate the direction of the front, γ. This is done with a linear projection between the last and third last detected points, n and $n-2$. All transects entering the plume will have a perpendicular direction to γ.

The $\delta-$band is defined by two salinity threshold values, $[S_{low}, S_{high}]$. When the vehicle measures these values it is triggered to change direction. It then effectively maps and samples data of the plume in the target band defined by S_{low} and S_{high} (Figure 4.1).

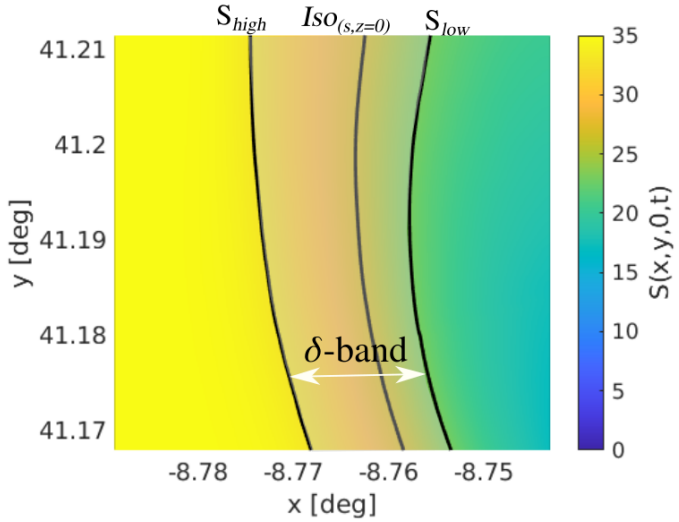


Figure 4.1: Representation of the δ-band defined by the salinity values S_{low} and S_{high}.

An additional requirement can be applied to the current problem:

iv) find the propagation speed of the isoline $Iso_{(s,z_d)}$.

Approach: The proposed approach for measuring the velocity of $Iso_{(s,z_d)}$ is to do multiple perpendicular crossings of the front in the same region. Due to the front's propagation, the position of the crossing points will not be the same at every transect. Hence, by comparing the two positions, P_{cross_n} and $P_{cross_{n-1}}$, where $\hat{S}(P_{cross_n},t_1) = \hat{S}(P_{cross_{n-1}},t_2) = s$ we can estimate the velocity of $Iso_{(s,z_d)}$ in that location,

$$\vec{v_s} = \frac{||\vec{P}_{cross_{n-1}} - \vec{P}_{cross_n}||}{t_2 - t_1}. \tag{4.7}$$

The plume propagation velocity does not influence the algorithm, so this calculation can be done in posterior analysis of the mission data. Not all trajectories developed are capable of performing two transects over the same region. If this is a requirement of the mission, then a certain trajectory must be chosen, this is further explained in section 5.3.3.

4.4 Dynamic tracking and full 3-dimensional mapping of the plume

In the next iteration of the problem, we take into account the 3-dimensional structure of the buoyant plume. Now, only **assumption 4** is true, since the agent will not be restricted to a fixed depth z_s. Furthermore, the initial position of the agent does not provide information to start the tracking algorithm. Thus, it is possible to define the final problem statement.

Problem 3. Given one vehicle V_i, an initial position $p_i(t_0)$ and a set of parameters, $Parameters_{Plume}$:

i) track a $\delta-$band centered on the s isoline of the plume located at depth z_d, $Iso_{(s,z_d)} = \{ (x,y,z,t) : \hat{S}(x,y,z_d,t) = s\}$.

ii) draw a map of the plume in this $\delta-$band.

iii) direction of travel, $\psi(p_i(t))$, of the agent at position p_i when crossing the plume front should be perpendicular to $Iso_{(s,z_d)}$ for half of the total crossings.

iv) Map the isosurface where $S(x,y,z,t) = s$ in the δ-band

Approach: From the initial position, $p_i = [x_i,y_i,z_i]^T$, determined by interpreting the external factors that determine the plume's behavior, an initial survey at depth $z = 0$ is performed by the agent to study the salinity profile of the plume. By sampling the salinity levels, $\hat{S}(x_i,y_i,0,t)$, we compute the maximum variation in salinity, thus defining the front's value as

$$\begin{aligned} t_{max} &= \{t : max\Big(\frac{\partial \hat{S}(x,y,0,t)}{\partial t}\Big)\}, \\ s &= \hat{S}(x,y,z,t_{max}). \end{aligned} \tag{4.8}$$

In addition to the zigzag trajectory (independent of z), we apply a yo-yo trajectory, where the agent continuously alternates between some depth $z = z_{max}$ and the surface. The angle between the trajectory and the surface is a constant value, θ, with $\pi > \theta > 0$. This allows the AUV to cross

the front in the bottom delineation of the plume, thus we can map the plume front surface defined by $S(x,y,z,t) = s$ with a set of sample points given by the condition,

$$\hat{S}(x,y,z,t) = s. \tag{4.9}$$

In addition, we expand the problem above for a multi-vehicle approach. The vehicles do not communicate with one another.

Approach: Two main approaches were devised. The first approach consists in the deployment of the vehicles at the edges of the plume, i.e. one AUV starts at the southern limit of the plume while the second agent starts at the northern limit, and from there, both follow the front in opposite directions. This is the same approach taken in [2], which proved to be successful in field testing on the Douro riverine plume. The second approach is the deployment of the vehicles at a similar position aiming to cross the plume at the same altitude as the river mouth, after crossing the plume both AUVs would continue the tracking algorithm in opposing directions.

Chapter 5

Approach

The approach for tracking the Douro River plume consists mainly of three different strategies for the vehicle's movement that fit the requirements for successfully mapping the plume. Intending to have an adaptive and flexible system, some of the characteristics of the algorithm are user adjustable.

In this chapter, the solutions to the problems described in Chapter 4 are presented. Firstly, the vehicle trajectories are explained, as well as the method for detecting and predicting the front. Furthermore, section 5.3 presents the algorithms for tracking a static and dynamic plumes at the surface level. Then, the approach for tracking and mapping the plume in 3D is introduced, as well as the method for finding the plume's parameters. Additionally, we do an overview of the main parameters that influence the behavior of the system. Finally, the implementation of the LSTS toolchain is described.

5.1 AUV trajectory adaptation

The previous mission on the Douro river plume used a zigzag trajectory [6], as mentioned in Chapter 3.

The system did not implement an adaptive trajectory, as such the first step was to develop an adaptive zigzag trajectory capable of following the front and retrieving valuable scientific measurements. Figure 5.1 shows the ideal trajectory capable of following a typical circular-shaped plume.

Assuming the plume as a continuous and closed shape S, and an initial vehicle position, P_i, inside the plume, we formulate the following algorithm to guarantee that the AUV is capable of progressing the plume's front without losing it.

An initial straight-line segment $\overline{P_i P_{out_1}}$ (P_{out_1} is the location of the first inversion of direction outside the plume) regardless of its direction, is guaranteed to exit S, because of the assumptions on the front. This applies to any segment of type $\overline{P_{in} P_{out}}$, i.e. any segment starting with a point inside the plume, P_{in}, given enough length it will cross the perimeter of S. Additionally, segments coming from the outside of S must also guarantee the crossing of the shape. This can be achieved

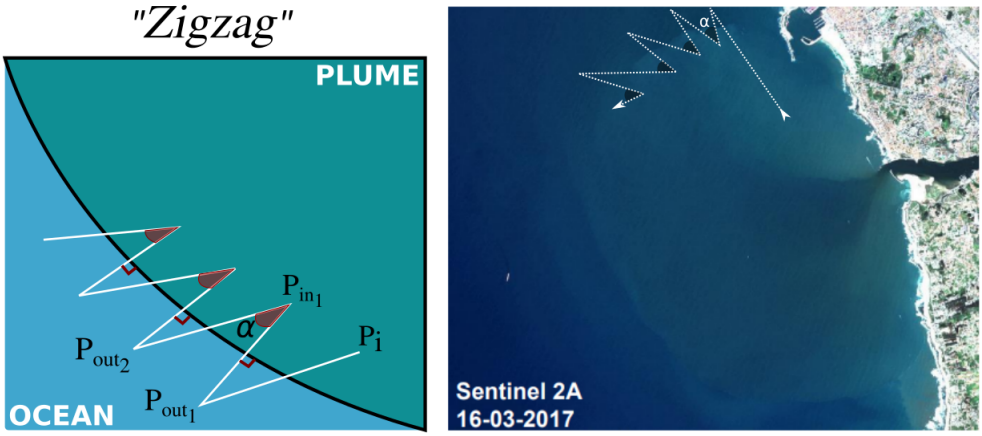


Figure 5.1: Schematic representation of the *Zigzag* trajectory and a true-color image process from Sentinel 2A satellite data acquired on 16 th March 2017 (background) of the Douro River Plume with a possible trajectory of the AUV (adapted from [2], Fig. 1).

by always tracing a segment perpendicular to S. One of these segments is always guaranteed to exist.

With this condition, a trajectory capable of continuously crossing the plume's front is guaranteed. However, a progression along the front is not yet established. For this, the following condition is formulated,

$$\angle P_{out_i} P_{in_i} P_{out_{i+1}} = \alpha. \tag{5.1}$$

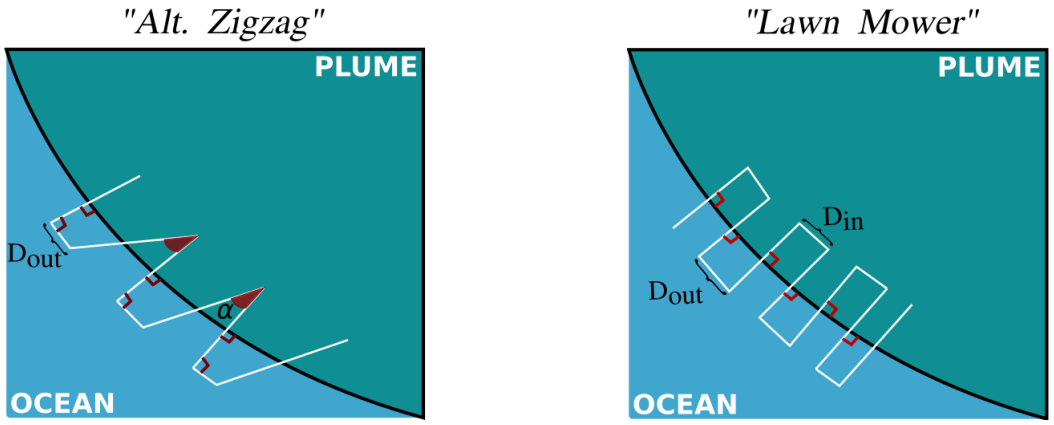
Equation 5.1 defines a direction for all segments exiting the plume, guaranteeing a progression on the plume's front always in the same direction, as seen in figure 5.1.

A successful front progression is achieved if tracing the multiple front crossing points, P_{cross_i}, results in a similar shape to S.

The following sections will detail the methods for replicating the trajectory autonomously, and with limited knowledge on the real shape of the front.

Two different trajectories will also be studied and tested. The first one, mentioned in this thesis as *Alternative Zigzag*, will have an additional segment parallel to the front, when outside the plume, as described by figure 5.2a. This trajectory could result in a faster front progression since it increases the space between front crossing points compared to the *Zigzag* trajectory.

The ideal crossing of the plume should be perpendicular to the front, as this is the situation that yields the ideal conditions for sampling the salinity profile of the plume. For this reason, a trajectory that always traverses the front with perpendicular crossings would be ideal. This is achieved by also performing a parallel transect to the front, when inside the plume. The trajectory is referred as *Lawn Mower*, and it behaves as described in figure 5.2b.



(a) (b)

Figure 5.2: Schematic representation of the *Alt. Zigzag* (a) and *Lawn Mower* (b) trajectories.

5.2 Front detection and prediction

As previously mentioned, the vehicle will not have any information on the shape of the front, this raises two main problems to the theoretical cases described above. Firstly, an initial trajectory must be capable of crossing the front without any previous information on its shape. Secondly, the direction of the front is not trivially known, it must be predicted in order to determine the perpendicular segments to the front.

The first three front crossings are determined by a classical zigzag trajectory with an angle α between all segments of the trajectory. After three crossing points, the trajectory adaptation is initiated.

For the front's prediction, the last and third last front crossing points, P_{cross_n} and $P_{cross_{n-2}} \in \mathbb{R}^2$, are used to perform a linear prediction. The vector, $\vec{f} = [f_1, f_2]^T$, is formed by the two points and represents the estimation of the front's direction, γ. Thus,

$$\begin{aligned} \vec{f} &= \vec{P}_{cross_n} - \vec{P}_{cross_{n-2}}, \\ \gamma &= \angle \vec{f} = \texttt{atan2}(f_2, f_1). \end{aligned} \tag{5.2}$$

The direction γ is used to compute the perpendicular segments. As mentioned in Chapter 4, the error between the true perpendicular direction and the predicted is ε.

Figure 5.3 shows a schematic example of the first zigzag trajectory and the process of predicting the direction of the front after detecting three crossing points.

A more complex method of prediction may give a more accurate representation of the front, however, the front has temporal propagation, thus a very accurate prediction at the time of computation may not be as accurate when the vehicle crosses the next front point. Therefore, a more intricate system may not be worth the extra layer of complexity added, as it was also mentioned in [40].

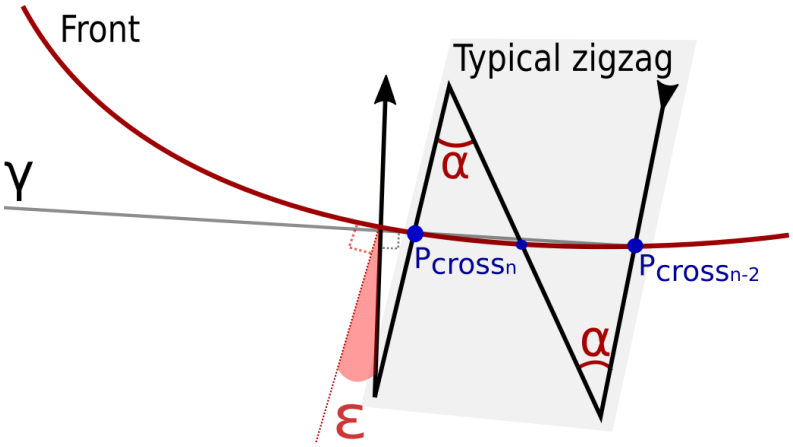


Figure 5.3: Schematic representation of the typical zigzag trajectory and the first front estimation, γ, after three crossings of the front.

If the front crossing points were determined without error, the best prediction of the front results when the two points are closer to each other. As the distance between points decreases, we tend to the front's tangent at the last crossing point, P_{cross}. However, in the real scenario, we do not achieve the location of the points with perfect accuracy, as such very close points with a small error in their location can yield a direction that does not reflect the front's orientation.

With this problem in mind, a minimal distance between front crossing points was added as a condition. D_{cross} is defined as,

$$D_{cross} = ||\vec{f}||, \tag{5.3}$$

if this distance is less than a defined constant D_{MIN_CROSS}, a new vector $\vec{f}$ is computed with a previous P_{cross} location. If it is the case that no point satisfies this condition, the vehicle performs a regular zigzag path, i.e. the new vehicle direction forms an angle α to the previous straight-line path.

The points used for calculating the front direction are $\{n, n-2, n-4, ...\}$. These locations were all determined by the vehicle crossing the front in the same direction. When taking into account the 3-dimensional tracking (explained in section 5.5) the possible region for detecting the plume is not a line, but a band with varying width. If the plume has a very steep gradient, it may be the case that the front is indeed approximated as a line, however, with a more gentle gradient the front detection region can be approximated as a band. Depending on the width of this region, it can result in a significant disparity in the location of the P_{cross} points, when approaching the front from different sides. By performing the calculation of γ with only points from the same direction, we guarantee a maximum equality of conditions, which results in more accurate predictions. This is exemplified in Figure 5.4, where a static plume was tracked by two different methods. The detected crossing points (in white) are not directly above of the front, $Iso_{(s,z=0)}$. We observe that the crossing is triggered before reaching the front. If we use these points indiscriminately, as seen in case (b) the front prediction results in an irregular tracking with a significantly worse performance than case (a).

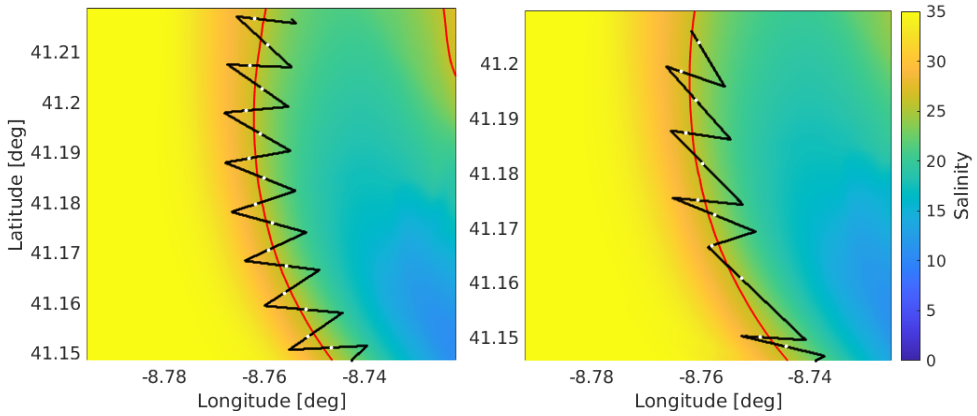


Figure 5.4: Tracking a static plume with two different methods of computing the front direction. (a) uses n and $n-2$ crossing points and (b) uses n and $n-1$.

Therefore, we can write the following algorithm describing the process of predicting the front's orientation:

Algorithm 2: Estimate front's direction.

```
j = 2 ;
Prediction = true;
f⃗ = P_cross[n] - P_cross[n-j] ;
γ = atan2( f_2,f_1) ;
while ||f⃗|| < D_MIN_CROSS do
    j = j + 2;
    if j>=n then
        Prediction = false;
        break
    end
    f⃗ = P_cross[n] - P_cross[n-j] ;
    γ = atan2( f_2,f_1) ;
end
```

Variable *Prediction* is implemented to signal that no prediction of the front's direction is achieved.

5.3 Front tracking at surface level

5.3.1 Static tracking

Let us first consider the problem definition described in problem 1, which indicates a surface traveling vehicle surveying a static front, and thus ignoring time dependencies.

We define the vehicle's position, $p \in \mathbb{R}^2$, as

$$p(t) = [x(t), y(t)]^T. \tag{5.4}$$

The agent is assumed to have a constant velocity v. The control input is the direction of travel $\psi(t)$. The dynamics of the agent are defined in (5.5).

$$\dot{p}(t) = \begin{bmatrix} v \cdot \cos \psi(t) \\ v \cdot \sin \psi(t) \end{bmatrix}. \tag{5.5}$$

The vehicle has access to the values of salinity, $\hat{S}$, and temperature, $\hat{T}$, in real-time. However, only the salinity data is used to control the vehicle's movement. Nonetheless, both values are sampled for posterior analysis.

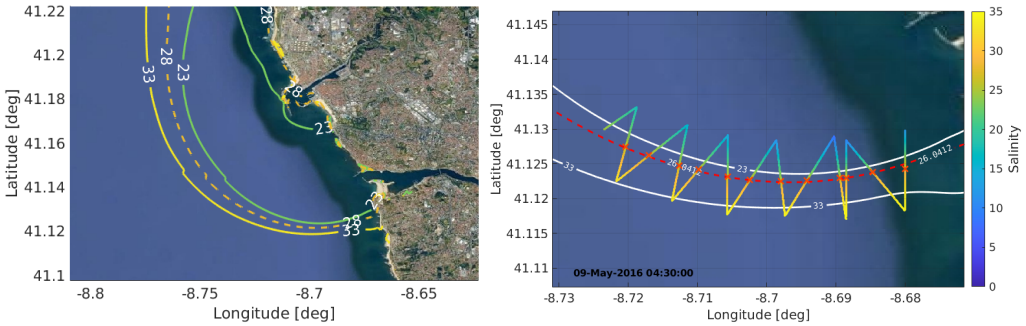
Starting with the *Zigzag* implementation, the vehicle's movement can be modeled as a hybrid system with 2 states $Q = \{0, 1\}$, $q = 1$ when moving away from the plume and $q = 0$ moving into the plume. The direction of travel is determined by the state of the AUV,

$$\psi(t) = \begin{cases} \beta(t), & \text{if } q = 1 \\ \sigma(t), & \text{if } q = 0 \end{cases}. \tag{5.6}$$

The agent changes state when hits certain salinity threshold values, S_{low} and S_{high} , that define the δ-band. An additional condition of a minimal and maximal distance, D_{min} and D_{max} after crossing the front is also considered. The travelled distance after a front detection is D_{front}.

Figure 5.5a shows the salinity threshold limits, as we can see, these do not maintain a regular spacing along the front. In this example, the south segment of the plume has a steeper gradient, therefore, with minimal distance the vehicle can cover the δ-band. Conversely, in the north segment of the plume the gradient is more gentle, and thus creates a wider band. These two parameters are not ideal to govern the algorithm. By taking into consideration the distance that the vehicle travels beyond the front we can perform a more regular tracking of the plume.

In addition, these values can be modified, thus achieving different types of mission configurations. In the case of the Douro river plume, we want at least 500 m of travel after the front and a maximum of 700 m. The application of these conditions can be seen in Figure 5.5b where the vehicle, even though it has surpassed the low threshold $S_{low} = 23$, still samples the plume for the minimum required distance.



(a) $S_{low} = 23$ and $S_{high} = 33$ (b) $D_{min} = 500$ m and $D_{max} = 700$ m

Figure 5.5: Representation of the $\delta - band$ described by the values S_{low} and S_{high} (a). Influence of the D_{min} and D_{max} conditions on the tracking behavior (b).

The two guard conditions that define the state machine are depicted in figure 5.6. Note that the plume's front salinity value, s, will be a number between the threshold values, the computation of s is explained in section 5.4.

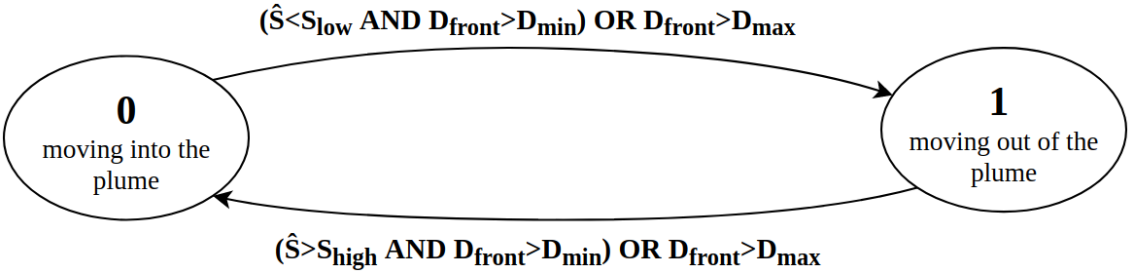


Figure 5.6: State machine for the *Zigzag* tracking strategy.

The initial position and direction of the agent, $p(0) = [x(0), y(0)]^T$ and $\psi(0)$, is set to be inside the plume and near the plume's front. The agent moves in straight lines, thus the control input, $\psi(t)$, is only updated at state transitions.

When the agent inverts its direction in the ocean water, i.e. $q = 0 \rightarrow q = 1$, the angle, α, between the previous and the new direction is constant, thus the direction of travel, β, is computed as

$$\beta(t) = \sigma(t) - \pi - \alpha. \tag{5.7}$$

Conversely, if $q = 1 \rightarrow q = 0$, the new direction of the agent, $\sigma(t)$, will be perpendicular to the plume's front prediction γ. Therefore, the direction of the AUV is

$$\sigma(t) = \gamma - \frac{\pi}{2}. \tag{5.8}$$

A test result is shown in figure 5.7, the black line indicates the path taken by the agent and the orange contour line indicates the isoline $Iso_{(s,z=0)} = \{ (x, y, 0, t_f) : S(x, y, 0, t_f) = 26.1 \}$.

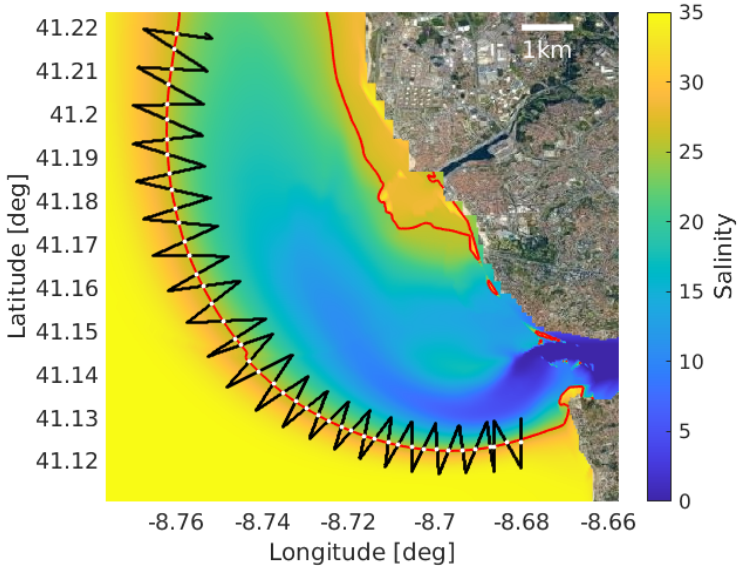


Figure 5.7: *Zigzag* approach applied to a static plume.

In this example, the progression along the front has a clockwise direction. This is achieved by having a positive value for α. If an anticlockwise direction is necessary, α must be set to a negative value.

The control algorithm of the *Zigzag* strategy looks as follows:

Algorithm 3: Control algorithm for the *Zigzag* strategy.

```
q = 1;
ψ = ψ_i ;
while true do
    if q=0 and ((Ŝ < S_low and D_front>D_min ) or D_front<D_max) then
        q = 1 ;
        ψ = ψ - π - α ;
    end
    if q=1 and ((Ŝ > S_high and D_front>D_min ) or D_front<D_max) then
        q = 0 ;
        if n<3 or prediction=false then
            ψ = ψ - π - α ;
        else
            ψ = γ - π/2 ;
        end
    end
end
```

Direction γ is computed as described in algorithm 2 and n is the total number of crossing points. Thus, if three crossing points are not yet detected or no prediction for the front is available, the trajectory is a simple zigzag, i.e. the condition $\psi = \psi - \pi - \alpha$ is performed even when the

vehicle is moving into the plume ($q = 0$).

In the *Alt. zigzag* strategy the movements of the agent are modeled with 3 states, $Q = \{0,1,2\}$, $q = 1$ when moving away from the plume, $q = 0$ moving into the plume and $q = 2$ when moving parallel to the front prediction. The direction of travel is then defined as,

$$\psi_{alt_zigzag}(t) = \begin{cases} \beta(t), & \text{if } q = 1 \\ \sigma(t), & \text{if } q = 0 \\ \sigma(t) + \frac{\pi}{2}, & \text{if } q = 2 \end{cases}. \tag{5.9}$$

Figure 5.8 illustrates the state machine for this strategy. Where d is the distance travelled in each state and D_{out} defines the minimum length of the parallel segment in meters.

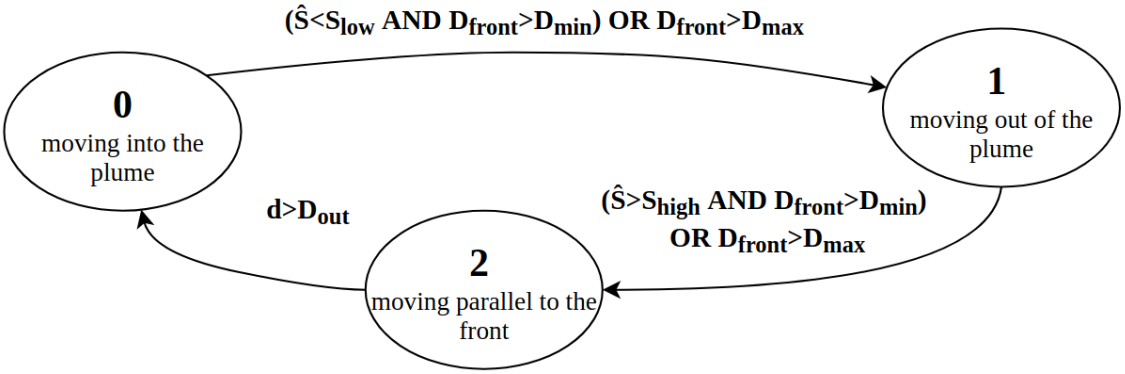


Figure 5.8: State machine for the *Alternative ZigZag* tracking strategy.

With the *"Lawn Mower"* strategy the movements of the agent are modeled with 4 states, $Q = \{0,1,2,3\}$, $q = 1$ when moving away from the plume, $q = 0$ moving into the plume, $q = 2$ when outside the plume moving parallel to the front prediction and $q = 3$ when inside the plume moving parallel to the front prediction. The direction of travel is then defined as

$$\psi_{LawnMower}(t) = \begin{cases} \sigma(t) - \pi, & \text{if } q = 1 \\ \sigma(t), & \text{if } q = 0 \\ \sigma(t) + \frac{\pi}{2}, & \text{if } q = 2 \wedge q = 3 \end{cases}. \tag{5.10}$$

Figure 5.9 illustrates the state machine for this strategy, where D_{in} defines the minimum length of the segment when moving inside the plume.

Figure 5.10 shows the two approaches applied to a static plume.

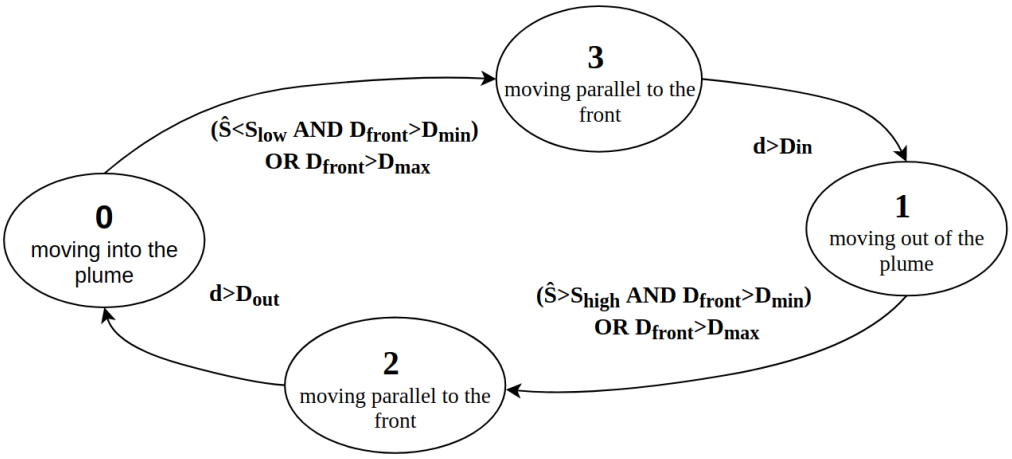
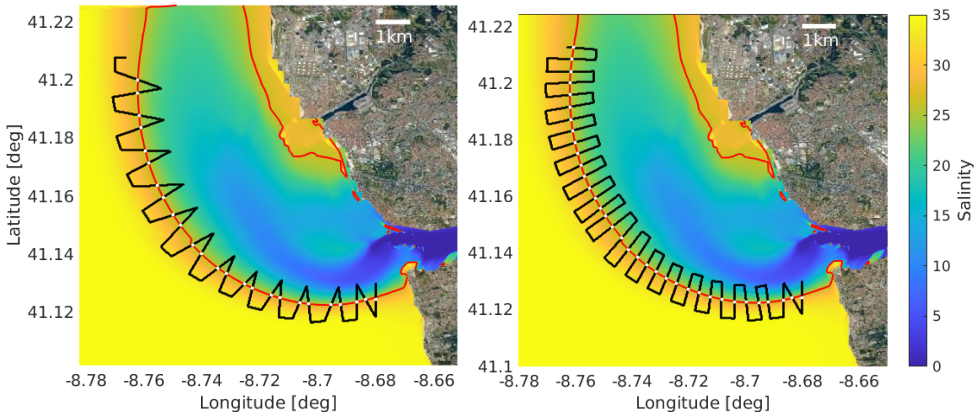


Figure 5.9: State machine for the *Lawn Mower* tracking strategy.



(a) *Alt. Zigzag* (b) *Lawn Mower*

Figure 5.10: *Alt. Zigzag* and *Lawn Mower* strategies applied to a static plume.

The control algorithms for the *Alt. Zigzag* and *Lawn Mower* strategies, respectively, algorithms 4 and 5, are presented as follows:

Algorithm 4: Control algorithm for the *Alt. Zigzag* strategy.

```
q = 1;
ψ = ψ_i ;
while true do
    if q=0 and ((Ŝ < S_low and D_front>D_min ) or D_front<D_max) then
        q = 1 ;
        ψ = ψ - π - α ;
    end
    if q=1 and ((Ŝ > S_high and D_front>D_min ) or D_front<D_max) then
        if n<3 or prediction=false then
            q = 0 ;
            ψ = ψ - π - α ;
        else
            q = 2 ;
            ψ = γ ;
        end
    end
    if q=2 and d>D_out then
        q = 0 ;
        ψ = γ - π/2 ;
    end
end
```

Algorithm 5: Control algorithm for the *Lawn Mower* strategy.

```
q = 1;
ψ = ψ_i ;
while true do
    if q=0 and ((Ŝ < S_low and D_front>D_min ) or D_front<D_max) then
        if n<3 or prediction=false then
            q = 1 ;
            ψ = ψ - π - α ;
        else
            q = 3 ;
            ψ = γ ;
        end
    end
    if q=1 and ((Ŝ > S_high and D_front>D_min ) or D_front<D_max) then
        if n<3 or prediction=false then
            q = 0 ;
            ψ = ψ - π - α ;
        else
            q = 2 ;
            ψ = γ ;
        end
    end
    if q=2 and d>D_out then
        q = 0 ;
        ψ = γ - π/2 ;
    end
    if q=3 and d>D_in then
        q = 1 ;
        ψ = γ + π/2 ;
    end
end
```

5.3.2 Dynamic tracking

The presented approach is now applied to a plume propagating through time (Problem 2 in section 4.3).

With the introduction of a temporal dimension, the plume's front presents a propagation speed. Thus, the vehicle's velocity becomes an important factor in the success of the tracking mission. If the velocity of the front is larger than the vehicle speed, it is impossible to track the front. The influence of the agent's velocity is studied in Chapter 6. Note that the typical velocity of the front does not exceed values of 0.5 m/s and the AUV velocity is 1 m/s.

Other temporal constraints must also be taken into account, namely, the start time of the mission and its duration. As mentioned in section 2.1.3, the mission is performed during the ebb phase and has a total duration of 6 hours. In fact, we initiate the mission 30 min after high tide where the ideal conditions for tide and currents occur.

For example, on May 9th 2016 the conditions for a mission start occur at 4:30. Figure 5.11 shows the tide evolution in May between the 7th and the 13th and plots the 6-hour mission window (in red) starting at 4:30 and ending at 10:30.

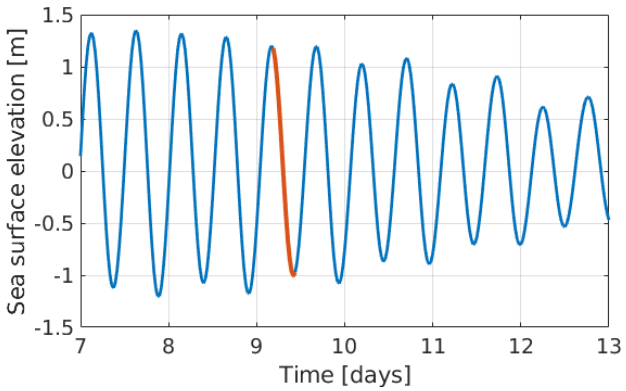


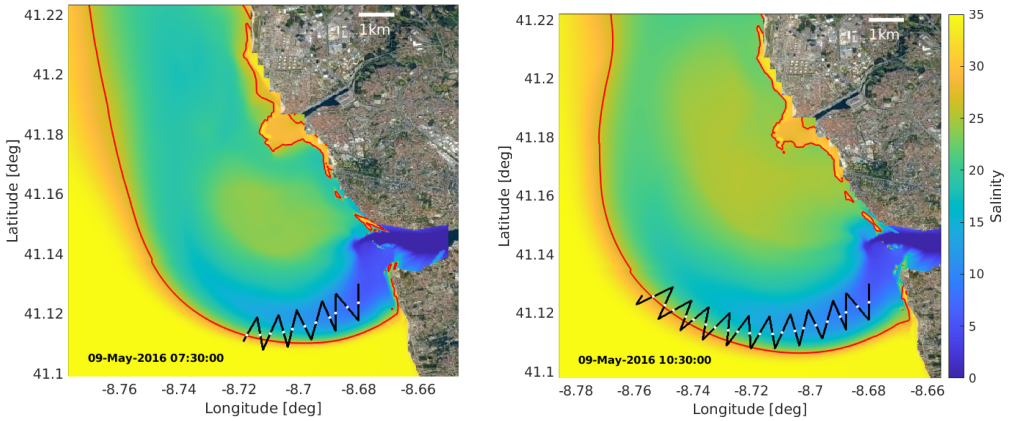
Figure 5.11: Sea surface elevation on the Douro estuary between May 7th and 13th in 2016. In red is plotted the mission window on May 9th during the ebb tide, starting at 4:30 and finishing at 10:30.

Figure 5.12 shows the *Zigzag* approach in the same condition as in figure 5.7 but now applied to a dynamic plume. The algorithm is still capable of following the front. Results for the other two approaches and different plume shapes are shown in Chapter 6.

5.3.3 Measuring plume propagation

The front's velocity could be calculated and analysed by passing the vehicle multiple times through the same transect. This can provide a good estimate of the plume's propagation by identifying the position of each crossing of the front. The true velocity of the front can only be accurately measured in the direction of the plume's gradient (i.e. perpendicular to field lines).

From the three strategies developed, only the *Lawn Mower* path is able to perform transect perpendiculars to field lines in both directions. In order to achieve a configuration capable of



(a) $t = 3h$ (b) $t = 6h$

Figure 5.12: Algorithm applied to a dynamic plume with an AUV velocity of $1m/s$. Salinity data from May 9th 2016 from 4:30 to 10:30 (ebb tide).

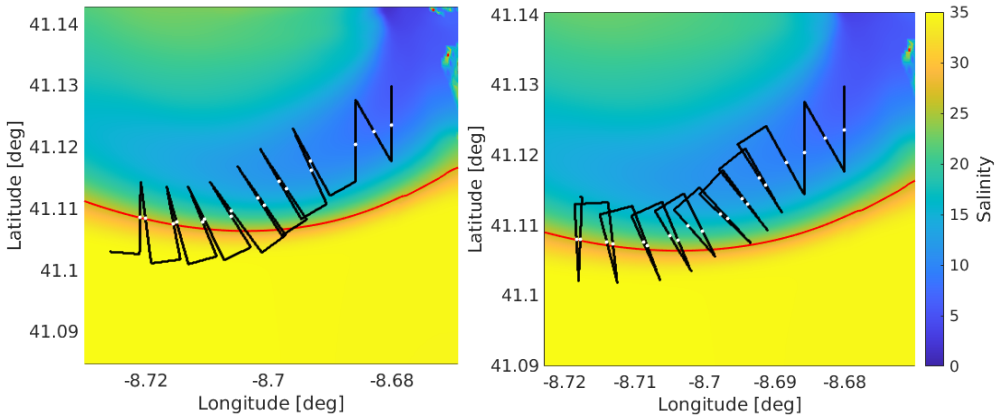
performing two transects in the same region, one of the parallel distances, D_{out} or D_{in}, must be zero.

Figure 5.13 shows an example of such behavior. We observe that two transects in the same region detect the front at different positions. The distance between points demonstrates that is possible to observe the progression of the plume with this method.

Note that this calculation or analysis is not a process done in real-time, as knowing the propagation velocity of the front has no influence on its prediction. If this method produces positive results in field testing, a consequent algorithm could use this calculation to improve the accuracy of the front's estimation.

Additionally, this trajectory could be implemented for only a limited number of front crossings. As it could only be necessary to perform such survey on one or two front crossings. The parameters that define the trajectory could be modified during a certain mission, and thus achieve different trajectory configurations for the same tracking of the plume.

One constraint of this approach is that the vehicle is limited to two transects in the same location. More crossings in the same region would provide a more accurate analysis of the plume propagation. However, such process with only the vehicle's data is not achievable in a trivial manner. Since the AUV must perform perpendicular transects, multiple segments in that same region would prevent the vehicle from performing any type of adaptation to its trajectory, since it would not be able to compute a valid estimation of the front's direction. Due to moments of fast plume progression, after continuously being in the same region for some time, the plume could be taking a significantly different direction. This problem is seen in the example from Figure 5.13, where we observe that the first detections of the front are now well in the interior of the plume at the end of the mission, this illustrates well the temporal variability of the plume.



(a) $D_{in} = 0$ and $D_{out} = 400$. (b) $D_{in} = 400$ and $D_{out} = 0$.

Figure 5.13: Vehicle's trajectory passing the region of the front two times. *Lawn Mower* strategy must have D_{in} or D_{out} set to zero.

5.4 Calculation of the plume parameters

Three main salinity values will characterize the behavior of the tracking mission – S_{high}, S_{low} and s.

S_{high} and S_{low} define the values that limit the survey ($\delta-$ band). These figures are predetermined and the front is always between these values.

The value that defines the front (i.e. the isoline $Iso_{(s,0)}$) will be determined by the first crossing of the plume. Therefore, for every different field mission, the front value is adapted to fit the best description of the current salinity profile.

Since the front is characterized by a steep gradient in the salinity field, a first survey of the plume's surface is completed to find the maximum derivative of the sampled values and the associated instant, t_{max}, of where that maximum rate of change occurs. s is the sampled salinity value at instant t_{max} (equation 5.11). This provides an estimate of the maximum gradient of the salinity field on that region.

$$\begin{aligned} t_{max} &= \{t : max\Big(\frac{\partial \hat{S}(x,y,0,t)}{\partial t}\Big)\} \\ s &= \hat{S}(x,y,z,t_{max}) \end{aligned} \tag{5.11}$$

Noise in the plume measurement can cause unwanted rates of change in the measured salinity, which would result in high derivative values. Thus, a low-pass filter is applied to $\hat{S}$, this is achieved with a moving average window of size $N = 10$. Thus,

$$\hat{S}_{lowpass}(T) = \frac{1}{N}\sum_{i=0}^{N-1} \hat{S}(T-i), \tag{5.12}$$

where T is the sample period of the CTD sensor, hence, $T = 1s$.

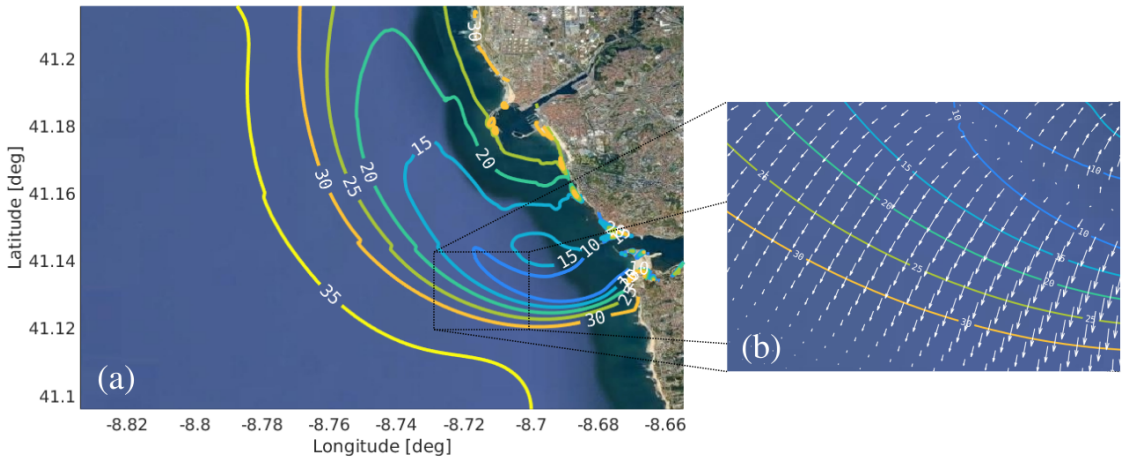


Figure 5.14: Salinity field of the Douro river plume represented by contour lines (a). Gradient of the salinity field (b).

After the calculation of s is completed, the size of the moving average window is modified to $N = 3$, since there is no need for calculating the derivative, the effect of the low-pass filter can be diminished.

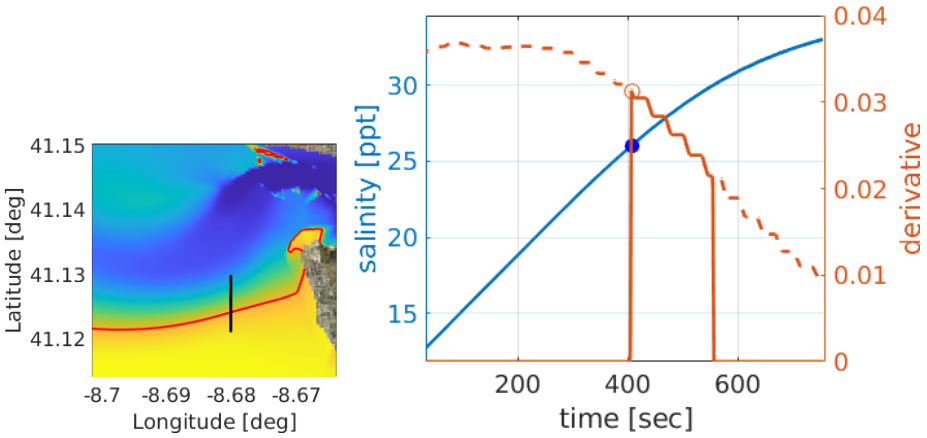
Image (a) from figure 5.14 shows a contour plot of the plume. In this situation, we observe that salinity values between $25 < S(x,y,0,t) < 30$ provide a valid front to perform a tracking mission. The agent must consider the front value, s, between these figures, otherwise, it will follow an isoline that does not characterize the plume's front.

As expected, the gradient is perpendicular to field lines (Figure 5.14b). Thus, undergoing the survey in a similar direction provides a fairly accurate estimation of the maximum gradient in the crossing region.

For this same example, it is possible to see that the gradient is not constant along the front (in the south region of the plume we have a large concentration of isolines, conversely to the upper region where there is less density). This indicates that a single reference value s does not accurately depict the behavior of the plume. In fact, a more accurate physical description of the plume describes the front as an area and not a single isoline.

A more obvious scenario that indicates that a single isoline does not describe the behavior of the plume, occurs in regions with a gentler gradient, where we can observe two local maxima in the salinity gradient. This is the case for the northern region of the exemplified plume.

Since the gradient and plume are a dynamic phenomenon and not physically described by a single isoline, it could be advantageous to update the value s at each transect. However, the main objective of the mission is to sample meaningful data from the front's area. The reference value s must provide a continuous contour line that helps guide the AUV, and thus sample transects of the plume in the $\delta-$band. Thus, the strategy that produces the best results is to keep the same reference value s for the entirety of the mission. By changing values we would be recording crossings of the front at a more accurate position in the maximum gradient of the plume, but we do not have a set of reliable points to predict the front's orientation, since these would not form a



(a) AUV trajectory. (b) Salinity plot and derivative.

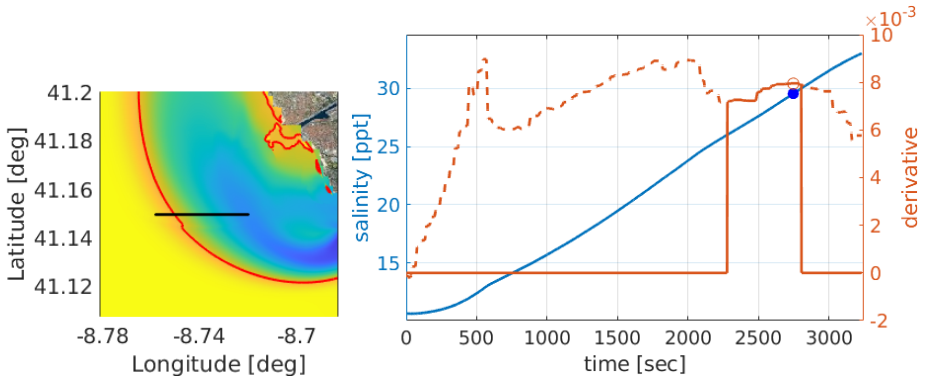
Figure 5.15: Initial survey to compute the reference salinity value s in the southern region of the plume.

continuous and smooth line.

Figure 5.15 shows an example of a surface survey. The biggest slope of the salinity values occurs at $\hat{S}(t) = 26$ and thus the front (Isoline $Iso_{(s=26,z=0)}$) is determined for this plume. We see in this example that the actual maximum derivative occurs at a lower salinity, however, we know that this value does not define the plume's front. In this instance, this occurs because the survey is done very close to the river mouth. To avoid this situation we define a possible value of the front, in this example, we compute the maximum derivative for values between $[26, 30]$. This problem is less frequent in locations further away from the river mouth. This provides a good method to autonomously detect a good isoline to serve as a reference for the vehicle. This window also provides a secure value if some error occurs or the vehicle does not have a perpendicular trajectory that could lead to a value s that does not represent the plume, in this case, a value in that window is assured and it will define the plume, even if not in an optimal manner.

Figure 5.16 shows a second survey on the same plume but at a different location. In this example, the situation of multiple local maxima is demonstrated. The computed reference value is $s = 29.52$, significantly higher than the previous location, this change is expected since the behavior of the plume differs in distinct locations. However, it is worth noting that both these values provide a valid front for the tracking algorithm. This result indicates that s will not stay as an optimal reference value throughout the mission.

Although this method indicates a value where a maximum rate of change occurs, it does not always correlate with an improvement in the tracking performance. In fact, in some situations in the Douro river plume, there seems to be no difference in performing this method or simply using a predetermined value, for instance $s = 28$, that is very likely to guarantee a good contour line for the vehicle to follow. However, in a more irregular plume where it is not possible to indicate with certainty a good reference value, this system could prove to be necessary. Since this method hints at bringing some advantages, it will be used in the algorithm, although further results, especially field tests are needed to further develop and retrieve meaningful conclusions.



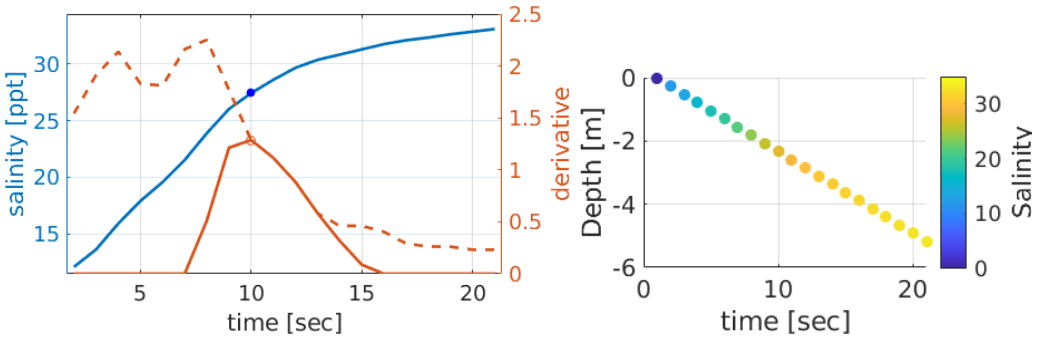
(a) AUV trajectory. (b) Salinity plot and derivative.

Figure 5.16: Initial survey to compute the reference salinity value s in the northern region of the plume.

The estimation of the maximum gradient is fairly accurate in the case of a perpendicular transect to the front, especially since this is done for hundreds of meters, and thus we have a very high-resolution image of the plume's surface profile. On the other hand, the AUV is performing a transect that does not sample the plume in its full spatial dimensions.

An alternative is to perform this survey, not at surface level, but crossing the isosurface $S(s,y,z,t) = s$, when performing the yo-yo segments. This would consist of significantly fewer values of salinity per survey, however, multiple surveys can be performed with every dive of the agent. From the multiple surveys, s is an average of the calculated values in each dive. Figure 5.17 shows an example of a single crossing.

The main advantage of this method is that it does not alter the main trajectory of the vehicle, meaning that the 3-dimensional tracking is not affected and it can start right from the vehicle's initial position. However, due to a very limited number of samples between the surface and the maximum depth, the derivative is not an accurate estimation of the front's profile. Additionally, we do not directly compute the front's surface, $S(x,y,0) = s$, which is the reference for defining the crossing points and consequently the estimation of the front γ. For the case of the Douro river plume, it does not bring any more accuracy than choosing a typical front value of the plume.



(a) Salinity plot and derivative. (b) Vertical profile of the sampled salinity.

Figure 5.17: Survey to compute the reference salinity value s with a vertical profile.

5.5 Mapping of the plume in 3-dimensional space

In this section, we consider the propositions depicted in problem 3. The agent must perform a 3D characterization of the plume structure. The AUV executes multiple dives along with the tracking of the front in a yo-yo trajectory. Figure 5.18 illustrates this trajectory. It is defined by the pitch angle, $\theta = 15^{\circ}$, between the path and the horizontal plane and the maximum depth, $z_{max} = 10m$.

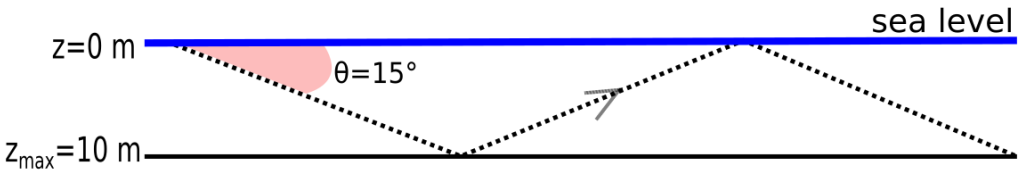


Figure 5.18: Schematic of the yo-yo trajectory taken by the vehicle.

With this added condition, the vehicle leaves the plume with every dive, since the plume has a typical depth of 2 meters. Since we are tracking the isoline $Iso_{s,z=0}$ the vehicle cannot detect a front crossing when intersecting the isosurface $S(x,y,z,t) = s$. Therefore, only values of $\hat{S}$ when $0 < z < 0.25$ will be used to control the tracking along the front. This ensures a similar behavior to the already capable algorithms studied in the xy plane.

This imposes a very different situation to the previous case, where the agent could continuously sample the surface and detect precisely the location of the front. The front may be missed by the vehicle, i.e. the isoline of value s in the plane $z = 0$ is located between surface points. This is exemplified by figure 5.19. The location of the crossing point, P_{cross_n}, is defined as the middle position of the two last surface points, $P_{surface}$ (Equation 5.13).

$$P_{cross_n} = \frac{P_{surface_i} + P_{surface_{i-1}}}{2} \tag{5.13}$$

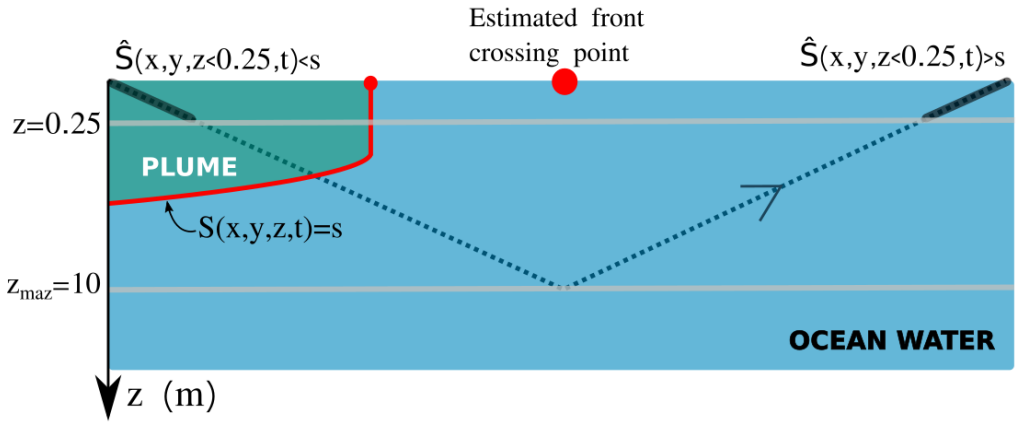


Figure 5.19: Vertical profile of the AUV yo-yo trajectory (black dashed line) while crossing the front. The estimated crossing point is the middle point between the surface locations.

The vehicle's position, $p \in \mathbb{R}^3$, is now defined as

$$p(t) = [x(t), y(t), z(t)]^T. \tag{5.14}$$

The vehicle's velocity, $\vec{v} \in \mathbb{R}^3$, is assumed to have a constant magnitude value v. In polar coordinates,

$$\vec{v} = (v, \psi(t), \theta(t)) \tag{5.15}$$

where ψ is the direction of travel on the horizontal plane, and θ is the angle between $\vec{v}$ and the xy plane.

Therefore, the dynamics of the agent are defined as,

$$\dot{p}(t) = \begin{bmatrix} v \cdot \cos\theta(t) \cdot \cos\psi(t) \\ v \cdot \cos\theta(t) \cdot \sin\psi(t) \\ v \cdot \sin\theta(t) \end{bmatrix}. \tag{5.16}$$

The same principles of the algorithms mentioned in section 5.3 are applied, except for the position of the crossing points that are defined by equation 5.13.

The space between surface points, defined by,

$$d_{surface} = 2 \cdot \frac{z_{max}}{\tan\theta}, \tag{5.17}$$

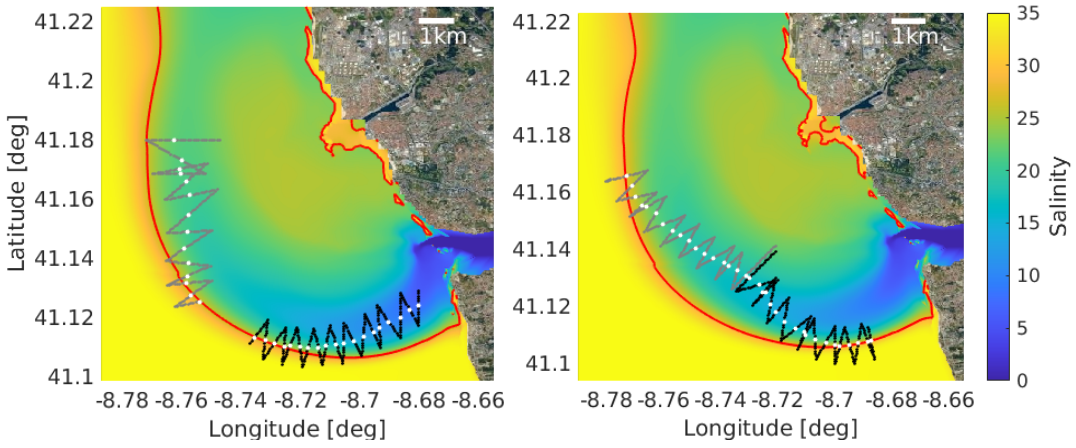
is the main difference from the application on two dimensions. The implication of $d_{surface}$ in the performance of the algorithms is explored in section 6.5.1.

5.6 Cooperative tracking

Multiple vehicles can be deployed when performing the tracking and mapping of the plume. Each vehicle runs independently, thus, no communication between agents occurs. We propose two main approaches to be tested in the Douro river plume.

The first approach is to position the vehicles at different edges of the plume. One AUV starts at the southern limit of the plume while the second agent starts at the northern limit, and from there, both follow the front in opposite directions. The algorithm is set to run in a clockwise direction for the vehicle located on the south of the plume, conversely, the second vehicle at the north of the operation area is set to a counter-clockwise direction. A similar approach was already field-tested in [2] and achieved positive results. An example of this first case is shown in Figure 5.20a.

The second approach consists of starting the vehicles in a similar latitude to the river mouth. The AUVs initiate the mission with similar orientations and then diverge in their front-following direction (as shown in 5.20b).



(a) (b)

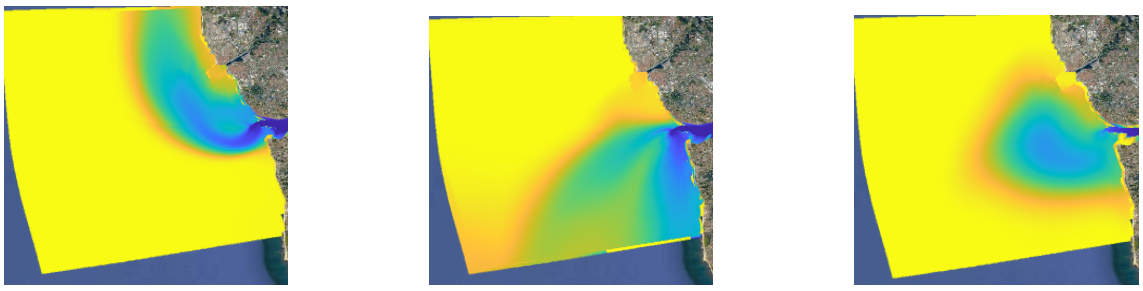
Figure 5.20: Simulation results for a two-vehicle mission. Situation (a) has the vehicle's start positions at the southern and northern edges of the plume. In (b), the AUVs have similar start positions and directions.

5.7 External and system parameters

In this section, we perform an overview of the main parameters that will influence the tracking system.

Starting by the external parameters – tide, wind and river outflow. As mentioned before, the tide determines the mission window – 6 hours during the ebb phase. The river outflow will be a consideration on the possible deployment of the vehicle, it can be the case that due to a very low or high river discharge there is not a defined plume or a plume too large to perform the mission. The wind plays a role in the possible shape of the plume, this will be the main metric to determine the AUV initial position. As mentioned in Chapter 2 there are three main shapes of the plume caused by South (5.21a), North (5.21b), and East or light winds (5.21c).

The initial direction of the vehicle is parallel to the coast. If there is a north wind the vehicle starts north of the river mouth with a northward direction. Conversely, if a south wind is identified we start the vehicle in a southward direction. The initial location is always fairly close to the river mouth to guarantee that the vehicle is inside the plume. Lastly, the southwest wind causes a circular-shaped plume where the vehicle can begin in both directions since it is always expected to cross the plume.



(a) South wind. (b) North wind. (c) East or light wind.

Figure 5.21: Possible shapes of the plume caused by wind direction.

The system parameters are divided into strategy, mapping, and vehicle/hardware parameters.

Therefore, the second set of parameters are the strategy parameters and the chosen trajectory. These will directly influence the behavior and performance of the front-following system. The three approaches to the tracking mission will yield different results, different parameters of those approaches will as well. The parameters are depicted in table 5.1. These values are user adaptable, meaning that before deploying the AUV a user may modify these values to achieve their mission goals in an optimal manner. The influence of these parameters is studied in section 6.6.

Strategy	**Parameters**	
Zigzag	α	
Alt. Zigzag	D_{out}	α
Lawn Mower	D_{out}	D_{in}

Table 5.1: Strategy parameters.

Additionally, we have the plume mapping parameters. These are chosen with the intention of describing the mapping behavior of the vehicle. These are the maximum vertical distance (z_{max}), the yoyo/pitch angle (θ), plume thresholds (S_{high}, S_{low}) and finally minimum and maximum distance covered by the agent after crossing the front (D_{min}, D_{max}). The requirements for the mapping of the Douro plume are defined in table 5.2.

Parameter	Value
z_{max}	10 m
θ	15 °
$[S_{high}, S_{low}]$	$[33, 23]$
Front window	$[26, 30]$
$[D_{min}, D_{max}]$	$[500, 750]m$

Table 5.2: Mapping parameters required for the tracking and mapping mission on the Douro river plume.

Finally, we have the vehicle and hardware parameters, these are typically constrained by energy consumption and hardware capability. The two main parameters to take note of are the velocity of the AUV and the sampling frequency of the CTD. The vehicle used in this mission has the following characteristics (Table 5.3).

Parameter	Value
Velocity	1 m/s
Sampling rate	1 Hz

Table 5.3: Vehicle parameters for tracking and mapping the Douro river plume.

These parameters will mainly influence the resolution of the sampled values, sampling rate and velocity will determine the number of samples per distance travelled. The velocity will also play an important role in the tracking of a dynamic front, its effect is shown in section 6.4.1.

5.8 Implementation on the LSTS toolchain

The implementation of the control algorithm on the AUV is accomplished through the LSTS toolchain, an open-source software that supports the deployment of ocean vehicles, developed by the Laboratório de Sistemas e Tecnologia Subaquática [7].

The main toolchain components are DUNE and Neptus, which communicate with the IMC message protocol [43]. Figure 5.22 illustrates a schematic of the LSTS toolchain structure.

Neptus is an external control software that monitors the planning and execution of the vehicle's movements [44]. The AUV can communicate with Neptus via acoustic signals (when underwater) or via Wi-Fi (when at the surface).

The DUNE software runs on the vehicle itself and provides a framework to develop the control algorithm. It handles the navigation and control of the vehicle, communication, logging and interactions with the vehicle's hardware.

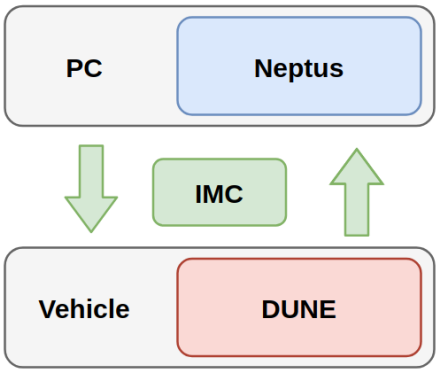


Figure 5.22: LSTS toolchain structure.

DUNE is supported for all POSIX-compliant systems and the code is written in C++. Its compilation is handled with the open-source software CMake.

The DUNE structure is divided into tasks. Tasks run in different threads with no shared resources and communicate with each other asynchronously with the IMC protocol (Figure 5.23).

A configuration file dictates which tasks are enabled and what are their parameters. DUNE also allows for the simulation of the vehicle and its sensors/actuators, this way the system can be validated before testing in the vehicle's hardware. The implementation for this thesis was validated with simulation tasks, in addition, we created a new task that simulated the salinity sensor with data from the DELFT3D model. Note that the code behaves in the same manner performing on the simulated tasks or the hardware, thus, by validating the system with the DUNE simulation environment we are certain that the vehicle will be compatible and functional with that same implementation.

Additionally, DUNE also provides safety mechanisms. A few examples are fuel level management, if it is detected that the vehicle is low in energy an automatic safety maneuver is activated, if communications are lost the vehicle immediately returns to base and, finally, the user can define an operation limits for the mission area and if a maneuver is planned to cross that area the plan is cancelled [7].

Let us now present our implementation.

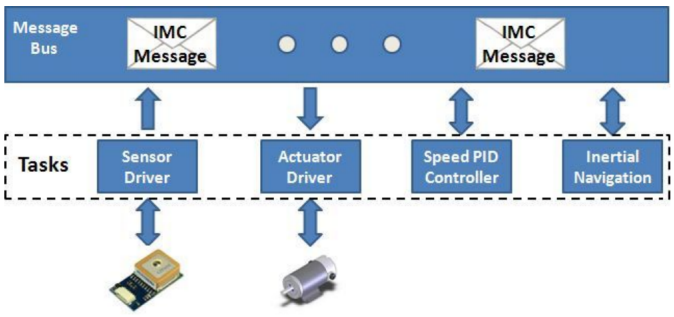


Figure 5.23: Description of DUNE structure. Image from the Laboratório de Sistemas e Tecnologia Subaquática.

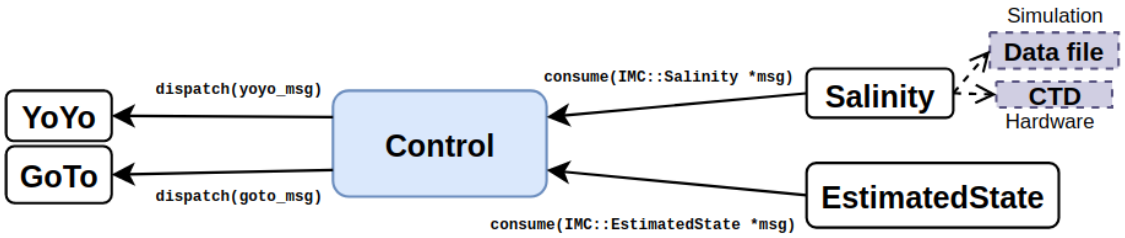


Figure 5.24: Structure of the Control task interactions. Received and dispatched IMC messages transmitted to/by other DUNE tasks.

The parameters of the algorithm are user adjustable, thus a configuration file can be modified to select the desired parameters. Part of the configuration file looks as follows:

```
[DouroPlume.Control] // Control task has the following
   parameters:
Start Longitude = -8.682607
Start Latitude  = 41.137636
Initial Heading = -180.0
z_max           = 10
theta           = 15
alpha           = 40
D_max           = 750
D_min           = 500
D_out           = 400
D_in            = 300
D_min_cross     = 500
Alt_zigzag      = false
Lawn_mower      = false
D_waypoint      = 5000
```

This file allows us to easily define the chosen strategy, its parameters and the initial position and heading of the vehicle. The variable `D_waypoint` is explained later in this section.

DUNE provides tasks able to interact with all sensors and actuators of the AUV. In the mentioned configuration file, all necessary tasks are enabled and their parameters defined. The shown section of the file indicates the parameters for the Control task developed to run the front-tracking algorithms. This task will directly communicate with other tasks through four IMC messages, as described in Figure 5.24.

The Control task implements the algorithms for front estimation and trajectory adaptation as described in the previous sections. To detect and determine the front crossing positions the Control task must have access to the position of the vehicle in its 3-dimensions and the salinity values measured by the CTD sensor. DUNE has two messages that communicate these values – Salinity and EstimatedState. Both messages are periodically broadcasted with the measured salinity and the physical state of the vehicle (including position, orientation and velocity).

The Control task subscribes to the messages with:

```
bind<IMC::Salinity>(this);
bind<IMC::EstimatedState>(this);
```

Upon binding to the necessary messages, it is possible to access them with the following functions:

```
void consume(IMC::Salinity const* msg) {
  // ... some code ...
  // save salinity value with corresponding coordinates
  salinity_stream.push_back({ lat, lon, msg->value });
}
void consume(IMC::EstimatedState const* msg) {
  // ... some code ...
  Coordinates::toWGS84(*msg, lat, lon);
  AUV_coordinates.lat = lat;
  AUV_coordinates.lon = lon;
  AUV_depth = msg->depth;
 }
```

Thus, the control algorithm has access to all needed variables to perform the tracking of the front. If the condition `AUV_depth < 0.25` is true, the algorithm runs. If a front crossing is detected, a new estimation of the front's direction γ is computed. And a new change of direction ψ is calculated if a salinity (S_{high}, S_{low}) or distance (D_{low}, D_{min}) threshold is met.

The movement of the vehicle is controlled by a system of maneuvers. Each maneuver has a different respective message. Two different maneuvers are used – YoYo and GoTo. Both messages have a set of parameters that describe the behavior of the maneuver.

The GoTo maneuver is specified by a target waypoint and a reference depth value. The vehicle will direct itself to the waypoint, described by latitude and longitude, at the determined depth.

The YoYo maneuver causes the vehicle to perform a yo-yo trajectory as described in section 5.5. This maneuver is defined by a waypoint, the amplitude of the trajectory and the pitch angle.

The messages are dispatched by the Control task with the function `dispatch(imc_msg)` as depicted in Figure 5.24.

The GoTo maneuver is used when the vehicle travels at the surface. Firstly, to guide the vehicle to the initial start position determined in the configuration file. The waypoint for the first maneuver is the start position coordinates. This is the first movement of the vehicle and is initiated when the Control task dispatched the message:

```
goto_msg.lat = Angles::radians(start_lat);
goto_msg.lon = Angles::radians(start_lon);
goto_msg.z = 0.0;
dispatch(goto_msg);
```

When the vehicle reaches the start position another GoTo maneuver is initiated with the direction of the initial heading and a waypoint located 5km away from the vehicle's current position, $p(t)$. This distance is determined by the variable `D_waypoint`.

The waypoint, $P_{waypoint}$, is determined as

$$P_{waypoint} = p(t) + \begin{bmatrix} D_{waypoint} \cdot \cos\psi \\ D_{waypoint} \cdot \sin\psi \end{bmatrix}. \tag{5.18}$$

The second GoTo maneuver will perform the first front crossing and will determine the front reference value s.

After the first transect, the subsequent maneuvers are all determined by YoYo messages, and the waypoint generation is determined in the same way as defined by equation 5.18.

When the algorithm updates the direction of the vehicle, `dispatch(yoyo_msg)` is called. Since we must cross the front, the waypoint is generated at a large distance to guarantee that the vehicle is able to detect the front. If the vehicle reaches the waypoint before crossing the front, the mission is terminated.

Figure 5.25 shows the Neptus interface, where it is possible to observe the path of the vehicle and the generated waypoint (in yellow). Figure 5.26 shows the salinity and depth data communicated by the vehicle to Neptus.

When the vehicle is underwater, it does not have access to its position. The influence of the water velocity can cause a deviation in the trajectory of the vehicle. Thus, every time the vehicle surfaces it adjust its orientation to the current waypoint location. This path correction method can minimize the effect of ocean currents. This method was replicated with the MATLAB simulation environment and is tested in section 6.7.

The simulations with the DUNE/Neptus systems achieved similar results to the MATLAB simulations. We consider the system to ready to be tested on the AUV hardware and be deployed in a field setting.

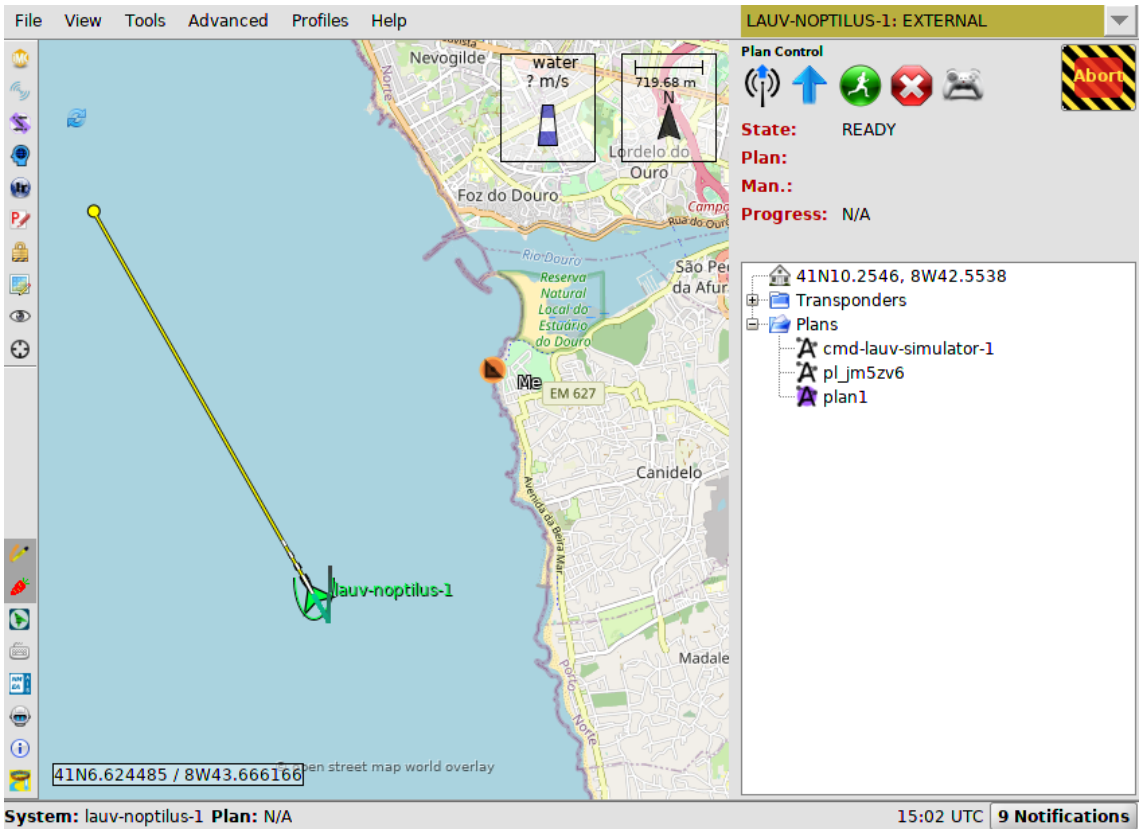


Figure 5.25: Neptus interface describing the vehicle's trajectory and current waypoint (in yellow).

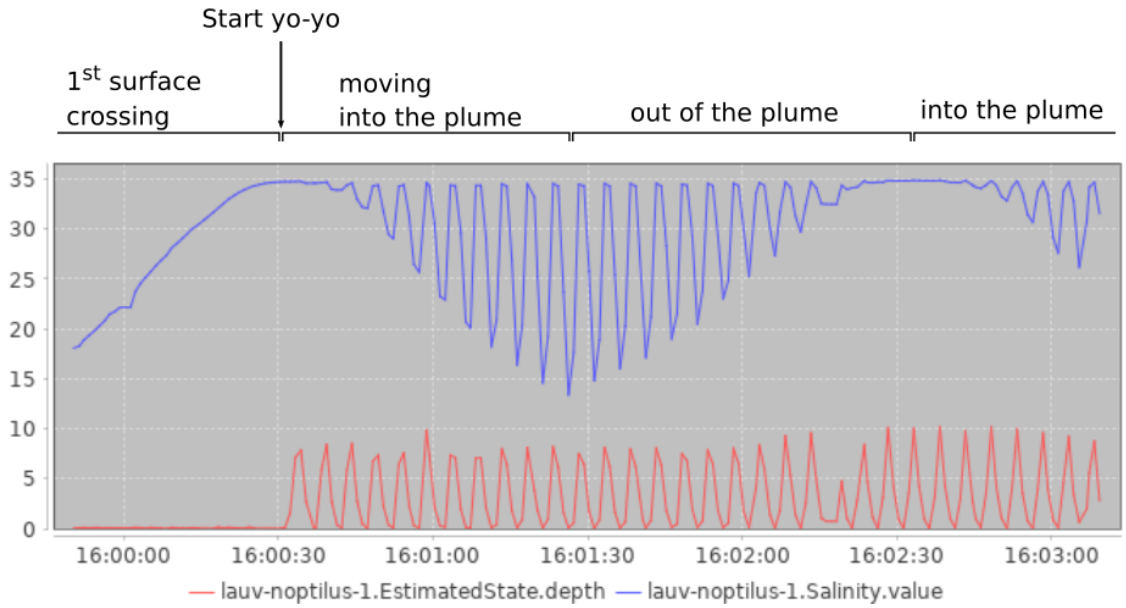


Figure 5.26: Salinity (in blue) and depth (in red) communicated by the vehicle to Neptus. Data is observed in the Neptus software.

Chapter 6

Results

The test results are shown in this chapter. Firstly, the MATLAB test environment is described, following a depiction of the test plan and the performance metrics that quantitatively evaluate the results. Simulations were done for static and dynamic plumes, surface and 3-dimensional tracking, and with and without the influence of ocean currents. In addition, all three strategies and their respective parameters were tested. Finally, we provide a discussion of the results.

6.1 Simulation environment

To test and validate the developed algorithms, a simulation environment was developed in MATLAB. The DELFT3D model provided the salinity and ocean current data, as mentioned in section 2.1.4. The vehicle dynamics were simulated as depicted in section 2.3.3. Additionally, the same system of waypoints generation, applied by the LSTS toolchain, is also present in this simulation environment, in this manner the path correction method applied by DUNE is accounted for these results. The simulation updates with a period of 1 second.

Since the values provided by the ocean model did not have a uniform distribution, the MATLAB function `scatteredInterpolant()` was used to create a uniform grid with a horizontal resolution in the *xy* plane of 0.0004 $\times$ 0.0004 degrees and a vertical resolution of 0.25 m. The final data grid is 500 $\times$ 525 $\times$ 41 points, with the following limits: $Longitude = [-8.86; -8.65]^{\circ}$, $Latitude = [41.03, 41.23]^{\circ}$ and $Depth = [0, 10]$m.

The resolution needed for the simulation is much higher than the provided by these values, as such, for each step of the vehicle, a bilinear interpolation is performed on the *xy* plane, following a linear interpolation regarding the vertical position. This ensures a continuous salinity field that resembles the real salinity distribution of the Douro plume.

The salinity and ocean current data produce time-varying fields, the provided resolution is only 30 minutes, thus, a linear interpolation is also performed between time limits.

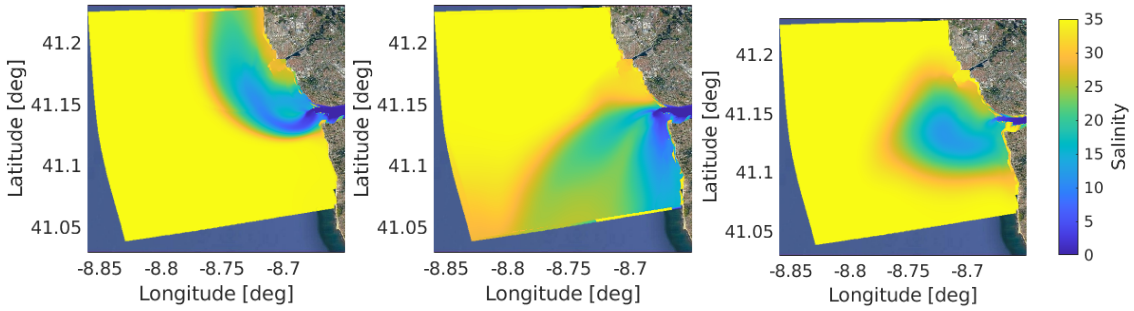
To calculate the angle difference between the direction of the vehicle and the perpendicular line to the front, we use the gradient direction of the field. The gradient is perpendicular to the

field lines, as was shown in section 5.4. The salinity gradient is computed with the function `gradient()`.

The vehicle's control algorithm was implemented as described in Chapter 5.

6.2 Test plan

The test cases shown in this chapter are executed in three plumes that represent the three main plume shapes (Figure 5.21). To ease their description, the three instances will be abbreviated as *N, S* and *LW*, representing the wind direction at each case.



(a) 9th May 2016 - *S plume* (b) 15th May 2016 - *N plume* (c) 2nd May 2017 - *LW plume*

Figure 6.1: Salinity fields of the plumes representing the three main variations caused by a Southern, Northern, East and/or light wind, respectively, (a), (b) and (c).

The plumes present some defining characteristics, the *S plume* is a fast propagating plume with a well-defined front caused by a steep gradient, and a plume depth of about 2 meters. Conversely, the *LW plume* is a slow plume with a wide horizontal front due to a low gradient, and a depth between 0.5 and 1 meter. The *N plume* presents itself as a middle ground between the two previous cases, in propagation speed as well as in the plume's depth. This selection provides varying characteristics that can be met in-field testing, the variety of these situations provides a good framework to test and validate the tracking algorithms in different conditions.

In addition to the type of plume, two main parameters characterize the plume in a certain test simulation – temporal behavior and ocean currents. The plume can be static or dynamic. In the case of a static plume, dependencies on time are dropped, while in a dynamic plume the mission is performed in the 6-hour mission window. Finally, the ocean currents can influence the trajectory of the vehicle or are assumed to be zero.

These three last characteristics define the plume field, to fully characterize a simulation, the vehicle's strategy and parameters must also be defined. The vehicle can be performing one of the three strategies – *Zigzag*, *Alt. Zigzag* and *Lawn Mower*. All three strategies must also be defined by their parameters (α, D_{out} and D_{in}). In addition, the AUV can perform a surface or 3-dimensional tracking trajectory. The 3D tracking implies the use of the yo-yo trajectory defined by the pitch angle θ and maximum depth z_{max}.

A total of 307 tests were simulated, all are described by a combination of the parameters described above. To organize and ease the reference of each individual simulation, the following table depicts how these will be numbered and referenced throughout the rest of Chapter 6.

#	Description
	Ocean currents
0	Ocean currents behavior are assumed to be zero.
1	Water velocity influences the vehicle's trajectory.
2	Path correction method is utilized.
	Temporal behavior of the plume
0	Static plume – no time propagation.
1	Dynamic plume with a 6-hour mission time.
	Vehicle's movement in the z axis
0	Surface tracking.
1	3-dimensional mapping – AUV performs an yo-yo trajectory.
	Plume
1	Test performed in the *S plume*.
2	Test performed in the *N plume*.
3	Test performed in the *LW plume*.
	Strategy
1	*Zigzag*.
2	*Alternative Zigzag*.
3	*Lawn Mower*.

Table 6.1: Description of the parameters that define the test conditions and respective number used to identify them.

For example, if a certain test assumes the water velocity as zero, with a static and 3-dimensional *N* plume and a vehicle performing a *Zigzag* strategy – the simulation ID is #00121.

Table 6.2 provides an overview of the simulations and the respective section where these are presented. To show that all possibilities of a certain condition were tested, the corresponding digit is represented with the letter *X*. In addition, we attribute a denomination for a given set of simulations.

Set	Conditions	No. of tests	Description	Section
A	000XX	9	Validate surface tracking systems for a static plume and compare the behaviors of the three different strategies.	6.3
B.1	010XX	9	Validate systems for a surface tracking of a dynamic plume.	6.4.1
B.2	01011 01021	16	Simulations performed for multiple AUV velocities, v.	6.4.2
C.1	011XX	9	Validate tracking algorithms for a 3D mapping of a dynamic plume.	6.5.1
C.2	001X1	30	Simulation for multiple yo-yo configurations (θ and z_{max}).	6.5.2
D.1	000XX	96	Testing for different strategy parameters (α, D_{out} and D_{in}).	6.6
D.2	011XX	120		
E	111XX 211XX	18	Test the influence of ocean currents and the path correction method.	6.7

Table 6.2: Description of all performed simulation sets.

Unless specified otherwise, the system parameters are the following:

θ	15 °
z_{max}	10 m
α	30 °
D_{out}	400 m
D_{in}	300 m
$[D_{min}; D_{max}]$	[500;700] m
v	1 m/s

Table 6.3: Default parameters for the simulations.

Note that θ and z_{max} are only applied to a 3-dimensional mapping of the plume since these describe the yo-yo trajectory. The AUV velocity, v, is only considered in a dynamic plume.

A description of the simulations sets now follows:

Set A: This set of simulations considers a surface tracking system for a static plume. The default parameters are considered. The conditions are tested for all three trajectories in the three plumes, providing a total of 9 tests.

Set B.1: Maintaining the configuration of set A, these simulations are now executed in a dynamic plume.

Set B.2: With the introduction of a time-varying plume the vehicle's velocity is now an important factor. To study its influence, eight different velocity values will be applied to the vehicle:

v [m/s]	0.5	1.0	1.5	2.0	2.5	3.0	4.0	5.0

Table 6.4: Velocity, v, values taken by the AUV in the test set B.2.

This will be executed with a *Zigzag* strategy and in the *S* and *N* plumes. Providing a total of 16 simulations.

Set C.1: Analogous to test sets A and B.1, the system is tested in a dynamic plume with the 3-dimensional mapping. Nine simulations are executed for all trajectories and plumes with the default system parameters.

Set C.2: With the introduction of the yo-yo trajectory, the vehicle samples the surface of the front periodically, and not continuously, as was the case for the previous simulations. To understand the influence of this effect, multiple configurations of the yo-yo trajectory are applied that modify the distance between surface points, $d_{surface}$:

θ [°]	15	15	15	15	15	15	10	8	5	2.5
z_{max} [m]	0.5	2.0	4.0	6.0	8.0	10.0	10.0	10.0	10.0	10.0
$d_{surface}$ [m]	3.7	14.9	29.9	44.8	59.7	74.6	113.4	142.3	228.6	458.1

Table 6.5: Yo-yo trajectory configurations executed in the test set C.2.

Set D.1 and D.2: In these sets of simulations the influence of the different parameters α, D_{out} and D_{in} are studied for a static and dynamic plume. Each parameter is individually studied. Apart from the variable in the analysis, all other parameters take their default values.

The execution of these sets of conditions resulted in 216 simulated missions.

The three parameters will take the following values:

α [°]	20	30	40	50	60	70

Table 6.6: Angle, α, taken by the AUV strategy in the test sets D.1 and D.2.

D_{out} [m]	100	200	300	400	500	600	700	800

Table 6.7: Distances, D_{out}, taken by the AUV strategy in the test set D.1 and D.2.

D_{in} [m]	100	200	300	400	500	600	700	800

Table 6.8: Distances, D_{in}, taken by the AUV strategy in the test set D.1 and D.2.

Set E: Finally, this set provides the most realistic simulations by accounting for the ocean currents. These results will be compared to the previous simulations where the currents are assumed to be zero. Additionally, the simulation is also executed with and without the path correction method applied by DUNE that adjusts the vehicle direction at surface locations.

6.2.1 Performance metrics

To evaluate and compare different test results, the following performance metrics were developed (Table 6.9).

Metric	Comments
Quality	Classifies the mission in regards to the quality of the front prediction.
Robustness	Ability to finish the mission without losing the front.
Front progression	Ability to sample the maximum area of the plume within the given time-window.
Efficiency	Ability to have the maximum number of optimal transects from the total crossings of the front.
Spatial resolution	Density of the retrieved salinity values of the plume.

Table 6.9: Performance metrics.

The quality of the mission will be measured by the average error of the front prediction. Defined as,

$$\varepsilon_{avg} = \frac{1}{n}\sum_{i=1}^{n}||\varepsilon_i||, \tag{6.1}$$

in addition, we also categorize each individual transect that has a direction perpendicular to the front prediction, allocated to the following categories represented in table 6.10. The classification is designed to reflect the quality of the transect with the prospect of a posterior oceanographic study.

Error interval (°)	Classification
$[0,20[$	Optimal
$[20,40[$	OK
$[40,90]$	Poor

Table 6.10: Classification of the perpendicular crossings to the front in respect to the error between the plume's front and its prediction, ε.

The robustness of the system is measured by the vehicle's ability to not lose the front and not terminate the mission prematurely.

The ability to map a large portion of the plume is desirable, as such, we will classify the plume progression by calculating the front's length over the mission time, t_M. The resulting value, v_{fp}, is measured in m/s. The plume progression is defined as,

$$v_{fp} = \frac{1}{t_M}\sum_{i=2}^{n}||\vec{P}_{cross_i} - \vec{P}_{cross_{i-1}}||. \tag{6.2}$$

The efficiency of the system, μ, provides a percentage of the total number of front crossings that provide an optimal transect of the plume (6.3).

$$\mu = \frac{\#\text{ optimal transects}}{\#\text{ total crossings}} \tag{6.3}$$

Finally, we have the spatial resolution of the plume mapping. This will depend on the parameters of the yo-yo trajectory, for example, a larger angle θ would make the vehicle perform

more vertical dives per transect providing more detailed information of the plume. The other main parameter is the sampling frequency. The requirements for the Douro plume mission are set with a sampling frequency of 1 Hz and a yo-yo with a pitch angle of 15° and a maximum depth z_{max} of 10 meters. As a result, a large majority of simulations are always executed with these values. For this reason, a quantitative metric will not be developed.

6.3 Surface tracking of a static plume

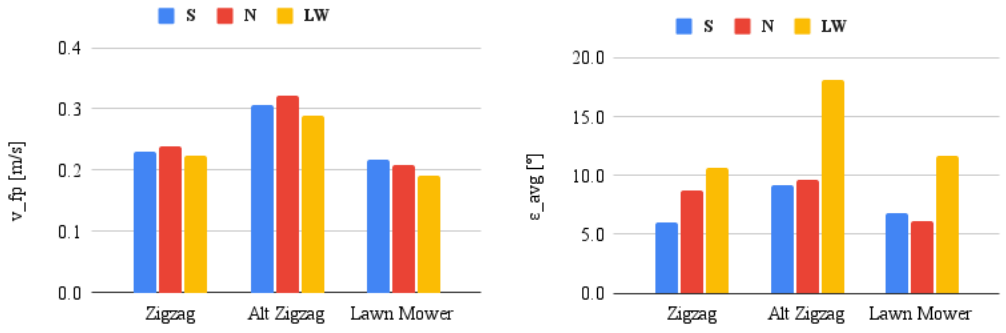
The first test results from set **A** simulate a surface tracking of a static plume. All three strategies were tested, by minimising external factors (such as time variability of the plume, 3D mapping and ocean currents) a good test condition is achieved to compare the main characteristics between the three different strategies.

Since there is no temporal variability to the plume, the simulation runs until the algorithm completes the entirety of the front. This allows us to have more significant results, and a more accurate comparison between strategies can be made.

The test results for a static plume are shown in Appendix A.1. All three strategies are able to successfully track the front.

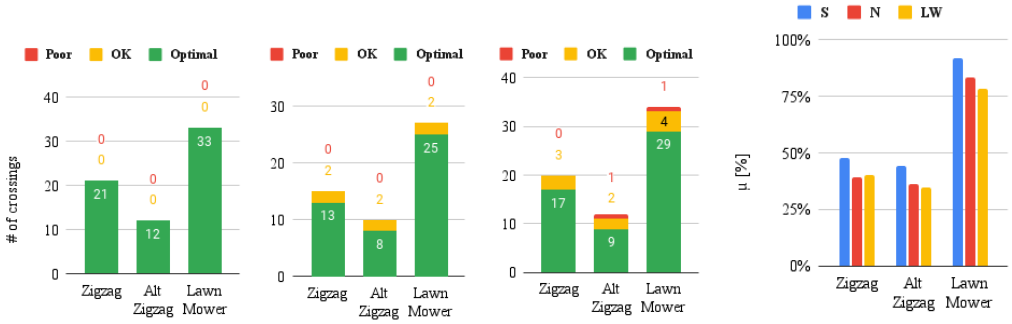
Figure 6.2a illustrates the front progression results. As expected, we see a faster progression occurring for the *Alt. Zigzag*. The slower progression is achieved by the *Lawn Mower* strategy. This metric is mainly influenced by the type of trajectory, as we can observe by the fairly similar results between the same type of plumes.

The strategy *Alt. Zigzag* also has the highest average error, ε_{avg} (Figure 6.2b). This can be explained by assessing the previous result – a faster progression results in less distance between crossing points, and thus a worse prediction of the front's direction is achieved. Although this effect is noticeable, the main influence on this metric is the irregularity and curvature of the front. Therefore, we get the worst result for the round-shaped plume, in the other plumes there is an initial curvature that follows a more rectilinear front that is more easily tracked by the system.



(a) (b)

Figure 6.2: Front progression, v_{fp}, (a) and average error, ε_{avg}, (b) results for a surface tracking of a static plume.



(a) *S plume* (b) *N plume* (c) *LW plume* (d)

Figure 6.3: Crossings distribution (a,b,c) and efficiency, μ, (d) results for a surface tracking of a static plume.

Figure 6.3 shows the distribution of the perpendicular transects in regards to their classification. The *Lawn Mower* strategy has a larger number of transects because the vehicle will cross the front at a perpendicular angle in both directions. The other strategies may have an optimal transect when going from the inside to the outside of the plume, however, this is not a planed perpendicular transect, therefore, it is not considered in these results. These figures demonstrate all runs to be successful in tracking a static plume.

Finally, we can see what strategy is more efficient (i.e. ratio of optimal transects over the total number of crossings). As expected, the strategy *Lawn Mower* has the best performance, since the vehicle is always performing perpendicular segments (Figure 6.3d). As seen in the previous metric, the vast majority of segments were classified as optimal, thus the efficiency reflects those ideal results. Note that the maximum result achieved by the first two strategies is 50%.

We can conclude from these results, that all strategies track a surface and static plume with considerable success.

6.4 Surface tracking of a dynamic plume

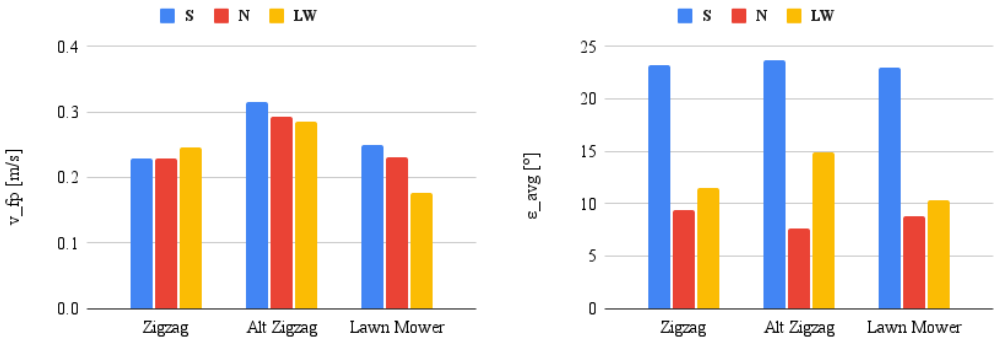
6.4.1 Effect of a dynamic plume

Following the structure from the previous section, the plume has now temporal propagation and the tests will have a real mission time of 6 hours. The following results are obtained by running the simulation set **B.1**.

The test results for a dynamic plume are shown in Appendix A.2. In all nine results, the vehicle was able to successfully follow the front.

The front progression velocity remains fairly similar to the static situation. This is expected since the main variable responsible for this is effect is the variation in trajectories. All parameters remain the same, thus, we observe a very similar graph (Figure 6.4a).

The southerly wind plume has a fast propagating front, especially at the beginning section of the tracking mission. The high variability of the front makes its prediction lose accuracy, as is



(a) (b)

Figure 6.4: Front progression,v_{fp}, (a) and average error, ε_{avg}, (b) results for a surface tracking of a dynamic plume.

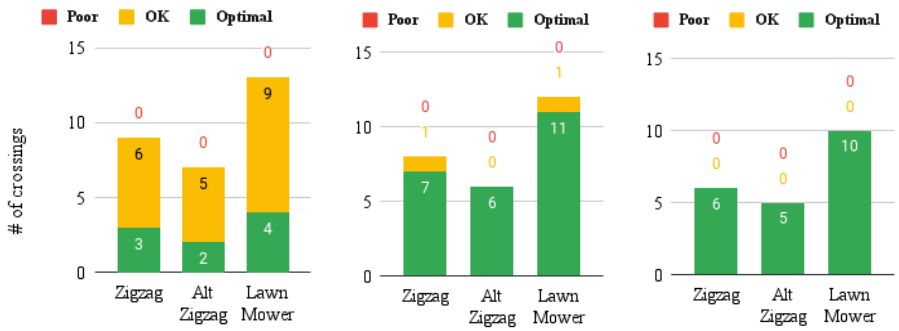
evident by Figure 6.4b, where the *S plume* suffers a higher increase in the average error compared to the other plumes that have a slower moving front.

The decrease in tracking performance for the S plume is also clear when accounting for the number of optimal transects in all strategies (Figure 6.5). Nevertheless, even with the difficult conditions of this plume, the tracking of the front is still achievable, and with transects that still map and sample the plume in a positive manner. Additionally, we see that for the other two plumes the performance was less affected and produced almost ideal results. This is good evidence that the tracking system scales well from a static to a dynamic plume.

6.4.2 Effect of the AUV velocity

Using the situation described in this section, we can study the relationship between tracking performance and vehicle velocity with the set of simulations **B.2**. The test results are shown in Figure A.3 in Appendix A.2.

In this set, we utilize the *S* and *N* plumes. We choose the *S plume* because it has a large propagation speed that negatively influenced the results, as was seen in figure 6.5. Conversely,



(a) *S plume* (b) *N plume* (c) *LW plume*

Figure 6.5: Crossings distribution for a surface tracking of a dynamic plume.

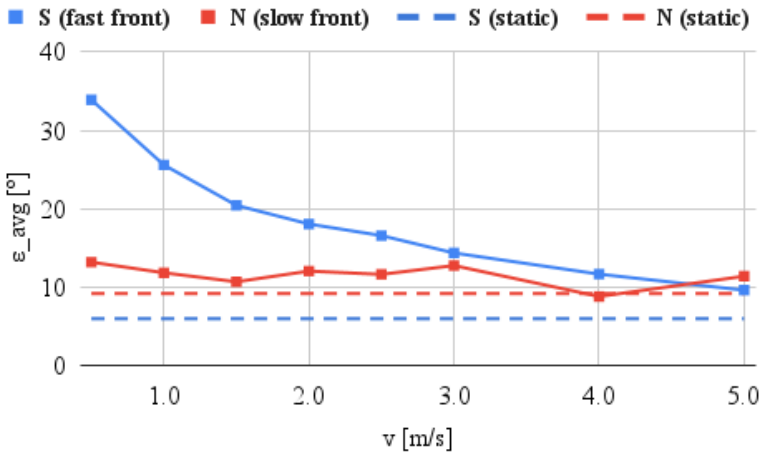


Figure 6.6: Average error, ε_{avg}, for different vehicle velocities, v.

the *N plume* will represent a slow front. The test was performed only with the *Zigzag* strategy. Since the plume propagation influences the performance in a similar manner for all strategies, these results are replicable for all three trajectories.

The travel distance of the AUV was limited to $15km$ to observe the influence of AUV velocity variation on the same region of the plume.

The results are shown in Figure 6.6. Note that some of the velocities used in this test are not feasible, however, these results are still helpful to conclude the role of the vehicle's speed in the tracking mission. Starting with the fast front (*S plume*) represented in blue, we observe that the increase in velocity produces a better front prediction. The dashed line indicates the average error for the same test, but with a static plume. In fact, we can say that the static plume condition represents an infinite vehicle velocity. Thus, as we increase the velocity, the error gets closer to that value. This effect was expected and is noticeable in a very fast propagating plume.

On the other hand, the *N plume* has a slower front, and the results show that an increase in velocity does not translate to an increase in performance. This is explained by the front being relatively constant in its location, thus, there is not a significant advantage for a faster vehicle velocity. Another good indicator for this explanation is the fact that the error for the static plume maintains itself very close to the values for different velocities.

As expected, the increase in the AUV velocity correlates to an increase in performance in plumes with higher temporal variability, and in plumes with a more constant position this effect is almost negligible.

6.5 3D tracking

6.5.1 Effect of the 3-dimensional trajectory

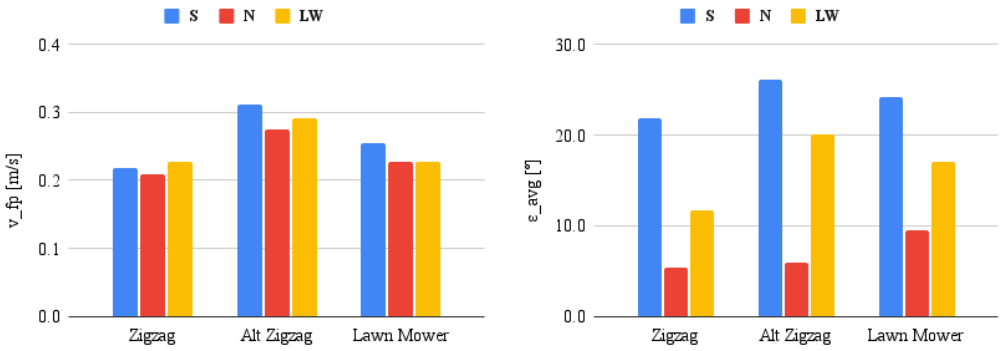
The full spatial mapping of the plume requires sampling salinity values at different depths, as mentioned before the approach for this is a yo-yo trajectory between $z = 0$ and $z_{max} = 10$ with a pitch angle of 15 degrees. Variables that govern the horizontal trajectory of the vehicle are the same as the previous tests. To test the system in a dynamic 3-dimensional plume, the simulations from set **C.1** were analysed.

All missions were able to follow the front for the entirety of the 6-hour time window. The results are shown in Appendix A.3.

With regards to the front progression, the inclusion of vertical dives does not impose a significant difference (Figure 6.7a). This is expected because the horizontal velocity of the vehicle has a similar magnitude as before. The vehicle's absolute speed is $v = 1\ m/s$, and with a pitch angle $\theta = 15^{\circ}$ the horizontal velocity, v_{bH}, is

$$v_{bH} = v \cdot \cos\theta = \cos(15^{\circ}) = 0.97\ m/s. \tag{6.4}$$

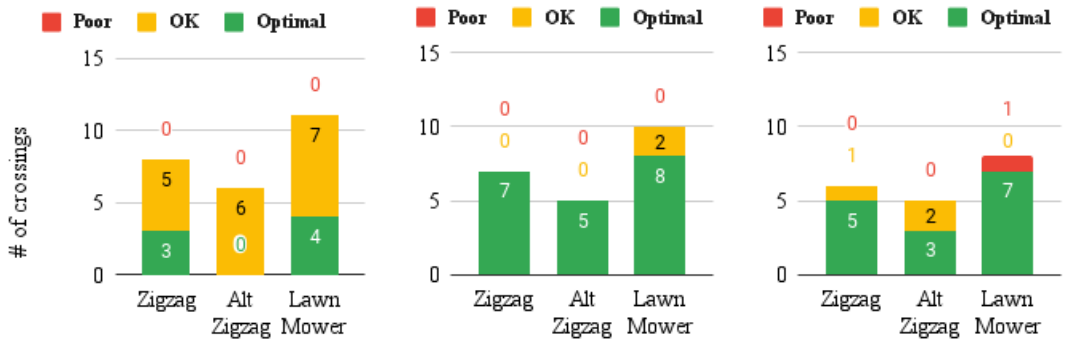
We can conclude that the difference in velocity caused by the yo-yo, with these parameters, does not affect in a significant way the front progression.



(a) (b)

Figure 6.7: Front progression, v_{fp}, (a) and average error, ε_{avg}, (b) results for the 3D tracking and mapping of a dynamic plume.

The distance between surface points, $d_{surface}$, is now 75 meters, therefore, the spatial resolution for finding the location of the front is $\frac{75}{2} = 37.5\ m$, since the front position, P_{cross_n}, is determined as the middle point between the two last surface locations. The drop in resolution is expected to cause an increase in the average error. In fact, an increase is observed for the *LW* and *S* plumes, however, such behavior is not present in the *N* plume. This indicates that this drop in resolution (distance between surface samples) is not as significant as expected. To understand at what value this effect becomes more significant, simulations were performed with different surface distances.



(a) *S plume* (b) *N plume* (c) *LW plume*

Figure 6.8: Crossings distribution for the 3D tracking and mapping of a dynamic plume.

In terms of transects, the performance of the mission was still positive. With the majority of the missions having multiple and in some cases all perpendicular transects in the optimal condition (Figure 6.8).

6.5.2 Effect of the distance between surface locations

To test the influence of the distance between surface points in the tracking performance, the simulation set **C.2** was studied. The test results are shown in Figure A.5 in Appendix A.3.

The test is performed with a static plume to measure the front estimation error in a full plume tracking, and thus achieving more significant results. A disadvantage of considering a dynamic plume is that changing the yo-yo trajectory will cause different horizontal vehicle velocities. This would add another variable to the tests while we are trying to isolate the results for $d_{surface}$.

The hypothesis that the average prediction error would increase with fewer surface locations is verified by the results in figure 6.9. An upwards tendency is verified in all three plumes.

It is important to note that in distances lower than 100 meters, this tendency is not noticed. This provides good evidence that in this range, the 3D mapping of the plume does not heavily influence the tracking performance.

In the *LW plume*, we observe a more irregular behavior. This plume has a smoother gradient, and consequently, a wider front detection region. This results in a higher variability and less accuracy for the plume crossing positions. Thus, the effect of less precision in locating the front is less noticeable in this case. Nonetheless, an overall upper tendency is still verified, although not in the linear manner as was observed for the *N* and *S* plumes.

It was expected that with distances larger than 400 meters the vehicle would not be able to track the plume. Even though the tests were executed with a static plume, these results were still surprising and show a high level of robustness.

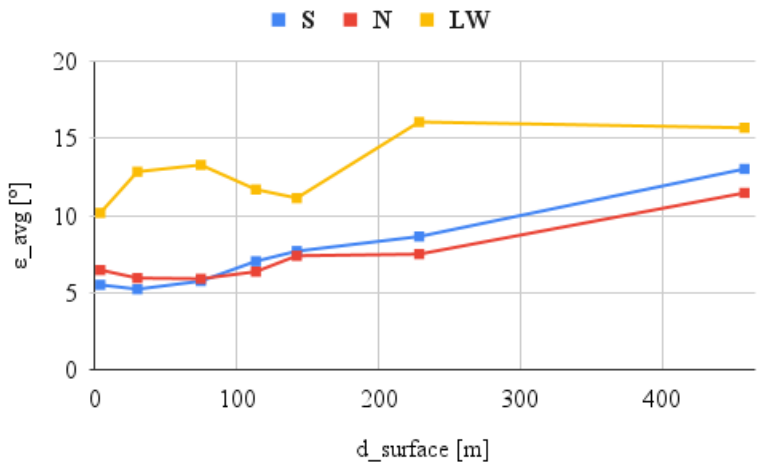


Figure 6.9: Effect of the distance between surface points, $d_{surface}$, on the average error, ε_{avg}, while the vehicle performs the *Zigzag* trajectory.

6.6 Effect of different trajectory parameters

Previous sections showed the different characteristics of the three tracking strategies and the influence of a dynamic and 3D mapping of the plume in the tracking performance. Now, we will observe the effect of the user variable trajectory parameters.

Following the same methodology, the tests are first performed in a static plume and will track the entire of the front (simulation set **D.1**).

Starting with the trajectory angle α, it was tested between 20 and 70 degrees on the *Zigzag* and *Alt. zigzag* strategies. The results are shown in Figures A.6 and A.7 in Appendix A.4.

Figures 6.10a and 6.10b indicate that there is a relation between the increase of α and a decrease in the quality of the front prediction.

In regards to the front progression, we can conclude that there is a linear relationship between the angle α and the velocity v_{fp} (Figure 6.10c). In addition, we observe that the faster front progression, v_{fp}, from the *Alt. Zigzag* strategy is not noticed for wider angles, in fact, the plot from the *Alt. zigzag* (dashed lines) seems to converge to the values for the *Zigzag* (solid lines).

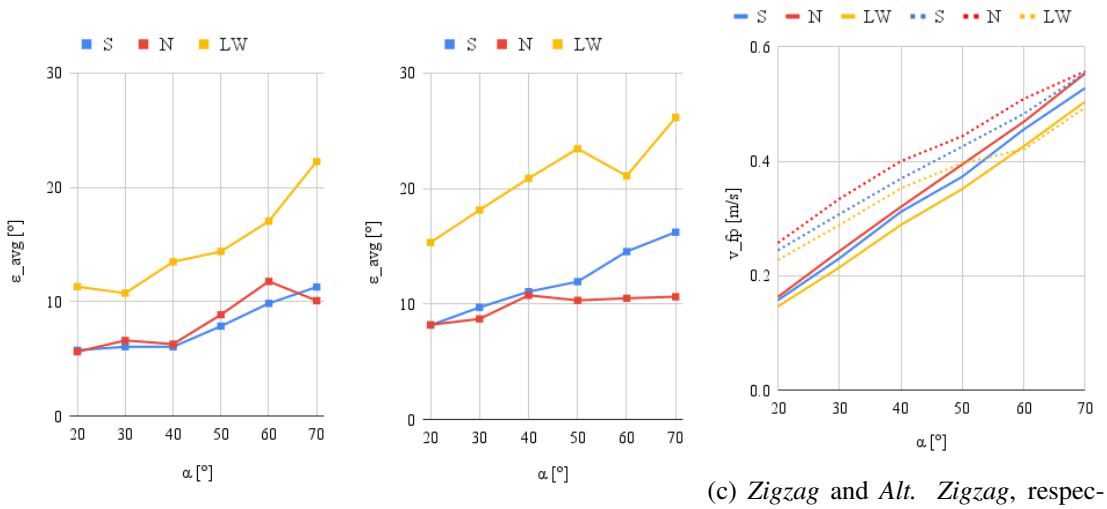
These results show a clear trade-off between front progression and accuracy in the front prediction.

Additionally, it is worth mentioning that angles as high as 70 degrees reduce the robustness of the system, the possibility of losing the front is significantly higher. Although all tests completed the front-tracking, the vehicle has instances of almost not crossing the front.

Altering the distance travelled parallel to the front, when outside the plume, D_{out}, produces similar results. Results shown in Figures A.8 and A.9 in Appendix A.4.

As before, a similar behavior is observed. A linear increase in the front progression is correlated with a worse tracking performance (Figure 6.11).

The effect of D_{in} is very similar to D_{out}, observed in Figure 6.12 (Results shown in Figure A.10).

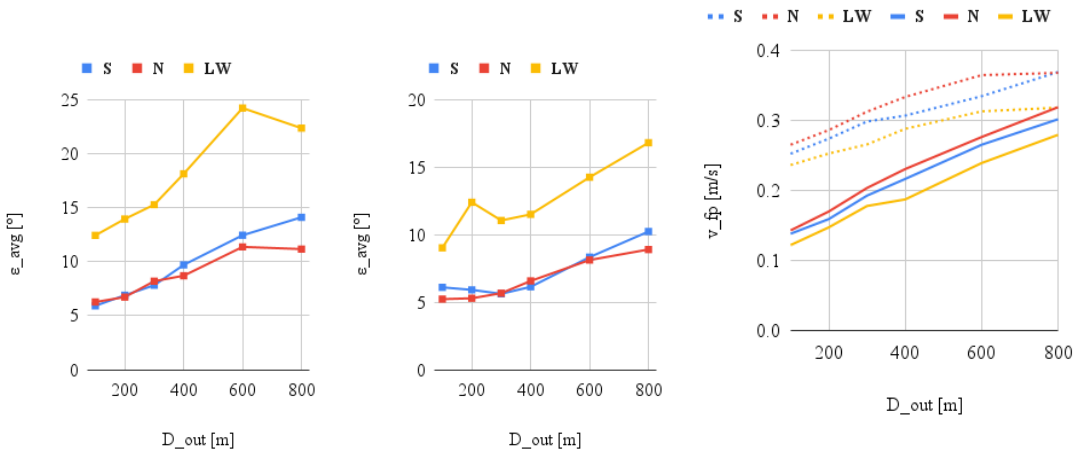


(a) *Zigzag* (b) *Alt. Zigzag* (c) *Zigzag* and *Alt. Zigzag*, respectively, solid and dashed line.

Figure 6.10: Effects of the trajectory angle, α, on the average error, ε_{avg}, and front progression, v_{fp}, with the vehicle performing the *Zigzag* and *Alt. Zigzag* strategies.

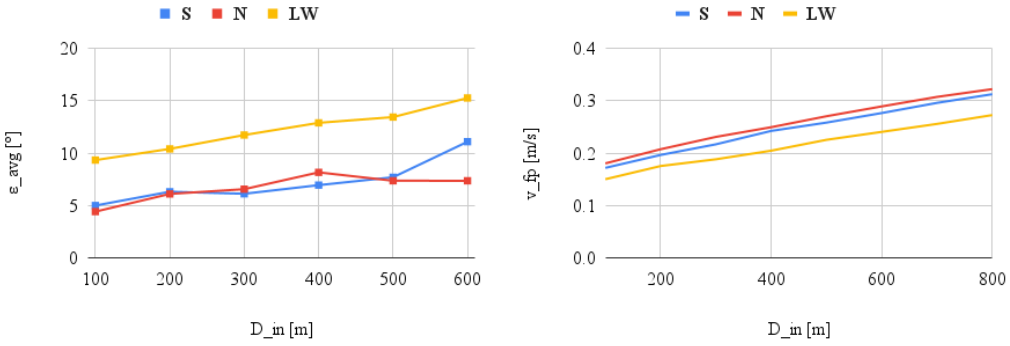
The front progression results, for all cases, show a clear linear behavior, v_{fp} increases as the distance between front crossing points increases as well.

For all the above results, the best prediction of the front is achieved when the variables are at their minimum (i.e. the front crossing points are closer to each other). Since these conditions represent a surface tracking of a static plume, the front is detected with a high level of accuracy, this can be seen in the figures from Appendix A.4, where the crossing points, P_{cross}, are directly on top of the front. Thus, as expected, the best prediction occurs when we have the two points as close as possible. However, these are not the conditions that the vehicle will face on the Douro river



(a) *Alt. Zigzag* (b) *Lawn Mower* (c) *Lawn Mower* and *Alt. Zigzag*, respectively, solid and dashed line.

Figure 6.11: Effects of D_{out} on the average error, ε_{avg}, and front progression, v_{fp}, with the vehicle performing the *Alt. Zigzag* and *Lawn Mower* strategies.



(a) *Lawn Mower* (b) *Lawn Mower*

Figure 6.12: Effects of D_{in} on the average error, ε_{avg}, and front progression, v_{fp}, with the vehicle performing the *Lawn Mower* strategy.

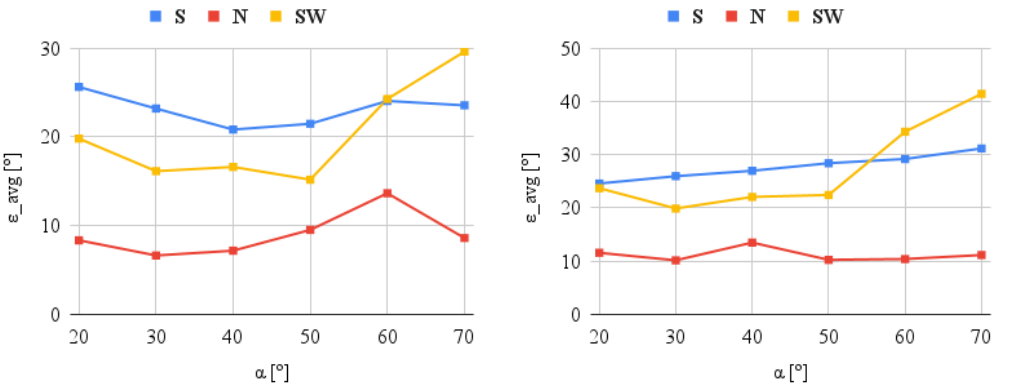
plume. Thus, we will now interpret the same results, but for a 3D tracking of a dynamic plume (simulation set **D.2**), which will present significantly less accuracy in locating the front crossing positions. The results are shown in Appendix A.5.

The results for the front progression are almost identical and present the same linear behavior for all test cases. Thus, we will focus on the results for the error, ε_{avg}.

Conversely, to the front progression results, the average error presents a very different behavior in the previous case, as an upper and almost linear tendency is not verified.

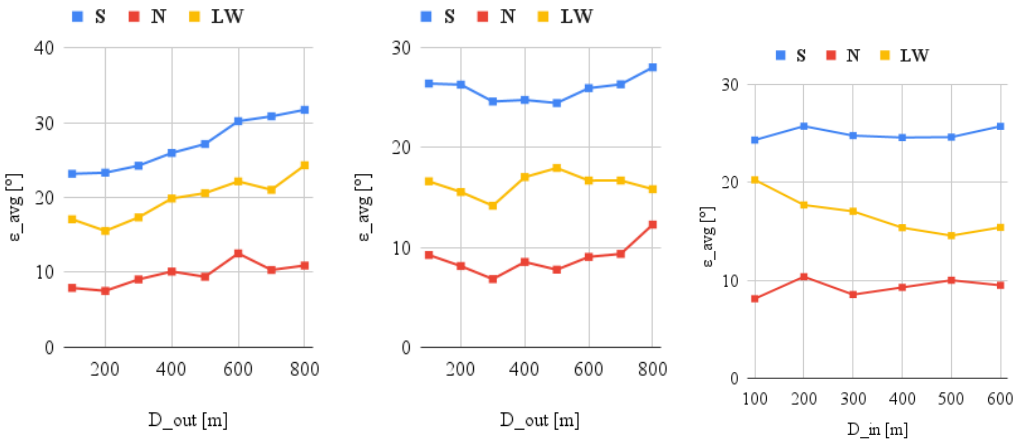
Figure 6.13 illustrates the results for different α values. The best performance is not verified when $\alpha = 20^{\circ}$ but in the interval between $\alpha = [30, 50]^{\circ}$. Thus, the best performance is not guaranteed when the front crossing points are at a minimal distance from each other. Due to the P_{cross} locations not being as accurate as before, the prediction with closer points does not always result in a more accurate description of the front's orientation.

This same conclusion can be made for different values of D_{out} and D_{in} (Figure 6.14). For



(a) *Zigzag* (b) *Alt. Zigzag*

Figure 6.13: Effects of the trajectory angle, α, on the average error, ε_{avg} with the vehicle performing the *Zigzag* and *Alt. Zigzag* strategies on a 3D and dynamic plume.



(a) *Alt. Zigzag* (b) *Lawn Mower* (c) *Lawn Mower*

Figure 6.14: Effects of the distances, D_{in} and D_{out}, on the average error, ε_{avg} with the vehicle performing the *Alt. Zigzag* and *Lawn Mower* strategies on a 3D and dynamic plume.

the distance outside the plume, the best results are observed for values in the interval $D_{out} = [200, 300]\,m$. For D_{in} the results do not present a clear optimal value for all plumes. The *LW plume* has the best performance for $D_{in} = 500\,m$, conversely, for *N plume* the best result is achieved with $D_{in} = 100\,m$. Nevertheless, the configuration that would give the best overall results for all plumes, seems to be about $D_{in} = 300\,m$, although a clear conclusion cannot be made for this set of results.

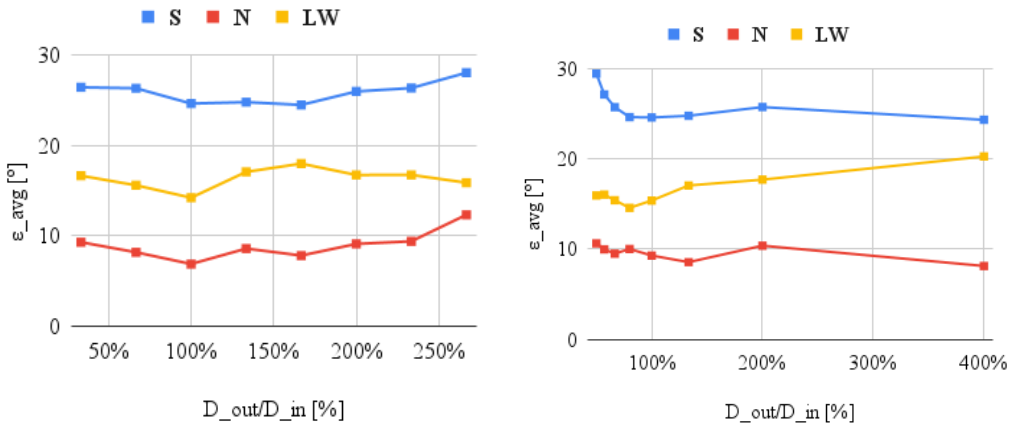
In Figure 6.15, the previous results were reorganized to reflect the ratio between the distance performed outside and inside the plume in the *Lawn Mower* strategy. When D_{in} is fixed at $300\,m$ and we let D_{out} take multiple values, the ratio that provides the best overall result for all plumes is $\frac{D_{out}}{D_{in}} = 100\%$. For the case where $D_{out} = 400\,m$ the results are not as unanimous. Although one observation can be made, the worst results are verified when $\frac{D_{out}}{D_{in}} < 50\%$.

This set of results is not comprehensive enough to conclude with certainty an optimal ratio. Nonetheless, we can observe that the overall best result occurs when D_{out} is larger or equal than D_{in} (i.e. $\frac{D_{out}}{D_{in}} > 100\%$).

After evaluating the effect of the trajectory parameters on the tracking performance and front propagation, the following values are proposed to be first tested in a field mission.

θ	15 °
z_{max}	10 m
α	35 °
D_{out}	400 m
D_{in}	300 m
$[D_{min}; D_{max}]$	[500;700] m
v	1 m/s

Table 6.11: Proposed parameters for a field mission on the Douro River plume.



(a) $D_{in} = 300$m (b) $D_{out} = 400$m

Figure 6.15: Effects of the ratio between the parallel transects to the front outside and inside the plume, $\frac{D_{out}}{D_{in}}$, on the average error, ε_{avg}, with the vehicle performing the *Alt. Zigzag* strategies on a 3D and dynamic plume.

To more carefully predict the effects of the parameters in the front progression, we devised the following method to calculate the theoretical front progression, v_{fp}, for a set of trajectory parameters.

If we assume a straight front and an ideal behavior for the tracking trajectory (i.e. $\varepsilon = 0°$), we can calculate the total distance travelled by the vehicle in one cycle of the trajectory, $D_{Trajectory}$, and the respective advancement made on the front in that cycle, D_F. With the vehicle's horizontal velocity, $v_{bH} = v \cdot \cos\theta$, we calculate the time, $T_{Trajectory}$, it takes for the agent to travel the distance $D_{Trajectory}$,

$$T_{Trajectory} = \frac{D_{Trajectory}}{v \cdot \cos\theta}. \tag{6.5}$$

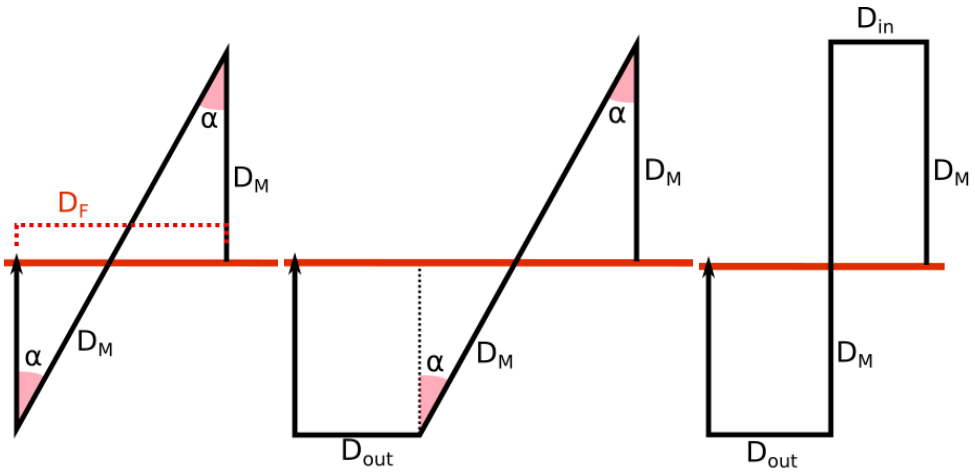
The time it takes the vehicle to progress along the distance of the front D_F is time $T_{Trajectory}$. Thus, we can define the velocity of the front progression, v_{fp}, as,

$$v_{fp} = \frac{D_F}{T_{Trajectory}} \tag{6.6}$$

To calculate $D_{Trajectory}$ we must know the distance travelled after the front, D_{front}, since this distance can fluctuate between D_{min} and D_{max} we considered the average distance D_M,

$$D_M = \frac{D_{min} + D_{max}}{2}. \tag{6.7}$$

Figure 6.16 illustrates a schematic of the three strategies.



(a) *Zigzag* (b) *Alt. Zigzag* (c) *Lawn Mower*

Figure 6.16: Schematic of one cycle of each trajectory. The front is represented in red.

After defining $D_{Trajectory}$ and D_F for all three different trajectories, we arrive at the following front progression formulas for the respective strategies,

$$\begin{aligned} v_{fp_{Zigzag}} &= \frac{\sin\alpha + \tan\alpha}{2 + \frac{1}{\cos\alpha} + \cos\alpha} \cdot v \cdot \cos\theta, \\ v_{fp_{Alt.Zigzag}} &= \frac{D_M(\sin\alpha + \tan\alpha) + D_{out}}{D_M(2 + \frac{1}{\cos\alpha} + \cos\alpha) + D_{out}} \cdot v \cdot \cos\theta, \\ v_{fp_{LawnMower}} &= \frac{D_{in} + D_{out}}{4D_M + D_{in} + D_{out}} \cdot v \cdot \cos\theta. \end{aligned} \tag{6.8}$$

To understand if these formulas correlate to the experimental results, we calculate the front progression for the default parameters and compare the results to the simulations performed so far. The results are shown in 6.17. The simulation results are the average front progression for the three different plumes. The calculated values confirm that the front progression is faster for the

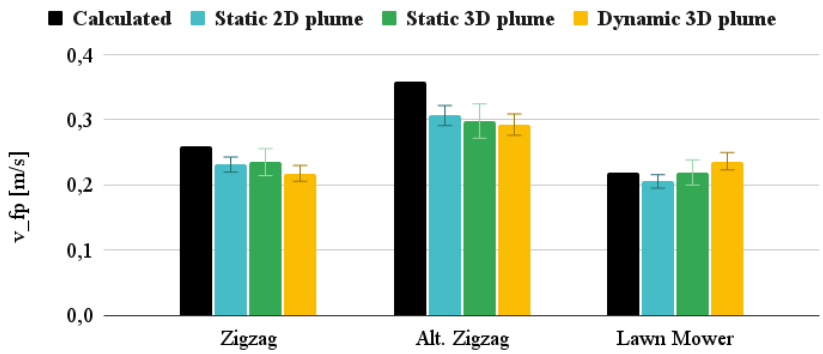


Figure 6.17: Front progression, v_{fp}, calculated for the default parameters and compared to the experimental results.

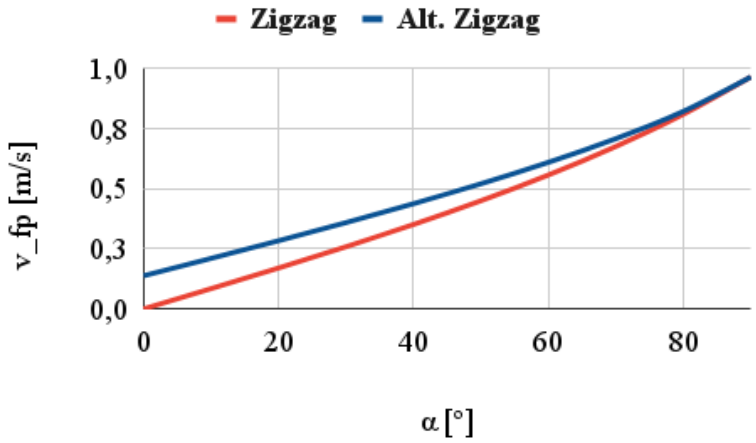


Figure 6.18: Effect of the angle α in the front progression, v_{fp}.

Alt. Zigzag and slower for the *Lawn Mower*. This same observation was made for the results in section 6.3.

Additionally, we observe that the calculated values correlate well with the experimental results. The experimental results show a slower v_{fp}. This variation is expected since the computed value assumes a straight front and an ideal tracking, which is not the case for the plume's front. Nonetheless, this shows that the computation can provide a useful estimation for the expected front progression of the mission.

Furthermore, we replicated the results for different parameters of α, D_{out} and D_{in}.

The results for different angles are shown in Figure 6.18. We observe a significant faster progression for the *Alt. Zigzag* for narrower angles. This difference to the *Zigzag* strategy is lost for wider angles and converges for $\alpha = 90^{\circ}$, where the front progression is equal to the horizontal velocity, v_{bH}. This behavior correlates to the plot seen in Figure 6.10c from the experimental results. From the previous analysis, we can confirm that the convergence of the two plots is indeed expected. This further proves that this calculation accurately describes the behavior of the trajectories.

The results when we let D_{out} take multiple values are plotted in Figure 6.19. The same behavior was observed for the experimental results (in Figure 6.11c).

The behavior for D_{in} is the same as D_{out}, since we are assuming a straight front the parallel distance travelled inside and outside of the plume has the same effect.

Since this calculation proved to be a good estimation of the true front progression of the algorithms, we can apply equations (6.8) to predict the expected length of the mapped front, D_{mapped}, that a certain set of parameters yields for a mission time, T_M. Thus,

$$D_{mapped} = v_{fp} \cdot T_M \tag{6.9}$$

Table 6.12 exemplifies this computation for a 6 h mission time ($T_M = 21600\ s$) applied to the default parameters used in the simulations.

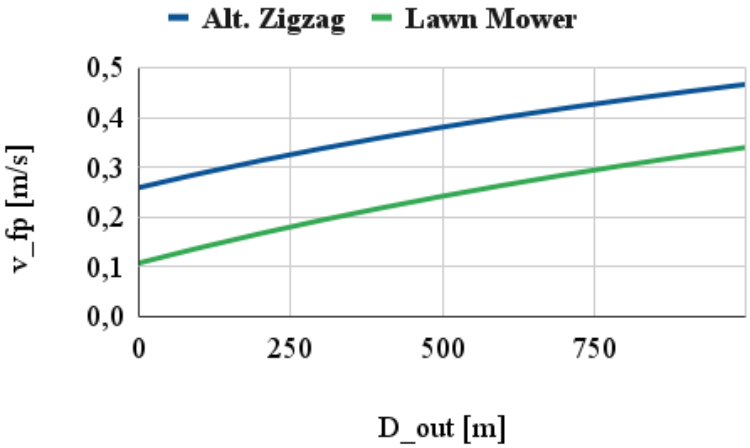


Figure 6.19: Effect of D_{out} in the front progression, v_{fp}.

Strategy	v_{fp} [m/s]	D_{mapped} [km]
Zigzag	0.26	5.59
Alt. Zigzag	0.36	7.76
Lawn Mower	0.22	4.71

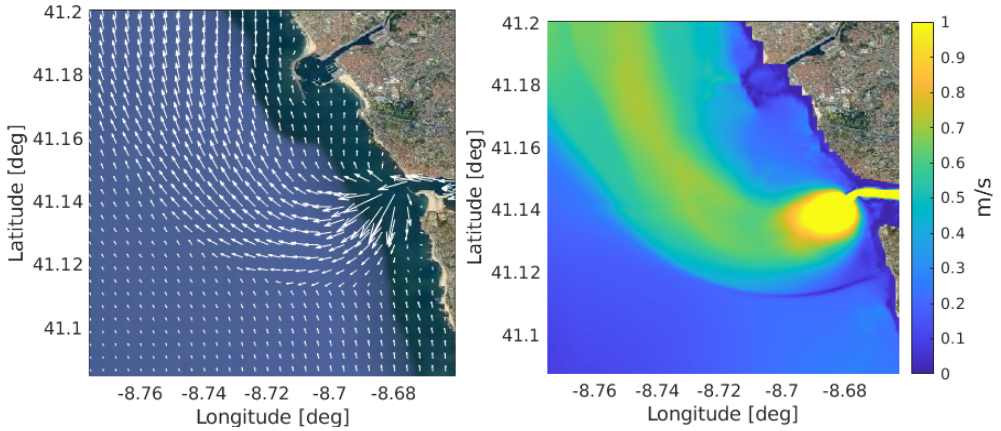
Table 6.12: Estimation of the total length of mapped front, D_{mapped}, for the default parameters defined in Table 6.3.

If a mission requires the mapping of a certain distance of the front, this computation will indicate if such a target is possible to be met for the current chosen trajectory. Thus, this method proves to be useful in the planning phase of the mission.

6.7 Trajectory deviation caused by ocean currents

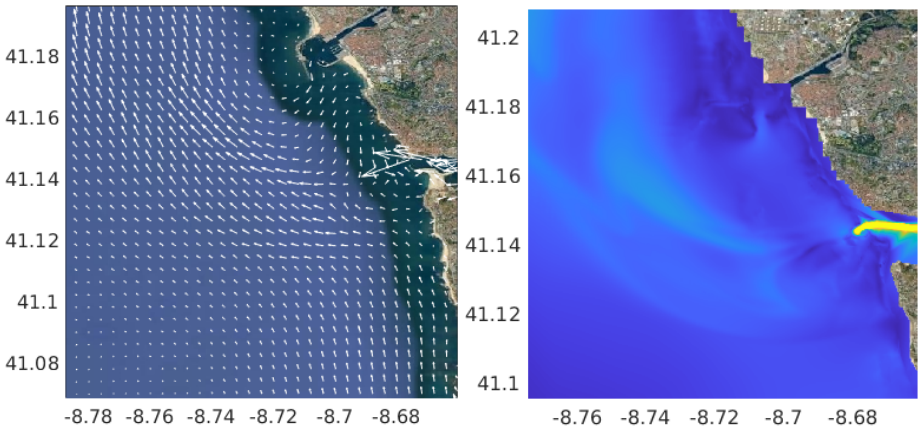
Ocean currents influence the trajectory of the AUV and consequently will also affect the tracking performance. To study the possible interference of the water velocity, we must understand its general behavior in the plume region.

One important characteristic in the coastal ocean is that water velocity tends to decrease over depth. Figure 6.20 shows the direction and velocity of the field at multiple depths in the *S* plume. The surface velocity is much higher at the surface than at the subsequent depths. Since the AUV is continuously altering its depth, the effect of the ocean currents will be mitigated by the movement at greater depths where the water has less velocity. It is also noticeable that there is a clear change in the vector direction when inside the plume. This effect also becomes less prominent as depth increases, where the velocity tends to a similar direction to the surrounding ocean water. This effect is seen in figure 6.20e. The behavior described here is observed in all tested plumes.



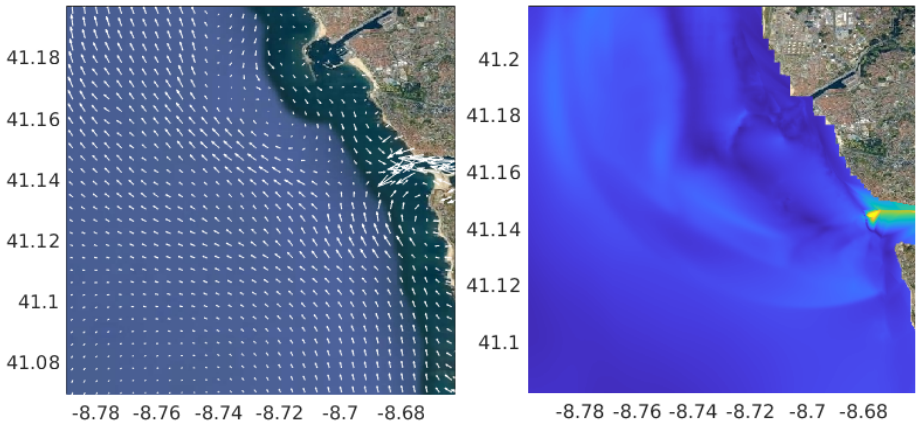
(a) Direction at $z = 0$ m

(b) Velocity at $z = 0$ m



(c) Direction at $z = 5$ m

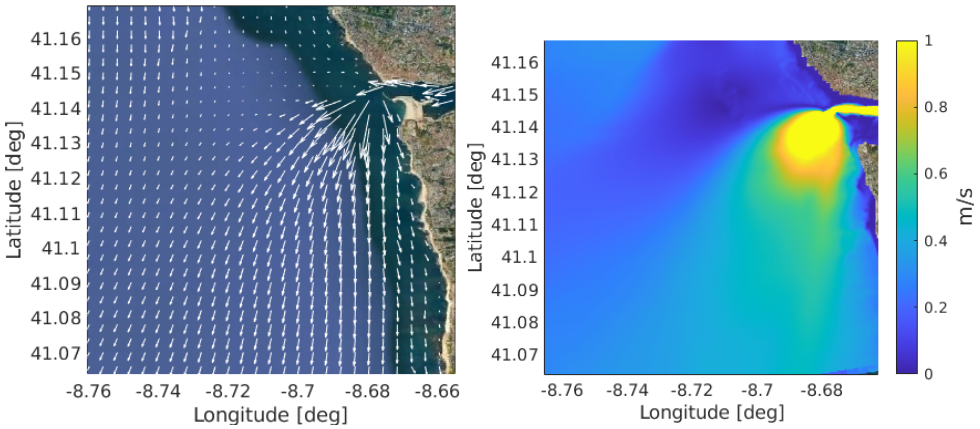
(d) Velocity at $z = 5$ m



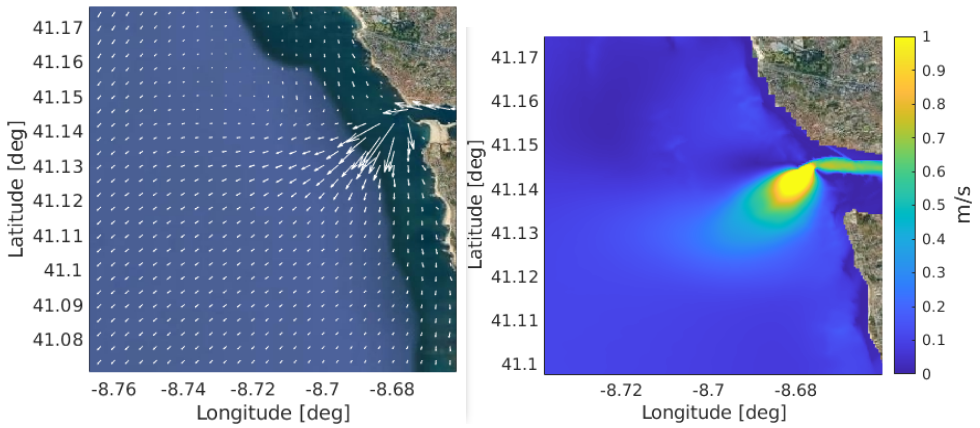
(e) Direction at $z = 10$ m

(f) Velocity at $z = 10$ m

Figure 6.20: Direction and velocity of the ocean currents at 04:30 May 9$^{\text{th}}$ 2016 (*S plume*). The data is shown at three depths: 0 m, 5 m and 10 m.



(a) Direction at $z = 0$ (b) Velocity at $z = 0$



(c) Direction at $z = 0$ (d) Velocity at $z = 0$

Figure 6.21: Direction and velocity of the ocean currents at 13:00 May 15th 2016 (*N plume*) and 21:30 May 2nd 2017 (*LW plume*), respectively, (a,b) and (c,d). The data is shown at surface level.

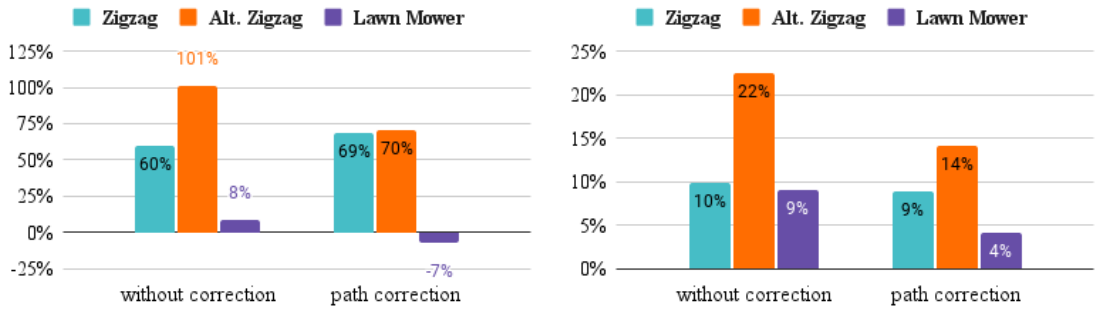
Figure 6.21 shows the same data as before for the two other plume's shapes at surface level, respectively, *N* and *LW* plumes. In all three cases, we see an expected peak velocity near the river mouth. The *S plume* has a higher velocity throughout the plume, conversely, the *LW plume* has a lower velocity in the front region. It is expected that the effect of the water velocity to be more noticeable on the *S* and *N* plumes.

To understand this effect the test made for the 3d-tracking was affected by the ocean currents and by the system of path correction (simulation set **D**).

Figure 6.22 shows the results for the *N plume*. The influence of the ocean currents on the average error is evident. The *Alt. Zigzag* strategy saw a two-times increase on ε_{avg}. The path correction diminished this effect. However, in the *Zigzag* case, such improvement was not verified. The *Lawn Mower* case was the least influenced by the water velocity with an increase of only 8%.

The front progression increased in all strategies and the inclusion of the path correction method mitigated the results in all three cases. However, the decrease in the *Zigzag* case was only 1%. The path correction did not have the desired effect for the *Zigzag* strategy in this plume.

Figure 6.23 illustrates the results for the *S plume*. It displays a similar behavior to the previous



(a) Effect on ε_{avg} in the *N plume*.

(b) Effect on v_{fp} in the *N plume*.

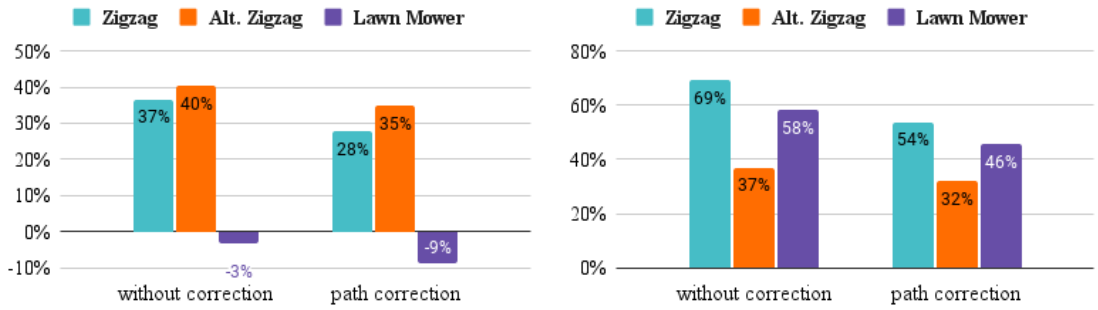
Figure 6.22: Effect on the tracking performance in the *N* plume compared to the results where water velocity is assumed to be zero.

plume. However, the path correction improves the tracking performance for all strategies. This is a significant result in favor of the utility of this method, since this test is being performed on the plume with the faster water velocities. Once again the *Lawn Mower* strategy avoids the influence of the water velocity in a very successful manner.

With the *LW* plume, we observe the least effect on the average error (Figure 6.24). Among the three plumes, this case was the least influenced on the front progression as well. Nevertheless, a positive effect of the path correction method is verified for all three trajectories. As expected the *LW* plume with weaker ocean currents is where the tracking performance is least affected.

We can conclude that the ocean currents have a substantial influence on the tracking mission. Overall the path correction method mitigates the effects of the water velocity. However, it does not completely correct the trajectory of the vehicle, and with even some cases not producing an improvement on the performance. We can safely assume that this method is opposing the water currents since the results for the front progression are clear and we can see a positive relation in all tested cases.

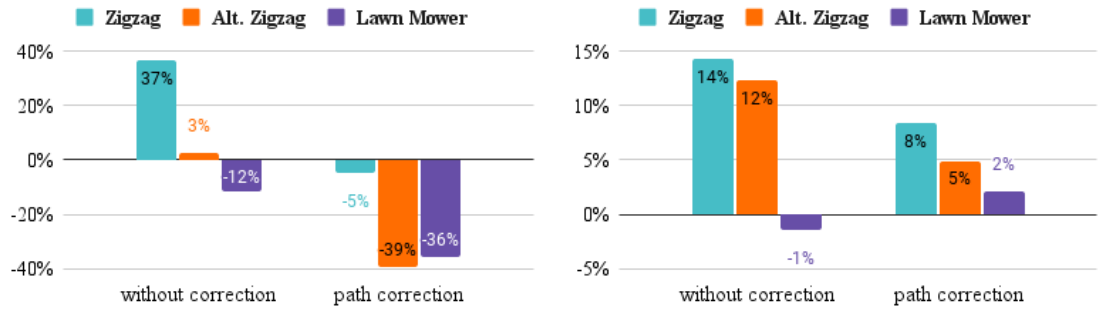
One promising result is the fact that the *Lawn Mower* strategy seems to be the least influenced



(a) Effect on ε_{avg} in the *S plume*.

(b) Effect on v_{fp} in the *S plume*.

Figure 6.23: Effect on the tracking performance in the *S* plume compared to the results where water velocity is assumed to be zero.



(a) Effect on ε_{avg} in the *LW plume*.

(b) Effect on v_{fp} in the *LW plume*.

Figure 6.24: Effect on the tracking performance in the *LW* plume compared to the results where water velocity is assumed to be zero.

by the water dynamics. With these conditions, this strategy proves to be the ideal deployment algorithm.

The currents will usually have a direction parallel to the front. Thus, the vehicle moves perpendicular to the water direction when crossing the front with the *Lawn Mower* strategy. This simulation environment did not account for the vehicle's hydrodynamics, thus, in these transects, the AUV's trajectory could be influenced by the water exerting a perpendicular force in the vehicle's body, that would further detour the agent from its ideal trajectory. This effect is not computed in this model, therefore, conclusions about which method achieves the best tracking performance can not be made with certainty. Nonetheless, the large difference in results, points to the *Lawn Mower* being indeed the best strategy, with the expectation of this difference being mitigated when the effect of drag is taken into account.

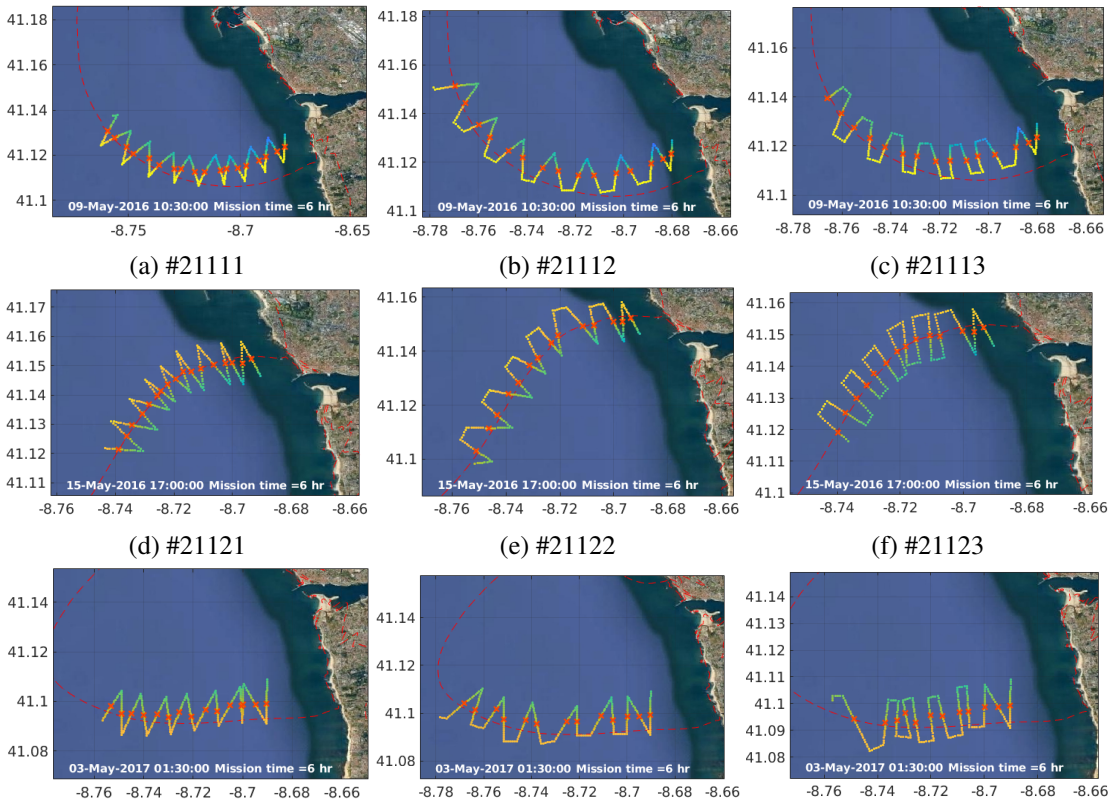
With the inclusion of ocean currents and the path correction in the model, we can present the final results where the simulation model is more realistic. We expect these results to be similar to the tests in the field. Figure 6.25 illustrates the AUV trajectory for the final results.

Starting with the average error, ε_{avg}, the results are positive, especially on slower plumes as was the case for the *N* and *LW* plumes (Figure 6.26). The faster propagating *S* plume presents the highest error by a large margin in all trajectories, we can conclude that the velocity of the front is the most important element affecting the tracking performance.

The mission has multiple transects with optimal performance, providing quality profiles of the front for oceanographic studies (Figure 6.27). The results for the *N* and *LW* plumes are very promising with almost all transects being classified as *optimal*. The worst performance on the *S* plume is evident in these results, with the *Zigzag* and *Alt. Zigzag* not able to perform a single transect in optimal condition. However, the *Lawn Mower* was able to perform half of its transects with an error of less than 20°.

These results favor the use of the *Lawn Mower* strategy. It produces the largest number of optimal transects in all conditions and it is the strategy that is less affected by the ocean currents.

The influence of the ocean currents is a critical problem in ocean surveys, therefore, the success of these results is a significant step in validating the system for field deployment.

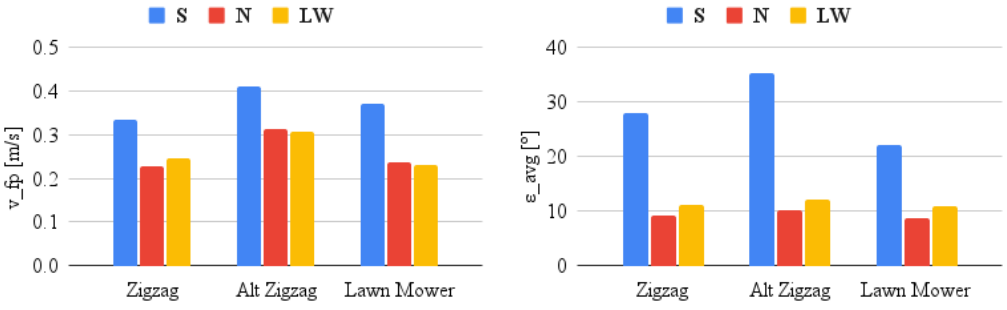


(a) #21111 (b) #21112 (c) #21113

(d) #21121 (e) #21122 (f) #21123

(g) #21131 (h) #21132 (i) #21133

Figure 6.25: Results of tracking a dynamic plume with ocean currents and the path correction method. From simulation set **E**.



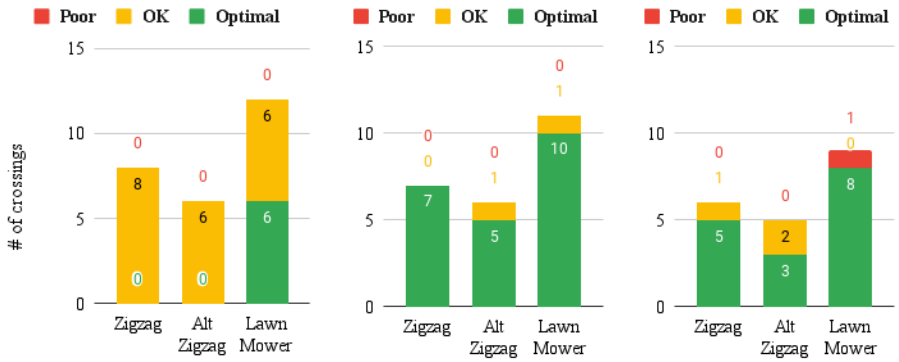
(a) (b)

Figure 6.26: Front progression, v_{fp}, (a) and average error, ε_{avg}, (b) results for the 3D tracking and mapping of a dynamic plume with active ocean currents.

6.8 Plume mapping

The results shown so far have been focused on tracking performance. In this section, we present the results of the plume mapping for simulation set **D** (presented in Figure 6.25).

Figure 6.28 plots the salinity values measured at the surface for a *Zigzag* tracking of the S plume. The result demonstrates that the vehicle maintains itself in the δ-band defined by the



(a) *N plume* (b) *S plume* (c) *LW plume*

Figure 6.27: Crossings distribution for the 3D tracking and mapping of a dynamic plume with active ocean currents.

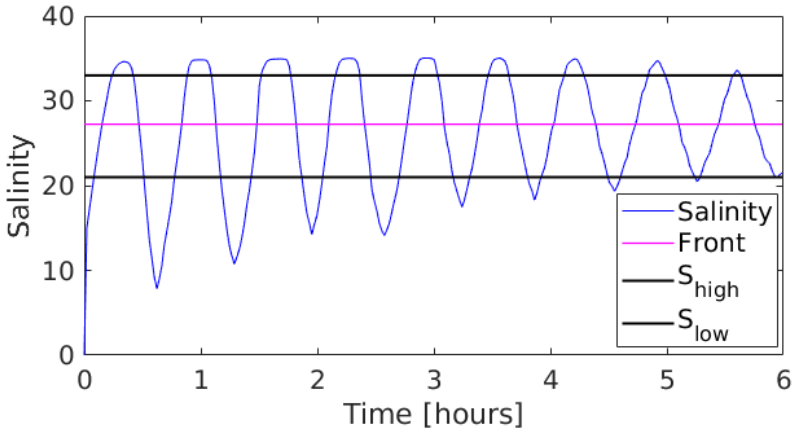


Figure 6.28: Measured surface salinity with the *Zigzag* strategy in the *S plume*.

values S_{high} and S_{low}. The salinity crosses these values when the minimal distance after the front has not yet been met. As expected, we also verify that the front is crossed with every transect.

The totality of the salinity values measured for all positions of the vehicle is presented in Figure 6.29, where the values are plotted by time and depth, representing the first crossing of the front on the *S plume* using the *Zigzag* strategy. This image provides evidence that the system is capable of retrieving information of the plume in respect to the z axis. From this sample, we can clearly have a good idea of the characteristic of the plume, such as depth (about 3 meters in this instance) and the gradient profile. Additionally, this result is very similar to the field experiments already performed on the Douro plume [2]. This indicates that the simulation environment was capable of replicating with some success the field results. This is, of course, not concluding evidence that the system will behave with this much success in a field setting. Since so far no such experiment was performed, conclusions will only be made after concrete evidence of successful field results are achieved.

Complementing the previous image, Figure 6.30a plots the entirety of the 6-hour mission. These are the desirable results of the full plume mapping, all instances of plume exits and entries

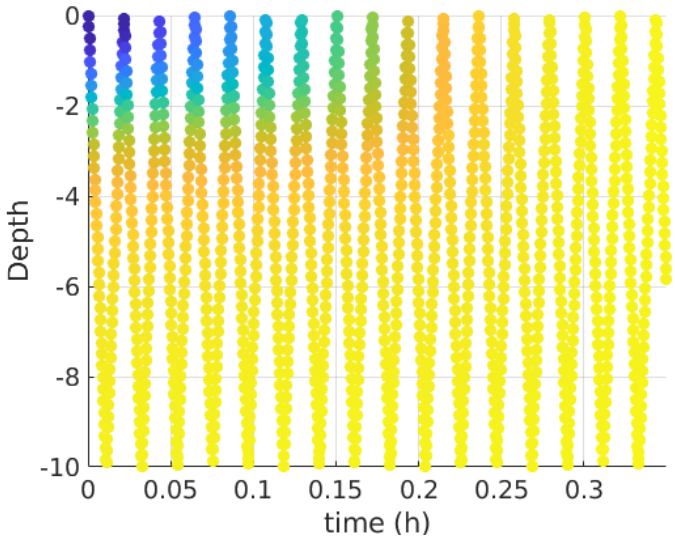


Figure 6.29: Sampled salinity data in a front crossing transect.

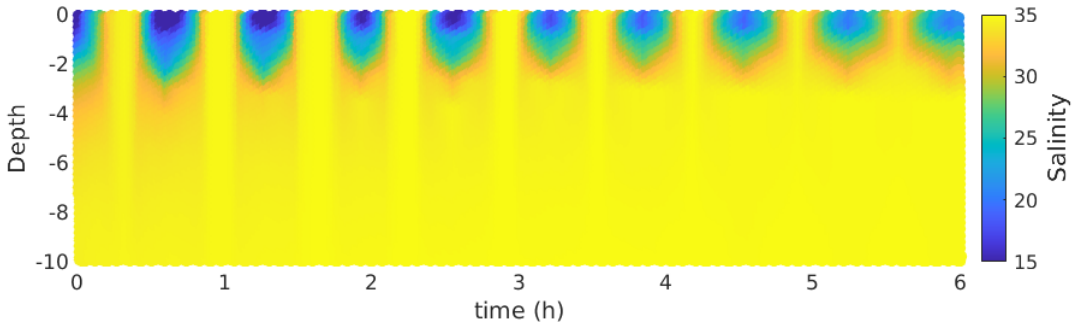
are well defined in this plot, additionally for the plume in question, we know that it has a steep gradient close to the coast at the starting position of the mission and that this gradient would faze out along the rest of the plume. This is also observed in these results wherein the final crossings we get a more smooth color gradient and even a larger plume width. Although this plume shape has the worst results in tracking performance it is still successful in retrieving a general plume characterization.

Figure 6.30a demonstrates the results achieved with a *Zigzag* trajectory and Figure 6.30b with the *Lawn Mower*. Some main differences can be highlighted, due to the extra parallel tracks present in the second strategy, we observe that the vehicle spends more time inside and outside of the plume, consequently, it has fewer crossing points. In the first case, we count 10 plume entries and on the second 8 entries. However, the second case has a better quality of transects, this improvement could be significant when analysing the data in more rigorous terms.

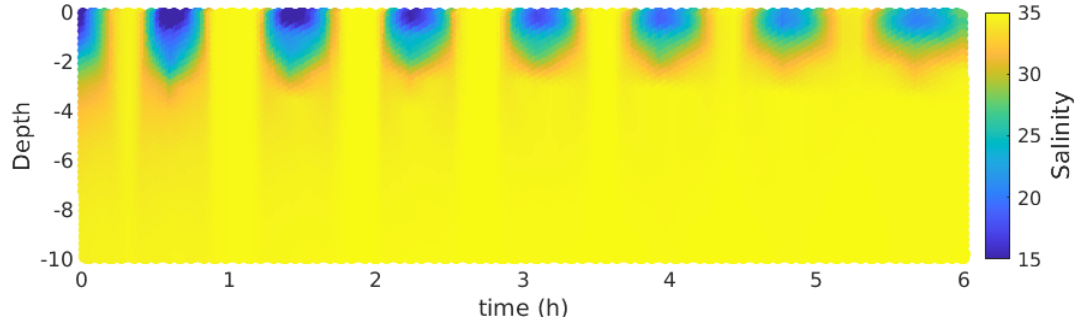
Figure 6.31 plots the vertical profile of the *N* and *LW* plumes from a mission with the *Zigzag* strategy. The two plumes show the smoothest gradient of the front and, consequently, a wider front in the *xy* plane. With this projection is evident that the *LW* plume has a very shallow plume, with some instances being lower than 1 meter. Plumes with such profile prove to cause more difficulties on the tracking algorithm, as was shown with the results and trajectories for this plume, which show significantly more irregularity and less precision in locating the front positions.

The salinity plots for all 9 missions of set **D** can be found in Appendix B.1.

Overall, the mapping results of the plumes are very successful. If these same results can be replicated in-field testing we would prove to have a fully capable system of mapping and tracking the Douro river plume.

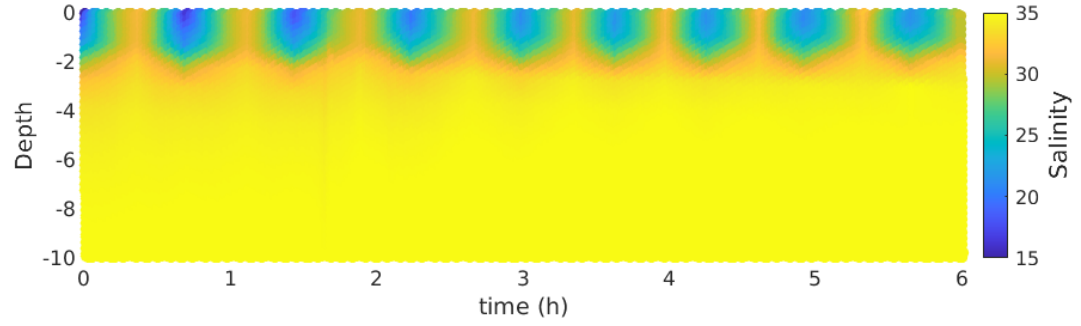


(a) *Zigzag*

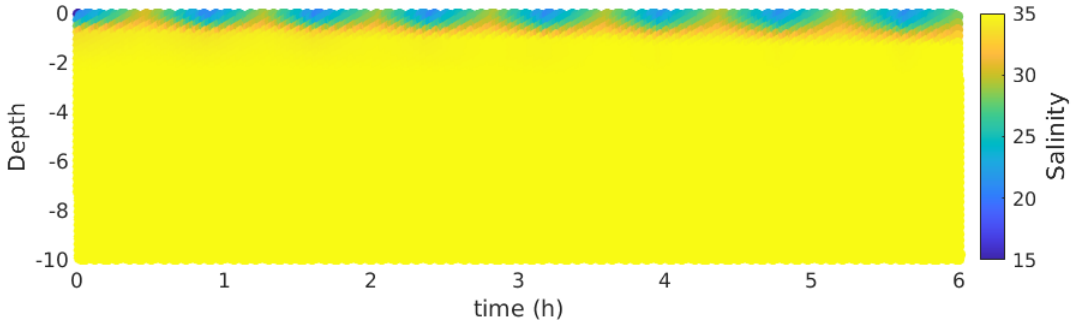


(b) *Lawn Mower*

Figure 6.30: Sampled salinity data for the entirety of the mission in the *S* plume.



(a) *N plume*



(b) *LW plume*

Figure 6.31: Sampled salinity data for the entirety of the mission in the *N* and *LW* plumes with the *Zigzag* strategy.

Chapter 7

Conclusions and future research

The Douro river plume has been studied in the past with a variety of methods due to the importance of understanding its behavior and the possible effect on the surrounding environments. Building on the previous work done by Laboratório de Sistemas e Tecnologia Subaquática (LSTS) to study the plume with an *Autonomous Underwater Vehicle* (AUV), we had as main objectives of this thesis to improve the tracking performance and successfully map the 3D structure of the plume.

To improve the performance of the front-following algorithm, we proposed an adaptive sampling method that measures salinity data with a CTD sensor and adapts the vehicle trajectory in real-time to autonomously follow the plume. This was achieved through a linear estimation of the front's orientation, which enables the vehicle to fully adapt its trajectory to the current state of the plume. This proved to be an improvement to the previous implementation, where the vehicle did not estimate the front's direction and had a pre-determined direction of travel.

With data from the DELFT3D model, that accurately represented the Douro plume's characteristics, we developed a simulation environment to show that our method was successful in following the front. Additionally, three main strategies (*Zigzag*, *Alt. Zigzag*, *Lawn Mower*) were devised and all were demonstrated to follow the plume's front with success.

We validated the three strategies for a surface and 3-dimensional plume, static and dynamic, and even with the effect of ocean currents.

To retrieve the best possible scientific data from the mission, we had the requirement of traversing the front in a perpendicular direction for at least half the front crossings, this would result in ideal data profiles of the plume for posterior study. We show that the *Zigzag* and *Alt. zigzag* are capable of meeting this requirement with a maximum average error of about 30° for fast propagating plumes, and 15° for slower-moving plumes. We consider these results as successful, as it is an overall improvement to the previous implementation, although, for fast plumes, this result does not always guarantee an optimal retrieval of data. The *Alt. zigzag* proved to be the trajectory that provided the fastest front progression. However, a faster front progression was correlated with worse tracking performance. We achieved the best results for the *Lawn Mower* strategy. This trajectory crosses the plume in a perpendicular direction for all transects, providing the best efficiency for the system (i.e. number of optimal transects over the total number of front crossings).

Additionally, even for a fast propagating plume, this strategy always registered optimal transects of the front, which can not be said from the other two strategies. This method also proved to be the best at minimising the effect of ocean currents in the tracking performance. We also shown the path correction method implemented by the DUNE software to mitigate the effect of the ocean currents for the majority of tests. Once again, the *Lawn Mower* strategy showed the best results, since with the application of this method the performance was improved in all tests.

All trajectories are described with user-defined parameters. Thus, the main characteristics of the vehicle's trajectory can be easily modified to meet the requirements of the mission. We provided an analysis of these parameters and documented their effects on the tracking performance. Additionally, we proposed the ideal parameters for a first tracking of the Douro river plume based on those results.

All strategies were capable of providing a characterization of the plume in 3 dimensions, such as plume depth and gradient profile. Thus, completing the second main objective of this thesis.

If the results shown in simulations are applicable in field experiments, the developed method can be considered to be highly successful.

The approach developed was implemented on the LSTS toolchain. The system was validated in the simulation environment provided by the toolchain, thus demonstrating that this approach is functional in the vehicle's hardware. Hence, we have a control algorithm implemented on the DUNE software ready to be tested on the field.

In future work, we hope to deploy the AUV on the Douro estuary and validate these results experimentally. And concluding with certainty the success of this approach. Multiple tests with different strategies can be executed. We hope to test the method for assessing the propagation of the front, as well as multi-agent missions.

Additionally, if the mission is successful with this method, more complex methods of predicting the front could be explored in the hope of accomplishing an improved tracking performance. If the approach for measuring plume propagation proves to be a reliable method in a field setting, the propagation velocity of the front could be implemented on the algorithms to improve the front estimation. Finally, extensions to cooperative mapping and tracking by multiple vehicles will also be studied.

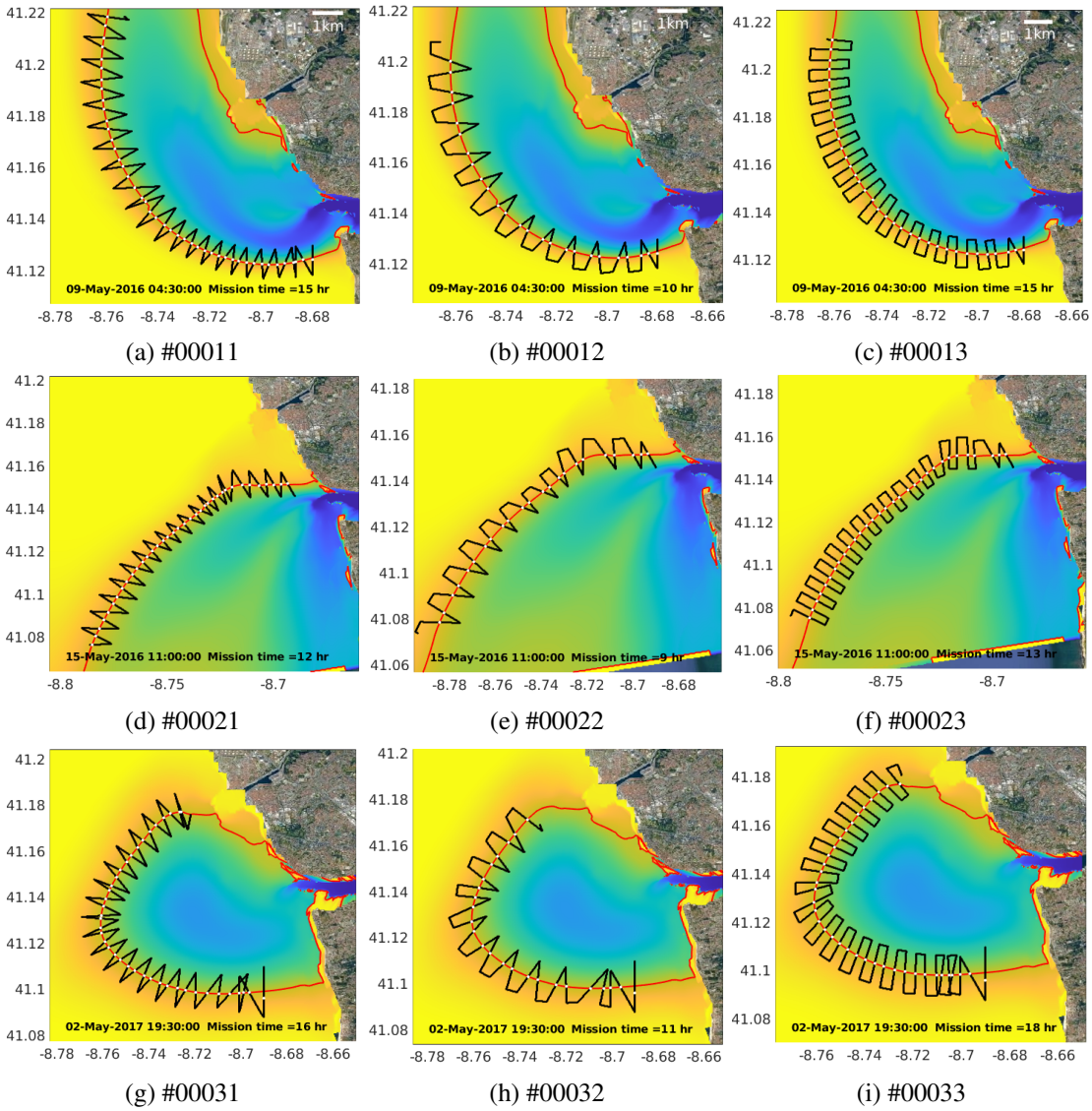
In addition, due to the adaptability and customization of the strategies, we hope to study the success of this method in other plumes and marine features with similar characteristics to the Douro river plume.

In conclusion, the main goals of this thesis were met. Strategies for tracking and mapping the Douro river plume with an AUV were accomplished and validated with a comprehensive and accurate model of the plume. The vehicle's software is ready to be tested and we hope to achieve similar results in a field setting, resulting in meaningful data for oceanographic studies of the Douro River plume.

Appendix A

Test Results

A.1 Surface tracking of a static plume



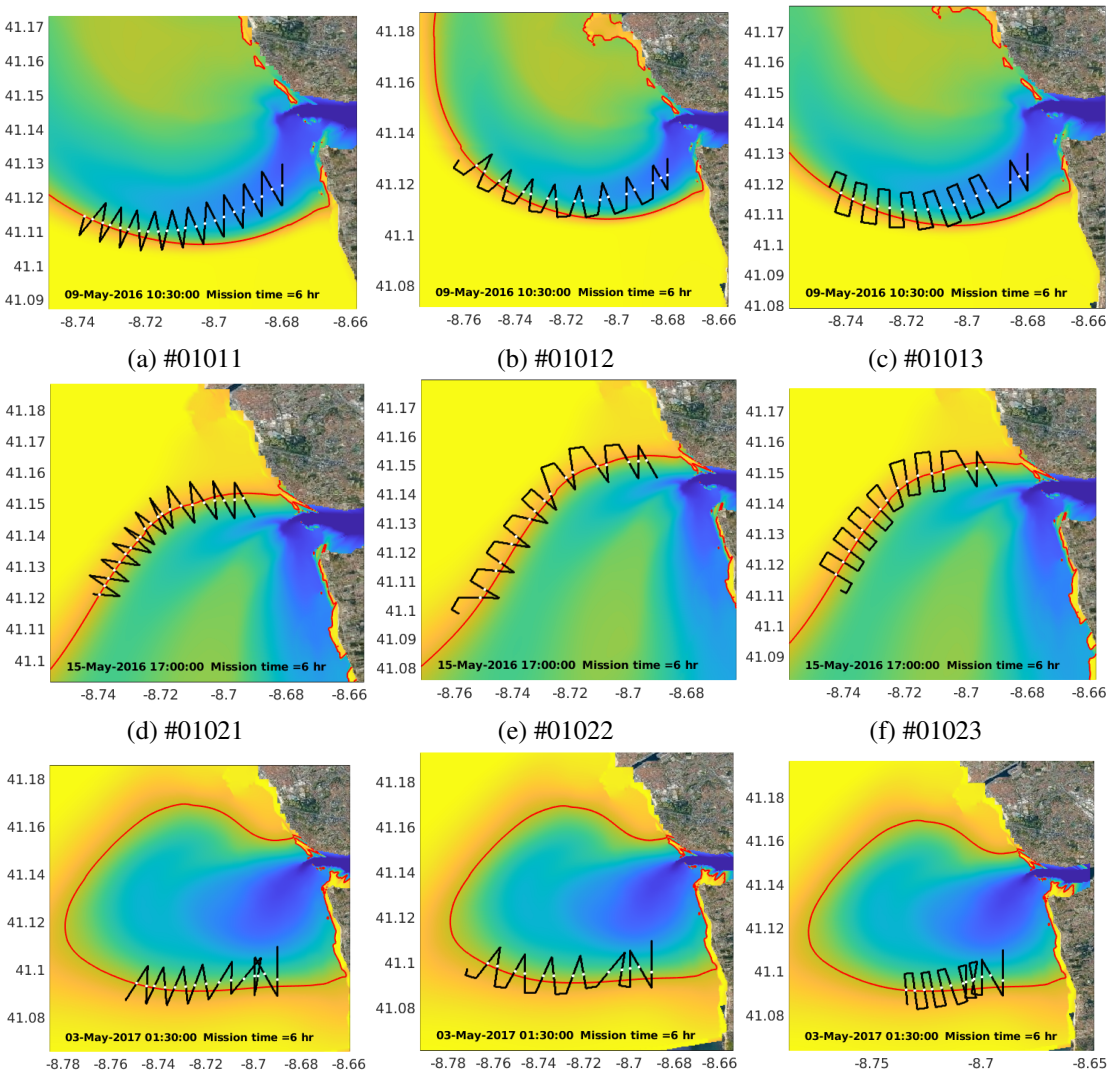
(a) #00011 (b) #00012 (c) #00013

(d) #00021 (e) #00022 (f) #00023

(g) #00031 (h) #00032 (i) #00033

Figure A.1: Test results for the surface tracking of a static plume. Simulation set **A**.

A.2 Surface tracking of a dynamic plume

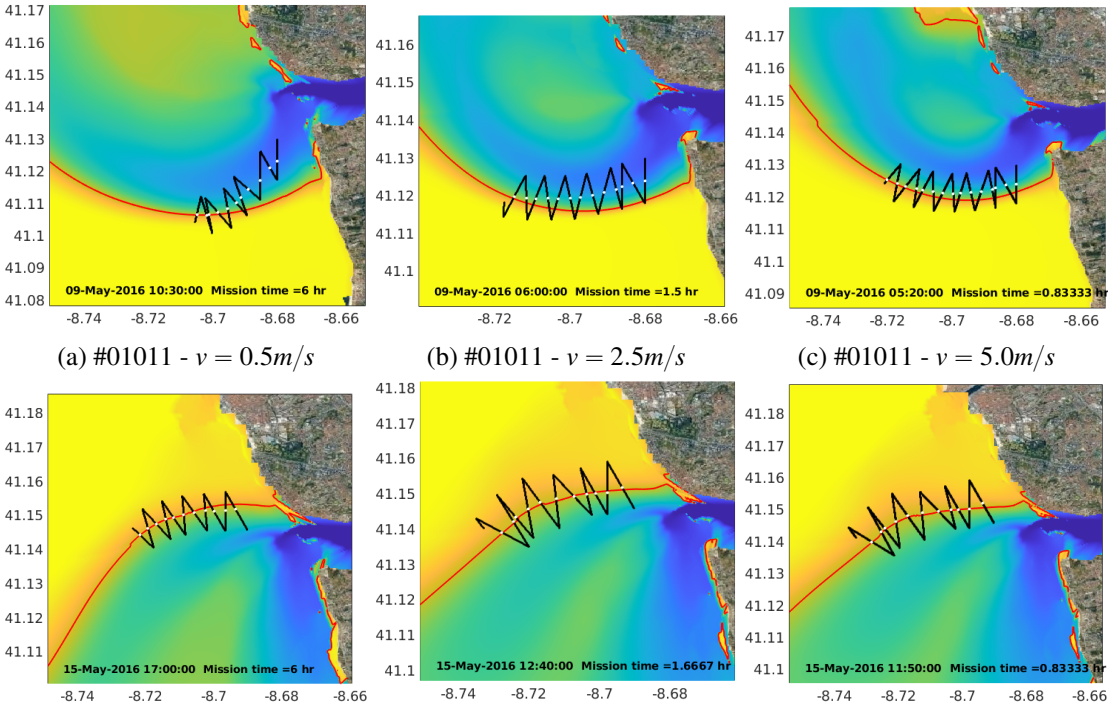


(a) #01011 (b) #01012 (c) #01013

(d) #01021 (e) #01022 (f) #01023

(g) #01031 (h) #01032 (i) #01033

Figure A.2: Test results for the surface tracking of a dynamic plume. Simulation set **B.1**.

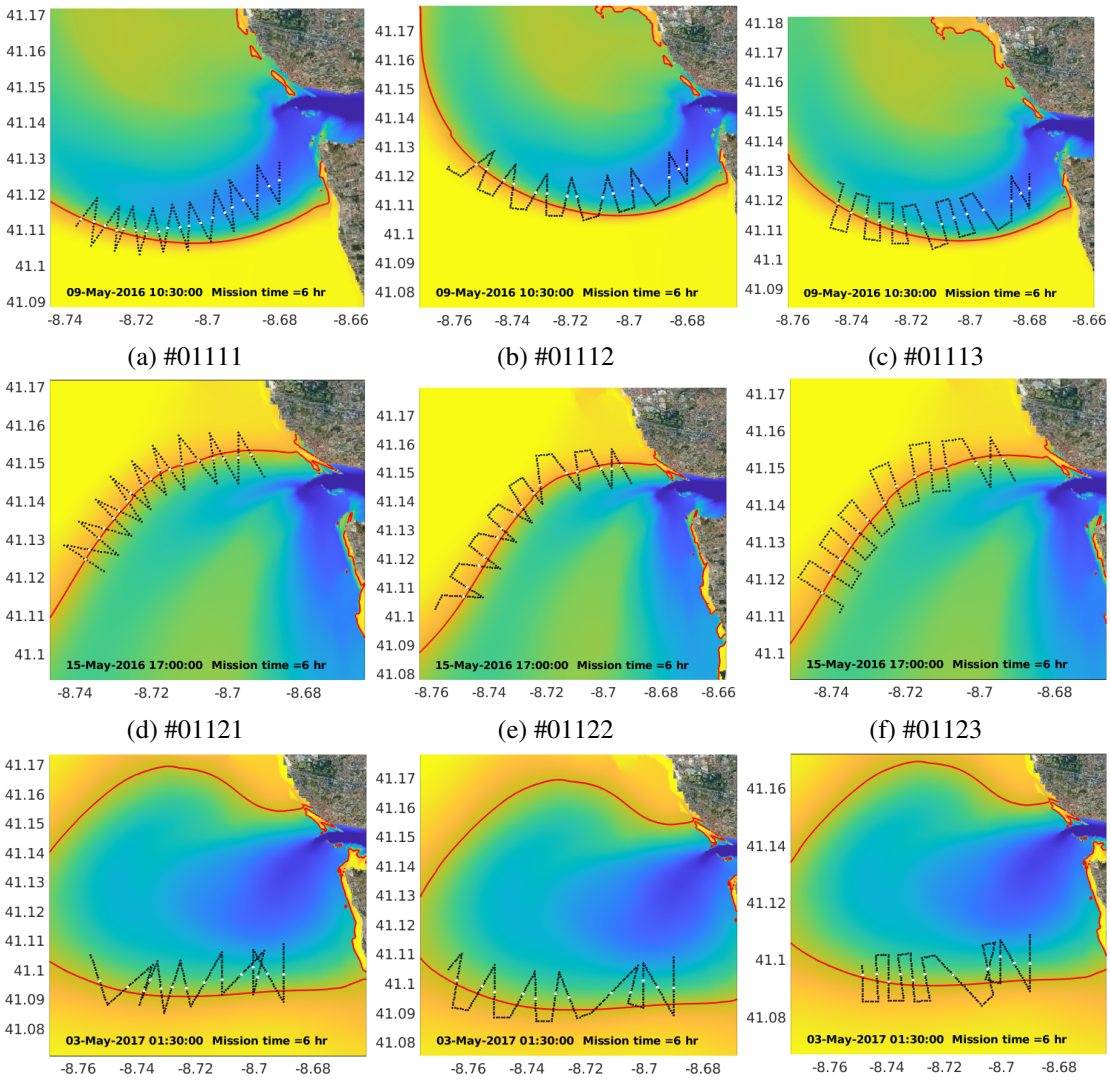


(a) #01011 - $v = 0.5m/s$ (b) #01011 - $v = 2.5m/s$ (c) #01011 - $v = 5.0m/s$

(d) #01021 - $v = 0.5m/s$ (e) #01021 - $v = 2.5m/s$ (f) #01021 - $v = 5.0m/s$

Figure A.3: Test results for the surface tracking of a dynamic plume for different AUV velocities, v. Simulation set **B.2**.

A.3 Tracking of a 3-dimensional plume

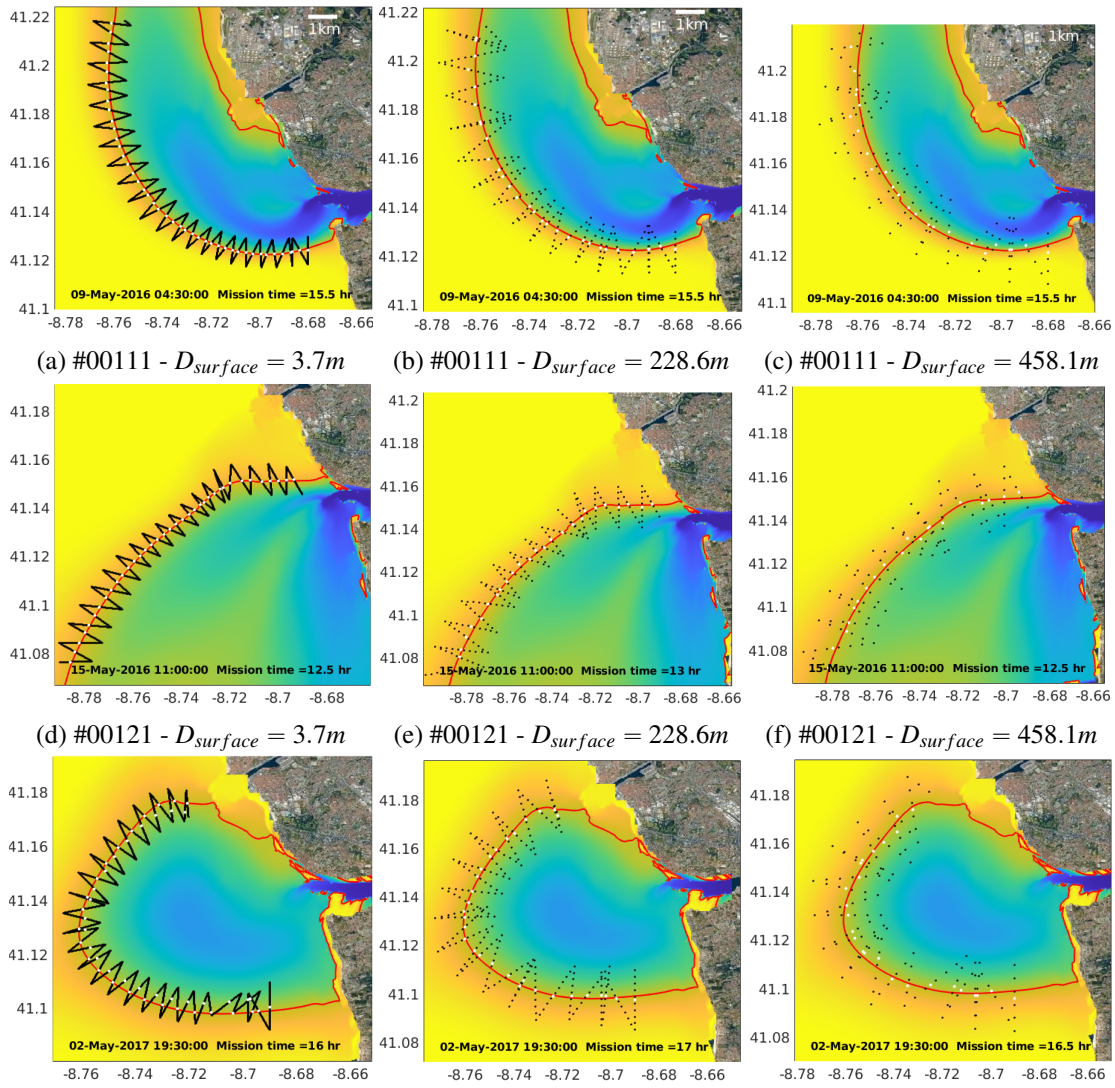


(a) #01111 (b) #01112 (c) #01113

(d) #01121 (e) #01122 (f) #01123

(g) #01131 (h) #01132 (i) #01133

Figure A.4: Test results for tracking a dynamic and 3-dimensional plume. Simulation set **C.1**.



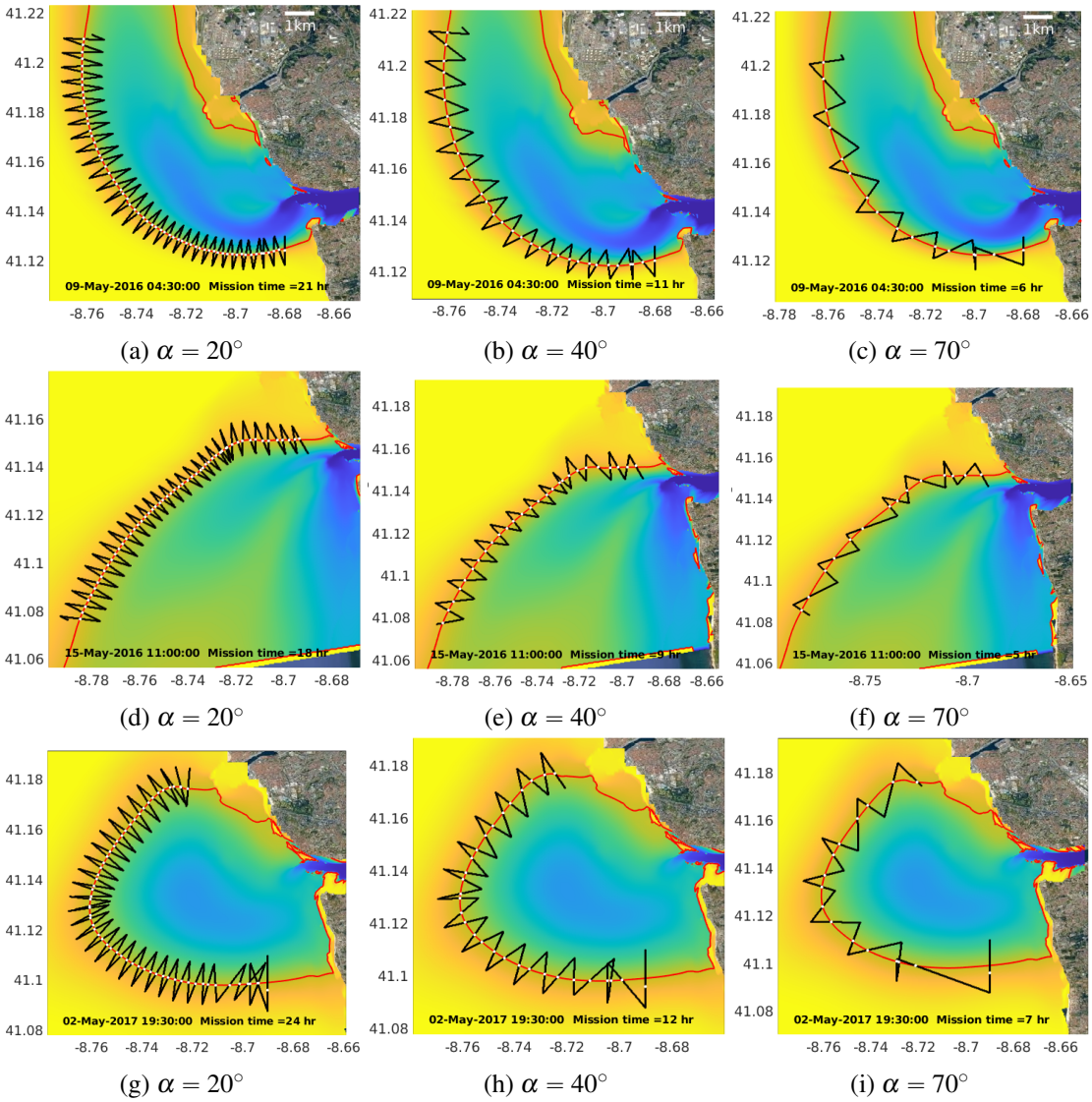
(a) #00111 - $D_{surface} = 3.7m$ (b) #00111 - $D_{surface} = 228.6m$ (c) #00111 - $D_{surface} = 458.1m$

(d) #00121 - $D_{surface} = 3.7m$ (e) #00121 - $D_{surface} = 228.6m$ (f) #00121 - $D_{surface} = 458.1m$

(g) #00131 - $D_{surface} = 3.7m$ (h) #00131 - $D_{surface} = 228.6m$ (i) #00131 - $D_{surface} = 458.1m$

Figure A.5: Test results for tracking a 3-dimensional static plume for different distances between surface locations, $D_{surface}$. Simulation set **C.2**.

A.4 Results for different trajectory parameters from set D.1

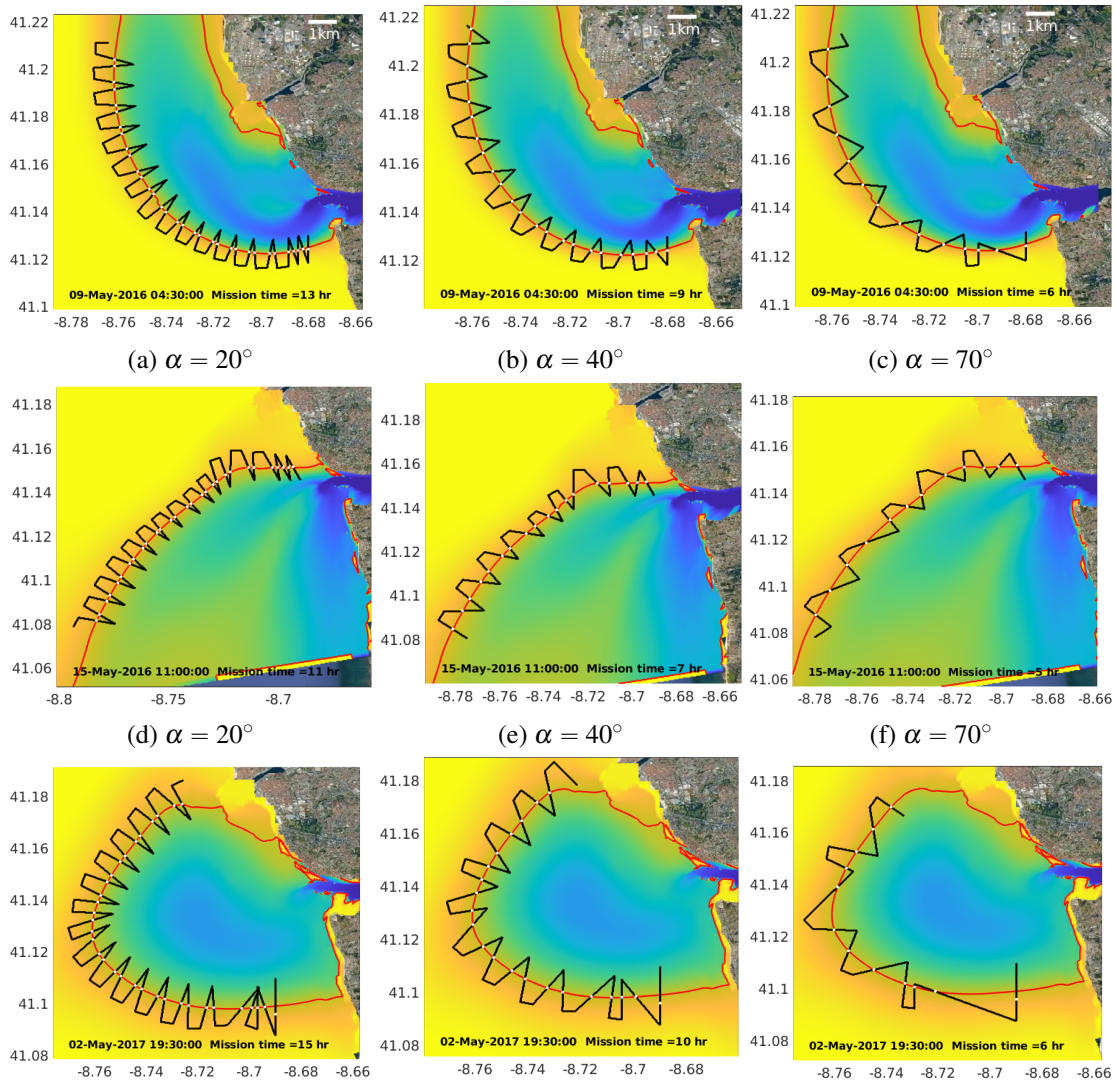


(a) $\alpha = 20^\circ$ (b) $\alpha = 40^\circ$ (c) $\alpha = 70^\circ$

(d) $\alpha = 20^\circ$ (e) $\alpha = 40^\circ$ (f) $\alpha = 70^\circ$

(g) $\alpha = 20^\circ$ (h) $\alpha = 40^\circ$ (i) $\alpha = 70^\circ$

Figure A.6: Results for different values of α using the *Zigzag* strategy. Conditions: (a,b,c) - #00011, (d,e,f) - #00021 and (g,h,i) - #00031. From simulation set **D.1**.

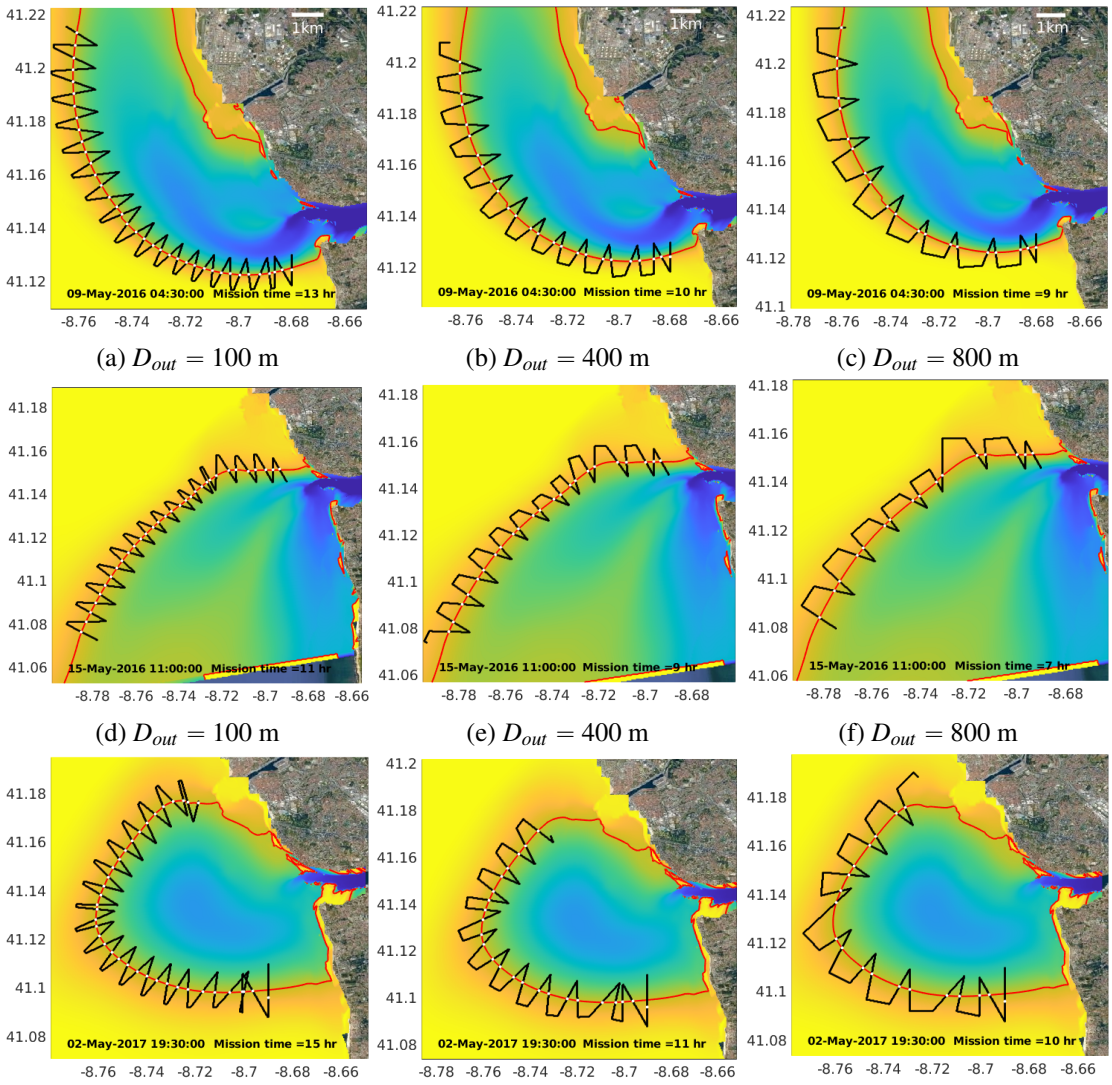


(a) $\alpha = 20^{\circ}$ (b) $\alpha = 40^{\circ}$ (c) $\alpha = 70^{\circ}$

(d) $\alpha = 20^{\circ}$ (e) $\alpha = 40^{\circ}$ (f) $\alpha = 70^{\circ}$

(g) $\alpha = 20^{\circ}$ (h) $\alpha = 40^{\circ}$ (i) $\alpha = 70^{\circ}$

Figure A.7: Results for different values of α using the *Alt. Zigzag* strategy. Conditions: (a,b,c) - #00012, (d,e,f) - #00022 and (g,h,i) - #00032. From simulation set **D.1**.

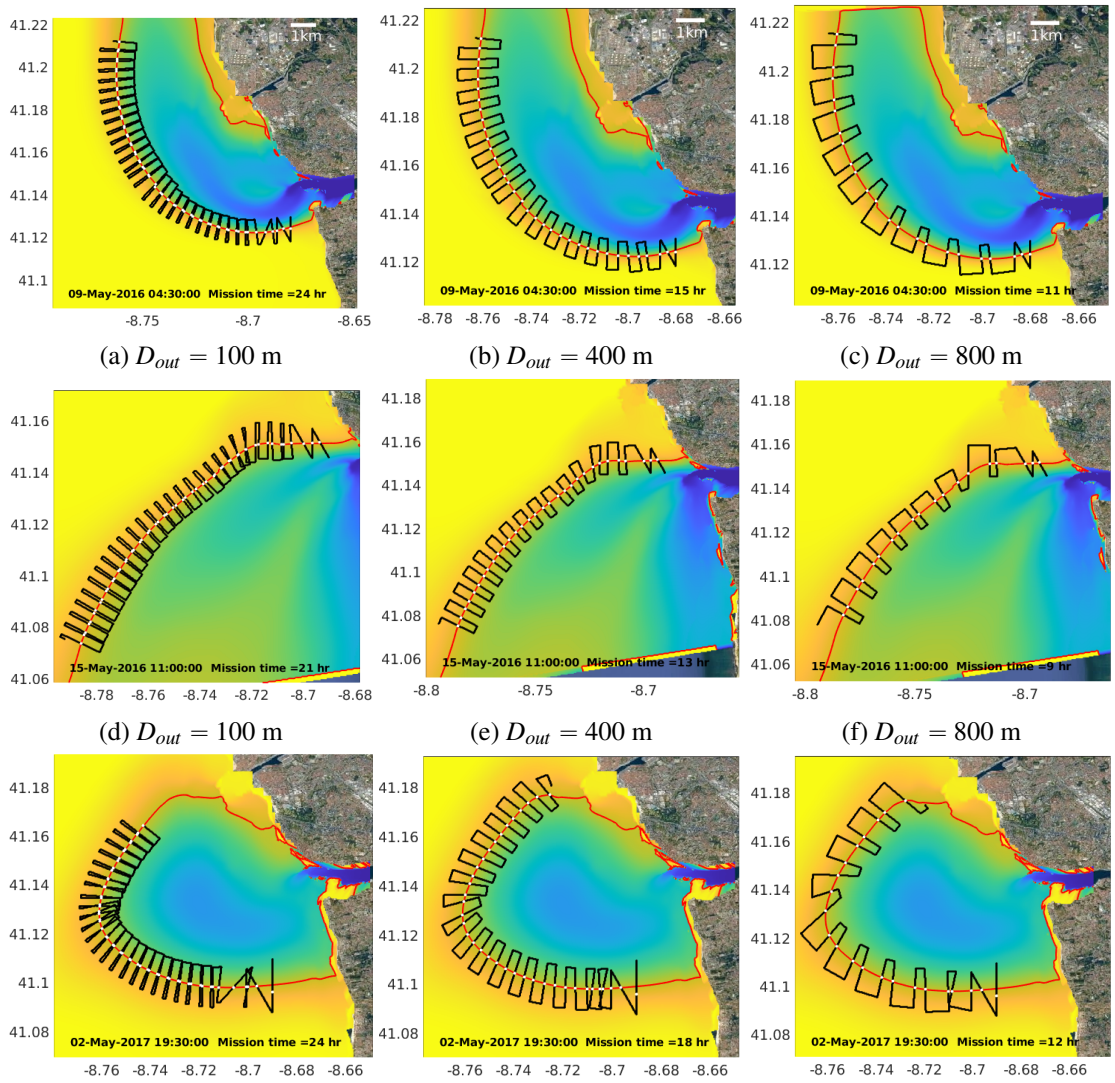


(a) $D_{out} = 100$ m (b) $D_{out} = 400$ m (c) $D_{out} = 800$ m

(d) $D_{out} = 100$ m (e) $D_{out} = 400$ m (f) $D_{out} = 800$ m

(g) $D_{out} = 100$ m (h) $D_{out} = 400$ m (i) $D_{out} = 800$ m

Figure A.8: Results for different values of D_{out} using the *Alt. Zigzag* strategy. Conditions: (a,b,c) - #00012, (d,e,f) - #00022 and (g,h,i) - #00032. From simulation set **D.1**.

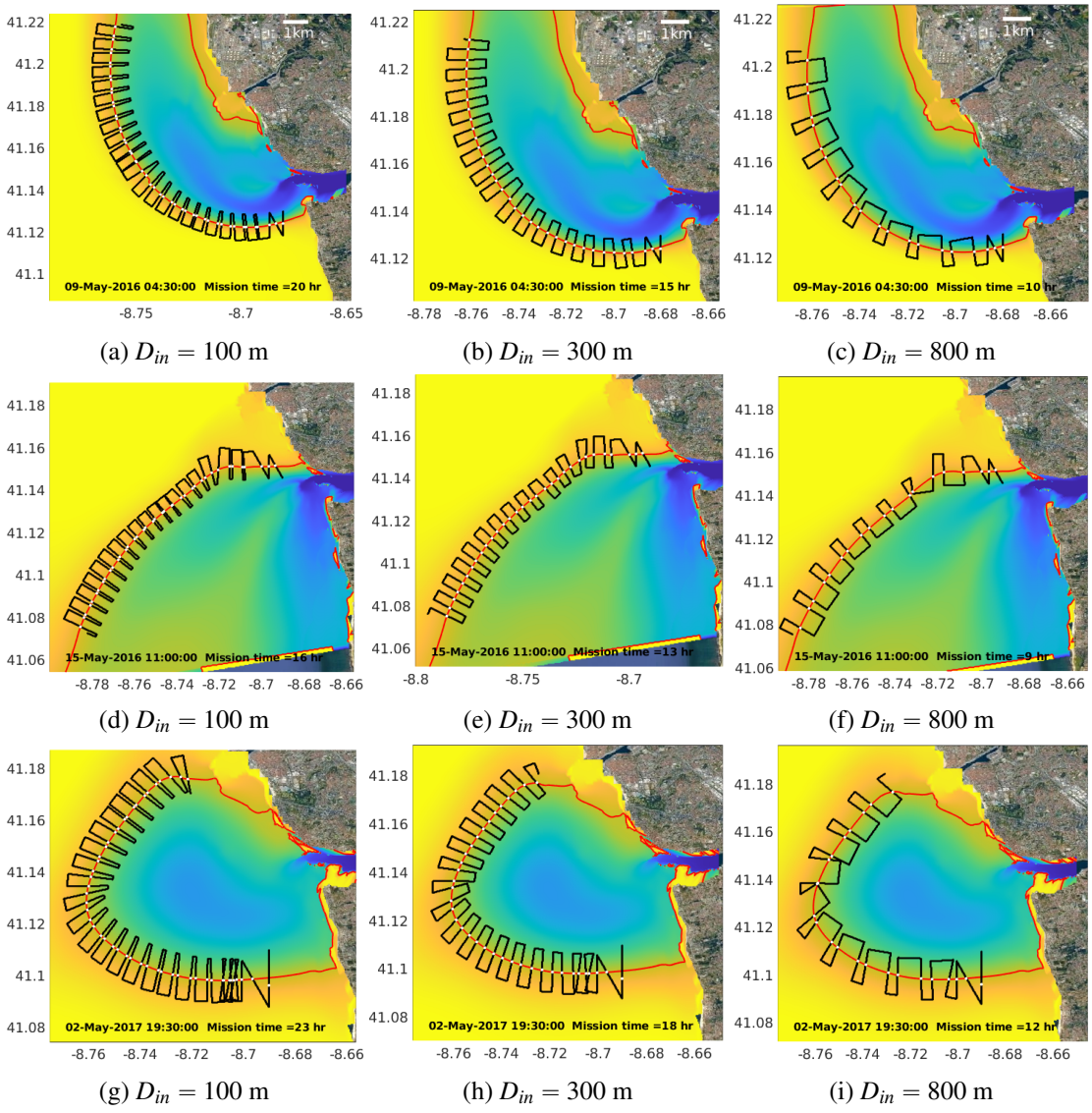


(a) $D_{out} = 100$ m (b) $D_{out} = 400$ m (c) $D_{out} = 800$ m

(d) $D_{out} = 100$ m (e) $D_{out} = 400$ m (f) $D_{out} = 800$ m

(g) $D_{out} = 100$ m (h) $D_{out} = 400$ m (i) $D_{out} = 800$ m

Figure A.9: Results for different values of D_{out} using the *Lawn Mower* strategy. Conditions: (a,b,c) - #00013, (d,e,f) - #00023 and (g,h,i) - #00033. From simulation set **D.1**.



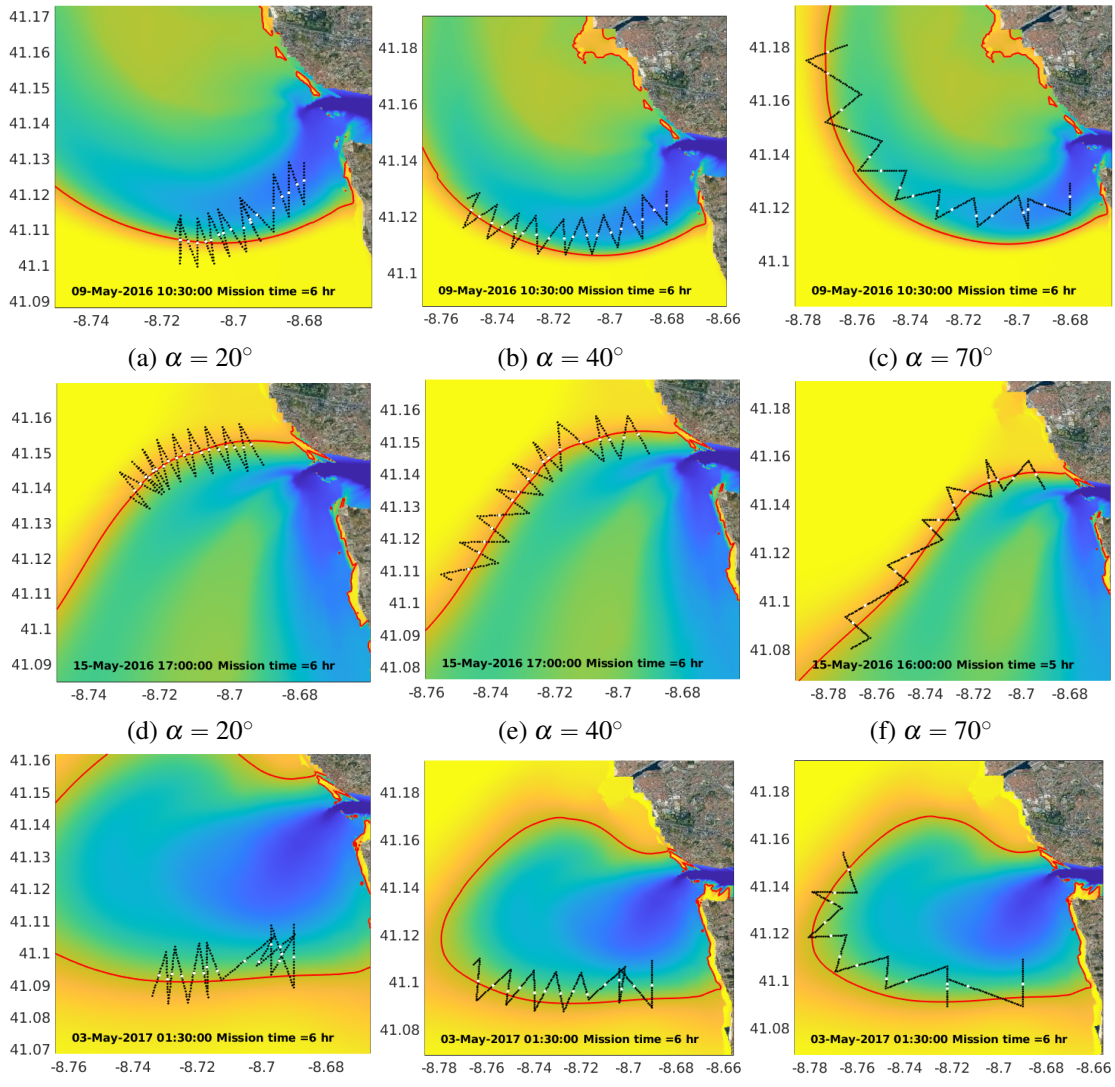
(a) $D_{in} = 100$ m (b) $D_{in} = 300$ m (c) $D_{in} = 800$ m

(d) $D_{in} = 100$ m (e) $D_{in} = 300$ m (f) $D_{in} = 800$ m

(g) $D_{in} = 100$ m (h) $D_{in} = 300$ m (i) $D_{in} = 800$ m

Figure A.10: Results for different values of D_{in} using the *Lawn Mower* strategy. Conditions: (a,b,c) - #00013, (d,e,f) - #00023 and (g,h,i) - #00033. From simulation set **D.1**.

A.5 Results for different trajectory parameters from set D.2

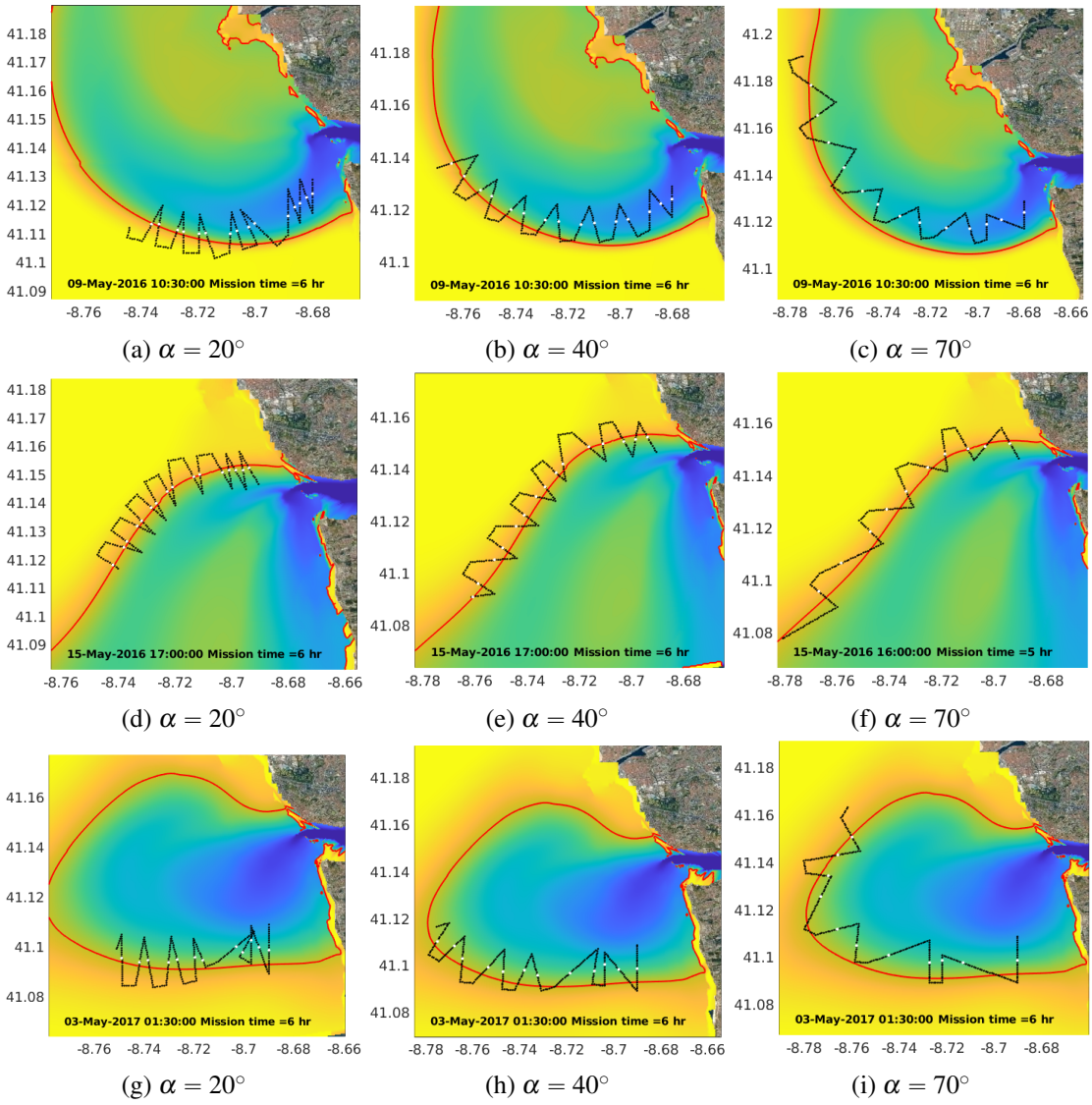


(a) $\alpha = 20^\circ$ (b) $\alpha = 40^\circ$ (c) $\alpha = 70^\circ$

(d) $\alpha = 20^\circ$ (e) $\alpha = 40^\circ$ (f) $\alpha = 70^\circ$

(g) $\alpha = 20^\circ$ (h) $\alpha = 40^\circ$ (i) $\alpha = 70^\circ$

Figure A.11: Results for different values of α using the *Zigzag* strategy. Conditions: (a,b,c) - #01111, (d,e,f) - #01121 and (g,h,i) - #01131. From simulation set **D.2**.

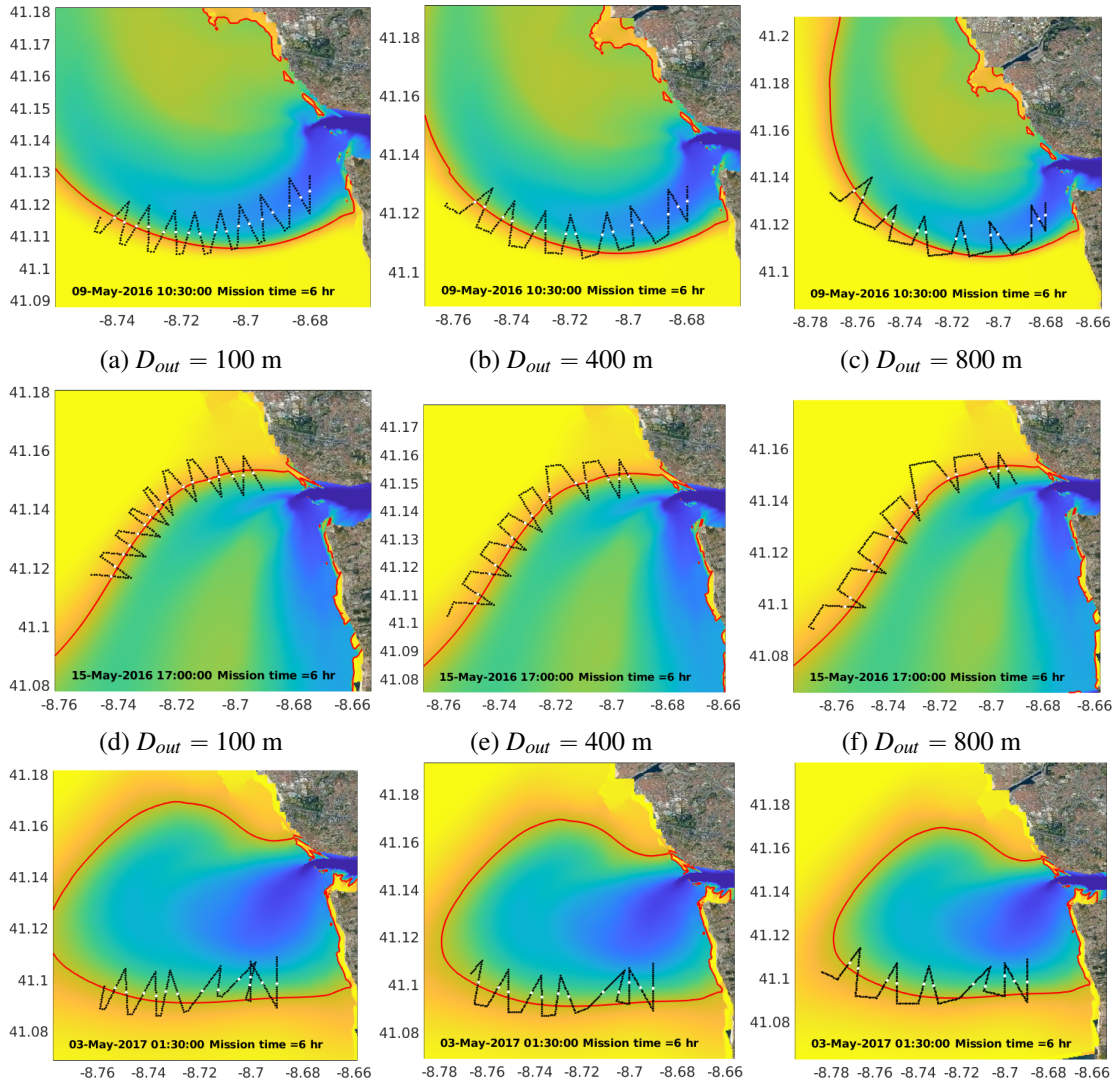


(a) $\alpha = 20°$ (b) $\alpha = 40°$ (c) $\alpha = 70°$

(d) $\alpha = 20°$ (e) $\alpha = 40°$ (f) $\alpha = 70°$

(g) $\alpha = 20°$ (h) $\alpha = 40°$ (i) $\alpha = 70°$

Figure A.12: Results for different values of α using the *Alt. Zigzag* strategy. Conditions: (a,b,c) - #01112, (d,e,f) - #01122 and (g,h,i) - #01132. From simulation set **D.2**.

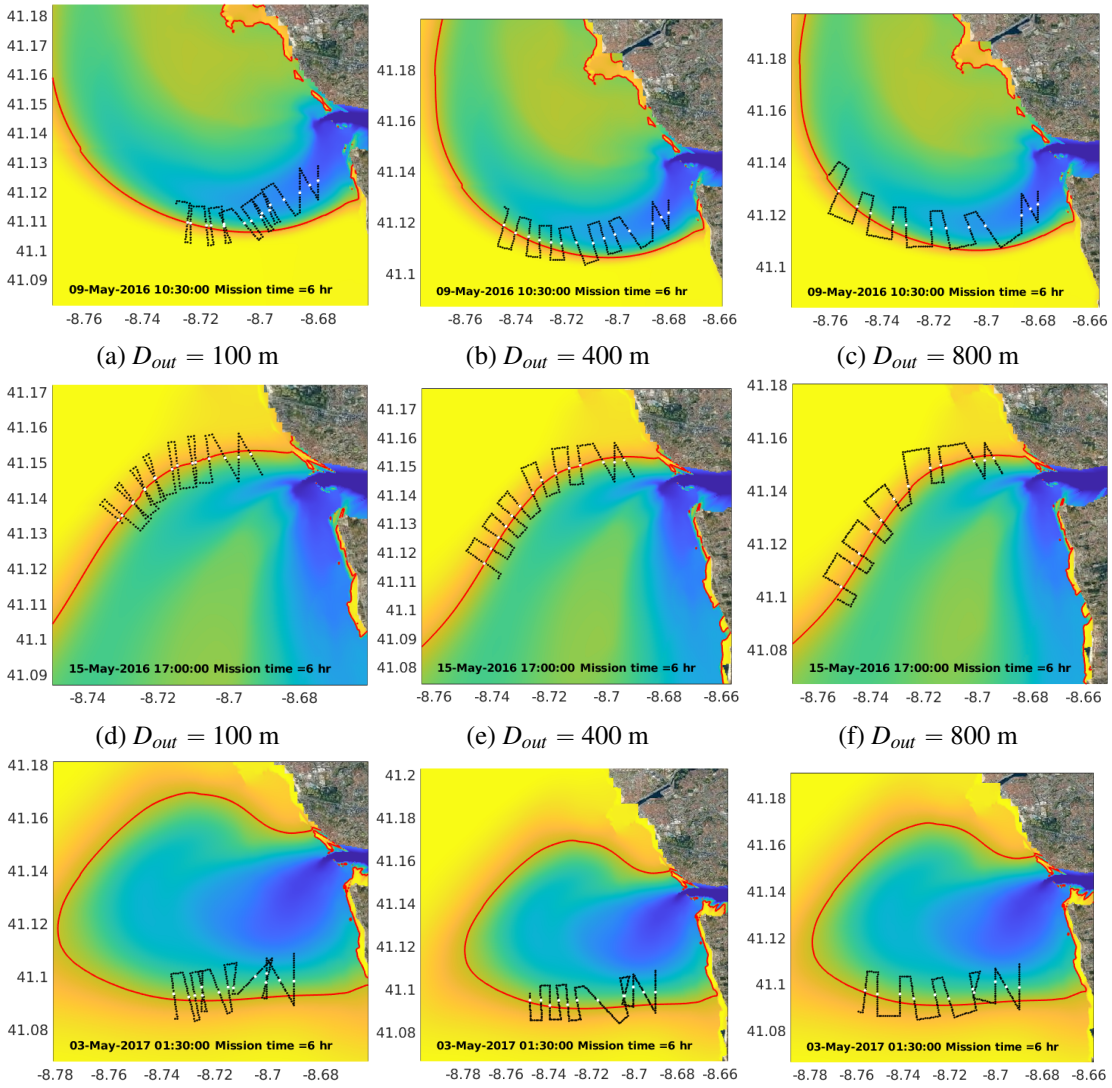


(a) $D_{out} = 100$ m (b) $D_{out} = 400$ m (c) $D_{out} = 800$ m

(d) $D_{out} = 100$ m (e) $D_{out} = 400$ m (f) $D_{out} = 800$ m

(g) $D_{out} = 100$ m (h) $D_{out} = 400$ m (i) $D_{out} = 800$ m

Figure A.13: Results for different values of D_{out} using the *Alt. Zigzag* strategy. Conditions: (a,b,c) - #01112, (d,e,f) - #01122 and (g,h,i) - #01132. From simulation set **D.2**.

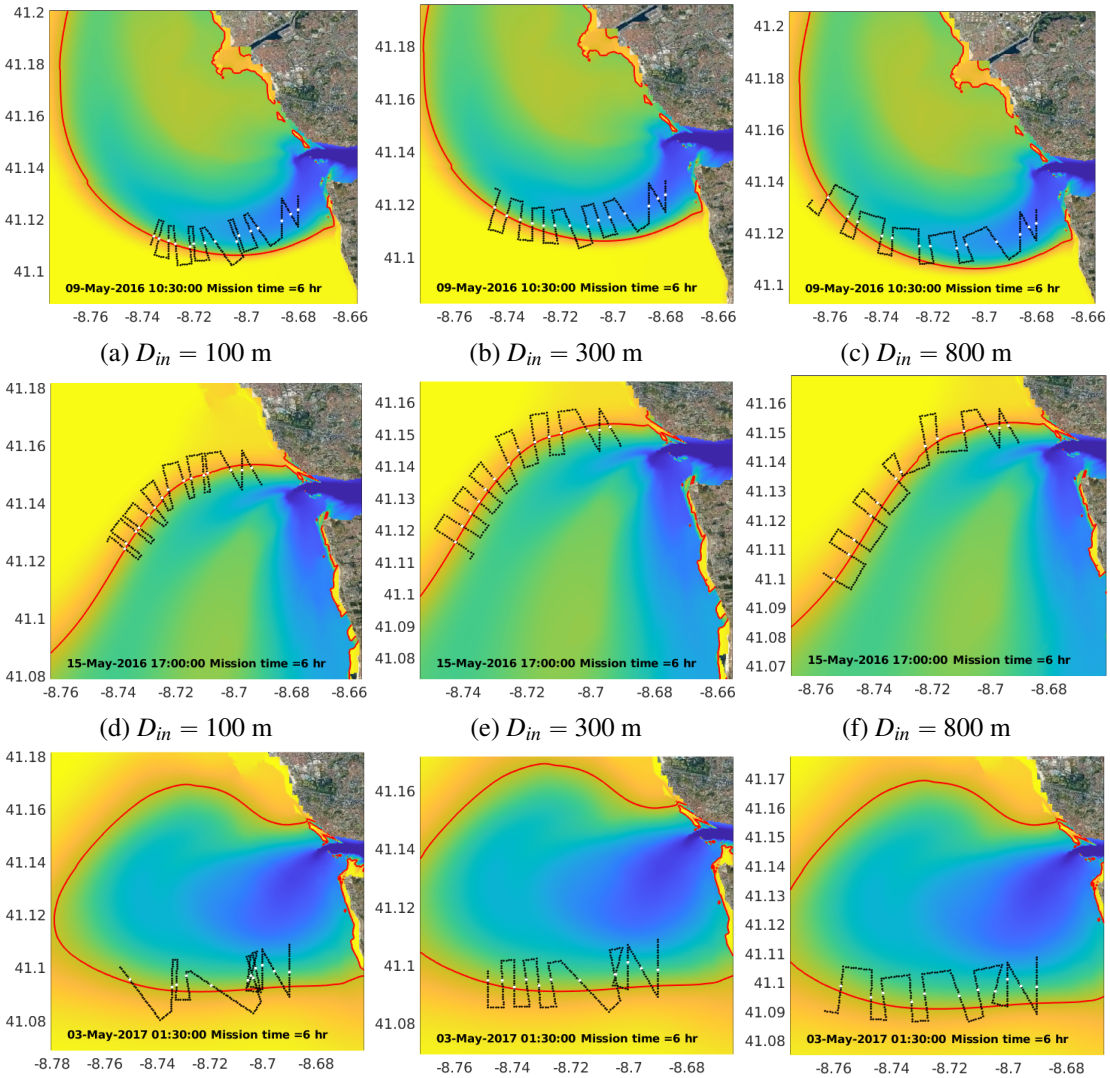


(a) $D_{out} = 100$ m (b) $D_{out} = 400$ m (c) $D_{out} = 800$ m

(d) $D_{out} = 100$ m (e) $D_{out} = 400$ m (f) $D_{out} = 800$ m

(g) $D_{out} = 100$ m (h) $D_{out} = 400$ m (i) $D_{out} = 800$ m

Figure A.14: Results for different values of D_{out} using the *Lawn Mower* strategy. Conditions: (a,b,c) - #01113, (d,e,f) - #01123 and (g,h,i) - #01133. From simulation set **D.2**.



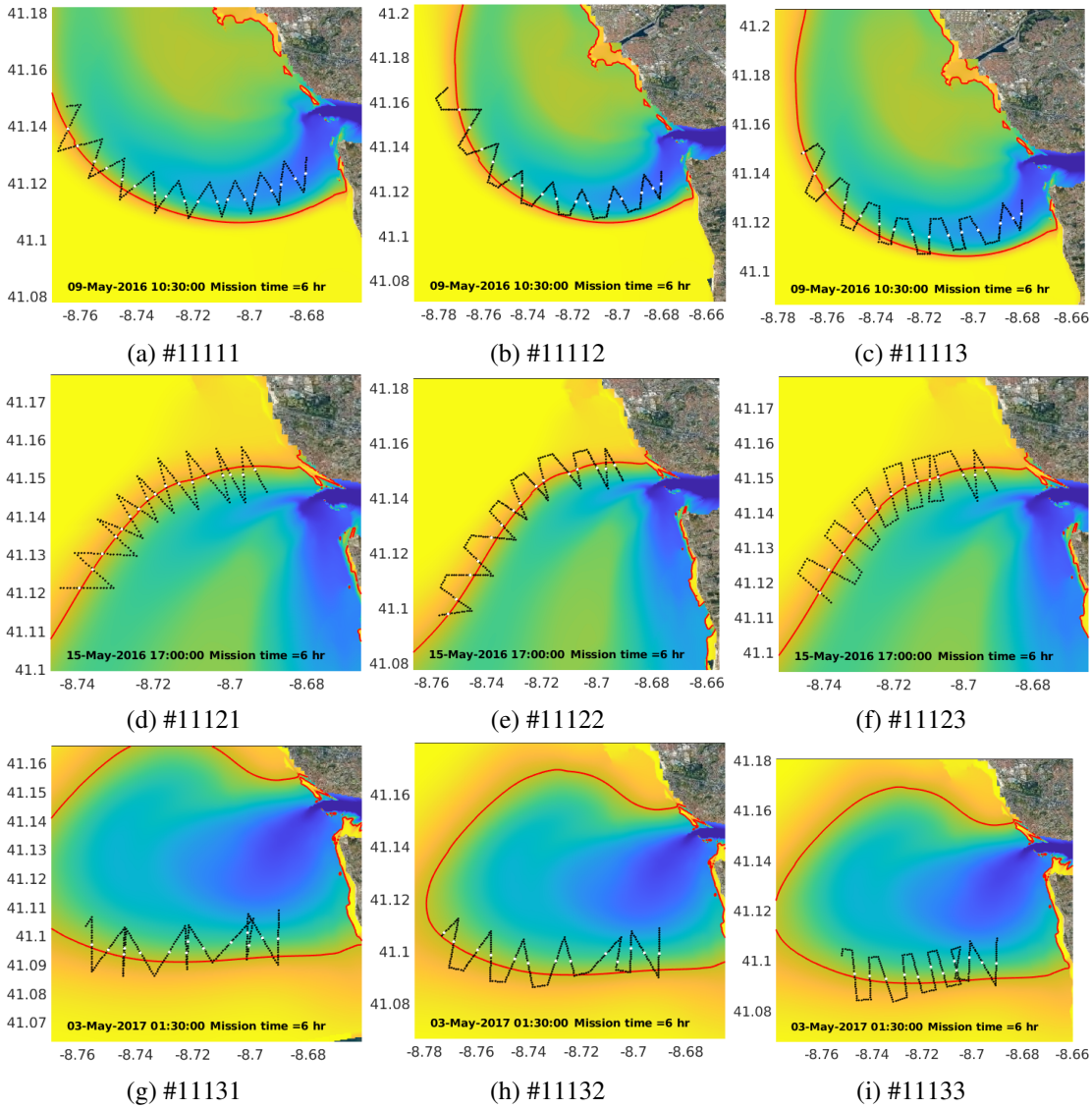
(a) $D_{in} = 100$ m (b) $D_{in} = 300$ m (c) $D_{in} = 800$ m

(d) $D_{in} = 100$ m (e) $D_{in} = 300$ m (f) $D_{in} = 800$ m

(g) $D_{in} = 100$ m (h) $D_{in} = 300$ m (i) $D_{in} = 800$ m

Figure A.15: Results for different values of D_{in} using the *Lawn Mower* strategy. Conditions: (a,b,c) - #01113, (d,e,f) - #01123 and (g,h,i) - #01133. From simulation set **D.2**.

A.6 Simulation with ocean currents

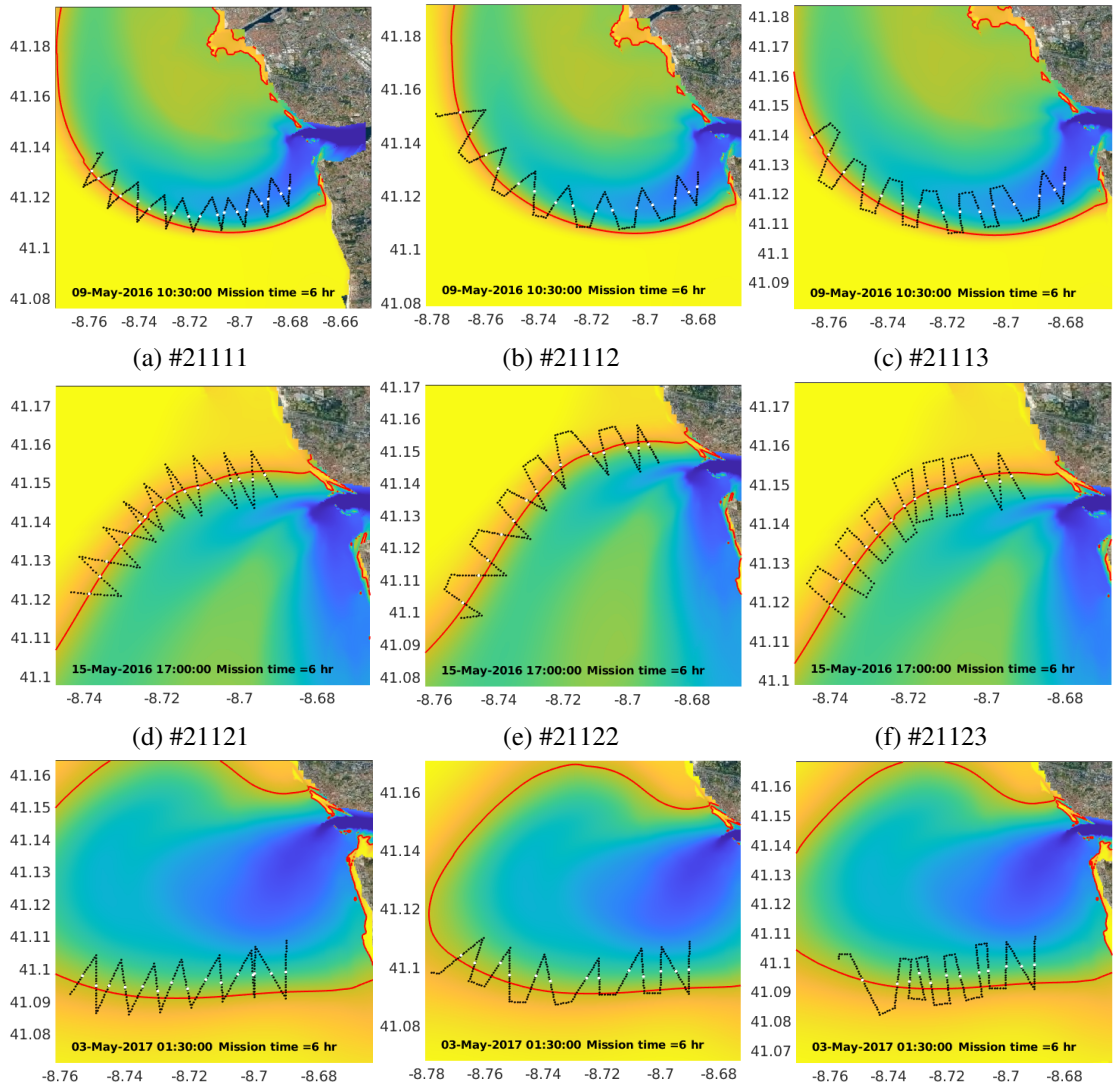


(a) #11111 (b) #11112 (c) #11113

(d) #11121 (e) #11122 (f) #11123

(g) #11131 (h) #11132 (i) #11133

Figure A.16: Test results for a 3D tracking of a dynamic plume with ocean currents. From simulation set **E**.



(a) #21111 (b) #21112 (c) #21113

(d) #21121 (e) #21122 (f) #21123

(g) #21131 (h) #21132 (i) #21133

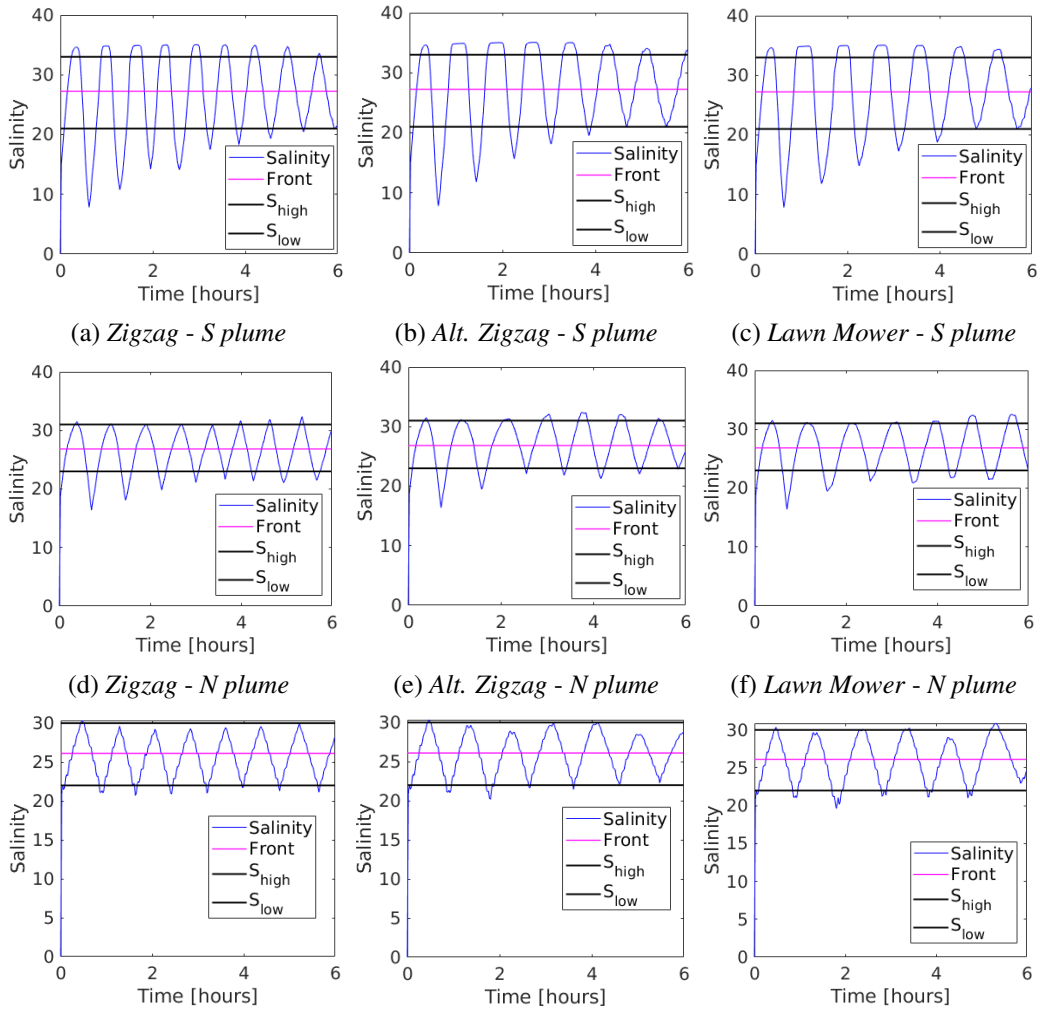
Figure A.17: Test results for a 3D tracking of a dynamic plume with ocean currents and path correction method. From simulation set **E**.

Appendix B

Plume mapping

B.1 Sampled salinity

The following figures represent the salinity sampled from the simulation set **D** with the path correction method active:

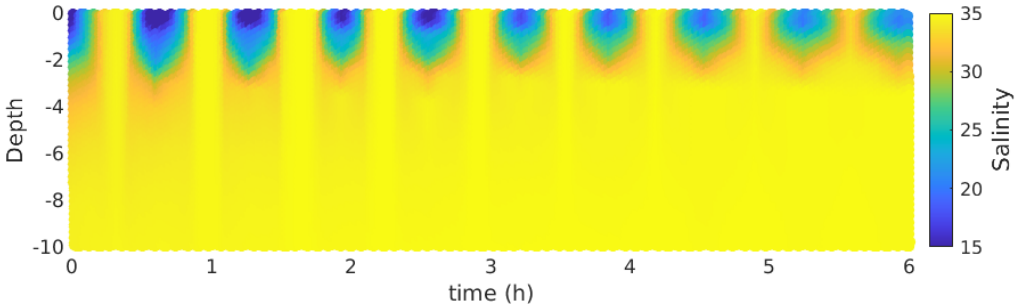


(a) *Zigzag - S plume* (b) *Alt. Zigzag - S plume* (c) *Lawn Mower - S plume*

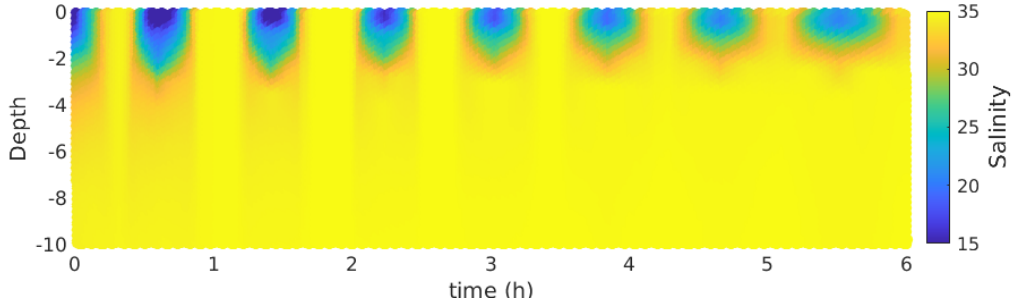
(d) *Zigzag - N plume* (e) *Alt. Zigzag - N plume* (f) *Lawn Mower - N plume*

(g) *Zigzag - LW plume* (h) *Alt. Zigzag - LW plume* (i) *Lawn Mower - LW plume*

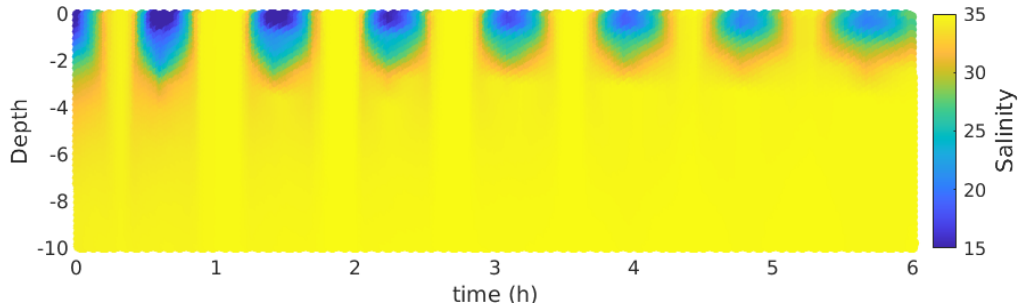
Figure B.1: Sampled surface salinity for all simulations from set **D** with the path correction method.



(a) *Zigzag*

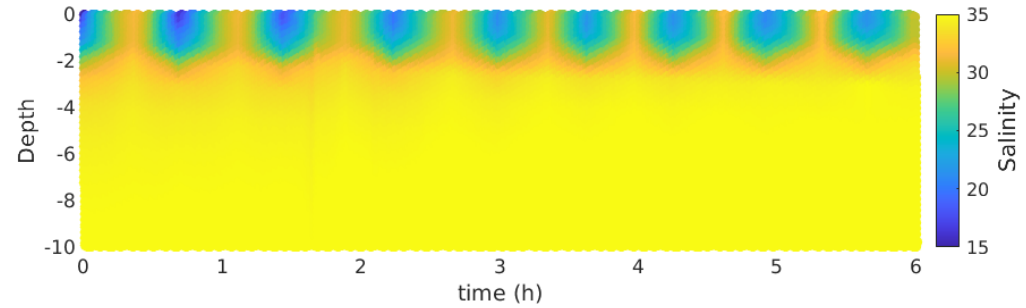


(b) *Alt. Zigzag*

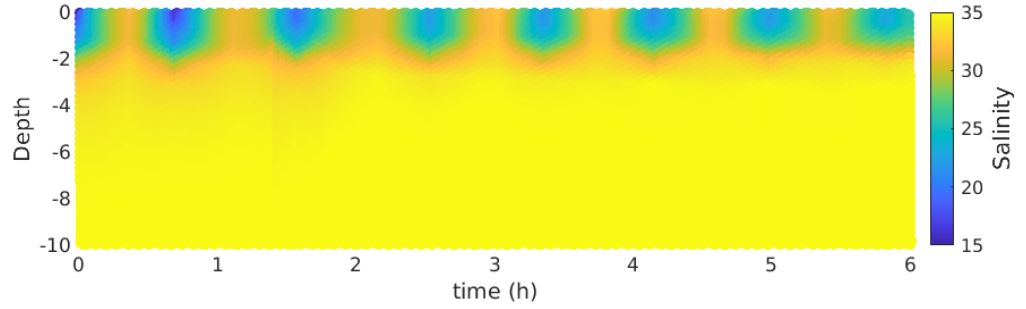


(c) *Lawn Mower*

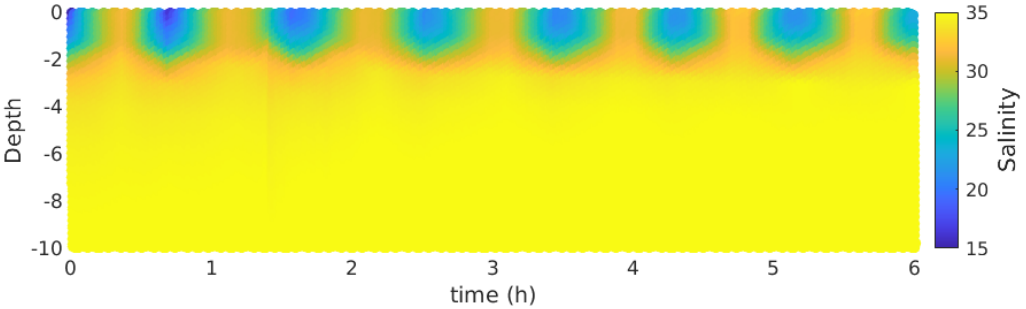
Figure B.2: Sampled salinity of the *S plume*.



(a) *Zigzag*

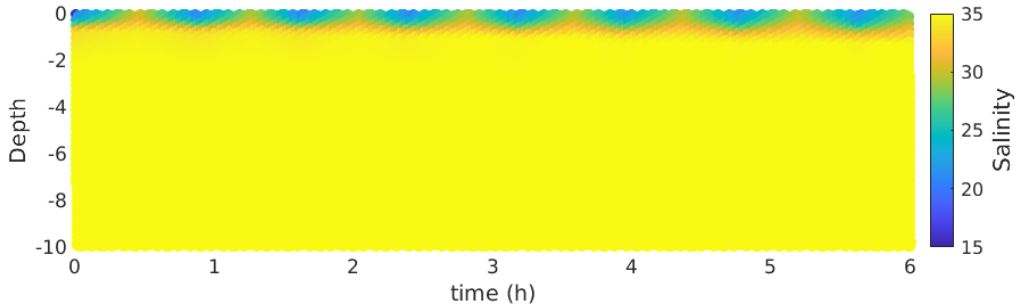


(b) *Alt. Zigzag*

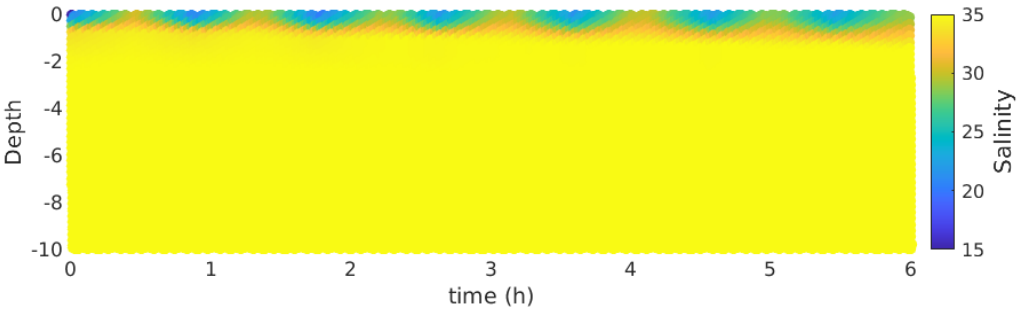


(c) *Lawn Mower*

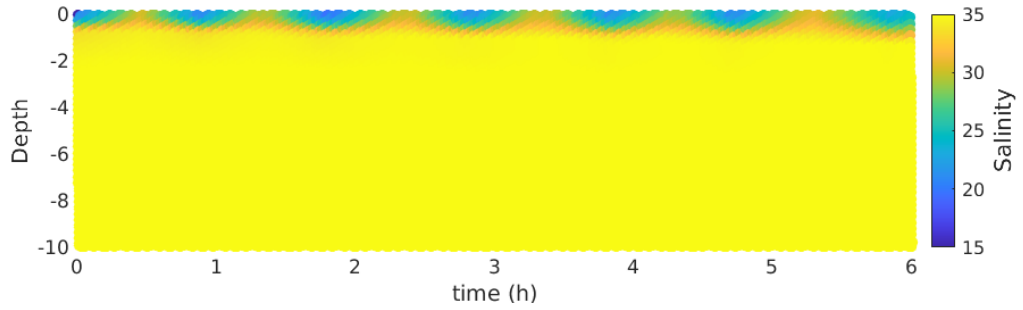
Figure B.3: Sampled salinity of the *N plume*.



(a) *Zigzag*



(b) *Alt. Zigzag*



(c) *Lawn Mower*

Figure B.4: Sampled salinity of the *LW plume*.

References

[1] Joe Jurisa. Columbia river plume. Accessed 24-06-2021. URL: https://joejurisa.wordpress.com/research/columbia-river-plume/.

[2] José Pinto, Renato Mendes, José CB da Silva, Joao M Dias, and Joao Borges de Sousa. Multiple autonomous vehicles applied to plume detection and tracking. In *2018 OCEANS-MTS/IEEE Kobe Techno-Oceans (OTO)*, pages 1–6. IEEE, 2018. doi:10.1109/OCEANSKOBE.2018.8558802.

[3] Renato Mendes. *The Douro estuarine plume: Detection, processes and dynamics*. PhD thesis, Universidade de Aveiro (Portugal), 2016.

[4] LSTS (Laboratório de Sistemas e Tecnologia Subaquática). Navigation messages. Accessed 24-06-2021. URL: https://www.lsts.pt/docs/imc/master/Navigation.html#estimated-state.

[5] Jimin Hwang, Neil Bose, and Shuangshuang Fan. Auv adaptive sampling methods: A review. *Applied Sciences*, 9(15):3145, 2019. doi:10.3390/app9153145.

[6] Renato Mendes, Gonzalo S. Saldías, Maite DeCastro, Moncho Gómez-Gesteira, Nuno Vaz, and João Miguel Dias. Seasonal and interannual variability of the Douro turbid river plume, northwestern Iberian Peninsula. *Remote Sensing of Environment*, 194:401–411, jun 2017. doi:10.1016/j.rse.2017.04.001.

[7] Jose Pinto, Paulo S. Dias, Ricardo Martins, Joao Fortuna, Eduardo Marques, and Joao Sousa. The LSTS toolchain for networked vehicle systems. In *OCEANS 2013 MTS/IEEE Bergen: The Challenges of the Northern Dimension*, 2013. doi:10.1109/OCEANS-Bergen.2013.6608148.

[8] R Mendes, N Vaz, D Fernández-Nóvoa, JCB Da Silva, M Decastro, M Gómez-Gesteira, and JM Dias. Observation of a turbid plume using modis imagery: The case of douro estuary (portugal). *Remote sensing of environment*, 154:127–138, nov 2014. doi:10.1016/j.rse.2014.08.003.

[9] Renato Mendes, Magda Catarina Sousa, Maite DeCastro, Moncho Gómez-Gesteira, and João Miguel Dias. New insights into the Western Iberian Buoyant Plume: Interaction between the Douro and Minho River plumes under winter conditions. *Progress in Oceanography*, 141:30–43, feb 2016. doi:10.1016/j.pocean.2015.11.006.

[10] Isabel C. Azevedo, Adriano A. Bordalo, and Pedro M. Duarte. Influence of river discharge patterns on the hydrodynamics and potential contaminant dispersion in the Douro estuary (Portugal). *Water Research*, 44(10):3133–3146, may 2010. doi:10.1016/j.watres.2010.03.011.

[11] Mário E.C. Vieira and Adriano A. Bordalo. The Douro estuary (Portugal): A mesotidal salt wedge. *Oceanologica Acta*, 23(5):585–594, oct 2000. doi:10.1016/S0399-1784(00)01107-5.

[12] Magda C Sousa, Américo S Ribeiro, Marisela Des, Renato Mendes, Ines Alvarez, Moncho Gomez-Gesteira, and João M Dias. Integrated high-resolution numerical model for the nw iberian peninsula coast and main estuarine systems. *Journal of Coastal Research*, pages 66–70, 2018. doi:10.2112/SI85-014.1.

[13] Hubert Loisel, Vincent Vantrepotte, Cédric Jamet, and D Ngoc Dat. Challenges and new advances in ocean color remote sensing of coastal waters. *Topics in Oceanography*, pages 1–38, jul 2013. doi:10.5772/56414.

[14] NASA (National Aeronautics and Space Administration). Modis, moderate resolution imaging spectroradiometer. Accessed 24-06-2021. URL: https://modis.gsfc.nasa.gov/about/.

[15] Thor I Fossen. Guidance and control of ocean vehicles. *University of Trondheim, Norway, Printed by John Wiley & Sons, Chichester, England, ISBN: 0 471 94113 1, Doctors Thesis*, 1999.

[16] Alex Alcocer, Paulo Oliveira, and Antonio Pascoal. Underwater acoustic positioning systems based on buoys with gps. In *Proceedings of the Eighth European Conference on Underwater Acoustics*, volume 8, pages 1–8, 2006.

[17] Jacob Anderson and Ryan N Smith. Predicting water properties with markov random fields for augmented terrain-based navigation in autonomous underwater vehicles. In *2018 OCEANS-MTS/IEEE Kobe Techno-Oceans (OTO)*, pages 1–5. IEEE, may 2018. doi:10.1109/OCEANSKOBE.2018.8559210.

[18] Ryan N Smith, Jonathan Kelly, and Gaurav S Sukhatme. Towards improving mission execution for autonomous gliders with an ocean model and kalman filter. In *2012 IEEE International Conference on Robotics and Automation*, pages 4870–4877. IEEE, 2012. doi:10.1109/ICRA.2012.6224609.

[19] Ryan N Smith, Jonathan Kelly, Kimia Nazarzadeh, and Gaurav S Sukhatme. An investigation on the accuracy of regional ocean models through field trials. In *2013 IEEE International Conference on Robotics and Automation*, pages 3436–3442. IEEE, 2013. doi:10.1109/ICRA.2013.6631057.

[20] Muhammad Fahad, Nathaniel Saul, Yi Guo, and Brian Bingham. Robotic simulation of dynamic plume tracking by unmanned surface vessels. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2654–2659. IEEE, 2015. doi:10.1109/ICRA.2015.7139557.

[21] Joana Fonseca, Jieqiang Wei, Karl H. Johansson, and Tor Ame Johansen. Cooperative decentralised circumnavigation with application to algal bloom tracking. In *IEEE International Conference on Intelligent Robots and Systems*, pages 3276–3281. IEEE, nov 2019. arXiv:1903.05993, doi:10.1109/IROS40897.2019.8967632.

[22] Joshua Grady Graver. *Underwater gliders: Dynamics, control and design*. PhD thesis, Princeton University (USA), 2005.

[23] Van T Huynh, Matthew Dunbabin, and Ryan N Smith. Predictive motion planning for auvs subject to strong time-varying currents and forecasting uncertainties. In *2015 IEEE international conference on robotics and automation (ICRA)*, pages 1144–1151. IEEE, 2015. doi:10.1109/ICRA.2015.7139335.

[24] Amy Kukulya, A Plueddemann, T Austin, R Stokey, M Purcell, B Allen, R Littlefield, L Freitag, P Koski, E Gallimore, et al. Under-ice operations with a remus-100 auv in the arctic. In *2010 IEEE/OES Autonomous Underwater Vehicles*, pages 1–8. IEEE, 2010. doi:10.1109/AUV.2010.5779661.

[25] Yanwu Zhang, Robert S McEwen, John P Ryan, James G Bellingham, Hans Thomas, Charles H Thompson, and Erich Rienecker. A peak-capture algorithm used on an autonomous underwater vehicle in the 2010 gulf of mexico oil spill response scientific survey. *Journal of Field Robotics*, 28(4):484–496, 2011. doi:10.1002/rob.20399.

[26] Yanwu Zhang, John P Ryan, Brian Kieft, Brett W Hobson, Robert S McEwen, Michael A Godin, Julio B Harvey, Benedetto Barone, James G Bellingham, James M Birch, et al. Targeted sampling by autonomous underwater vehicles. *Frontiers in Marine Science*, 6:415, 2019. doi:10.3389/fmars.2019.00415.

[27] Per Espen Hagen, Øivind Midtgaard, and Øistein Hasvold. Making AUVs truly autonomous. In *Oceans Conference Record (IEEE)*. IEEE, 2007. doi:10.1109/OCEANS.2007.4449405.

[28] S. Li, Y. Guo, and B. Bingham. Multi-robot cooperative control for monitoring and tracking dynamic plumes. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 67–73, 2014. doi:10.1109/ICRA.2014.6906591.

[29] Jimin Hwang, Neil Bose, Hung Nguyen, and Guy Williams. Auv adaptive sampling to delineate subsurface oil plumes. In *2020 IEEE/OES Autonomous Underwater Vehicles Symposium (AUV)(50043)*, pages 1–2. IEEE, 2020. doi:10.1109/AUV50043.2020.9267920.

[30] Jimin Hwang, Neil Bose, Hung Duc Nguyen, and Guy Williams. Oil plume mapping: Adaptive tracking and adaptive sampling from an autonomous underwater vehicle. *IEEE Access*, 8:198021–198034, 2020. doi:10.1109/ACCESS.2020.3032161.

[31] Yanwu Zhang, Robert S. McEwen, John P. Ryan, and James G. Bellingham. Design and tests of an adaptive triggering method for capturing peak samples in a thin phytoplankton layer by an autonomous underwater vehicle. *IEEE Journal of Oceanic Engineering*, 35(4):785–796, oct 2010. doi:10.1109/JOE.2010.2081031.

[32] Erik Burian, Dana Yoerger, Albert Bradley, and Hanumant Singh. Gradient search with autonomous underwater vehicles using scalar measurements. In *Proceedings of the IEEE Symposium on Autonomous Underwater Vehicle Technology*, pages 86–98. IEEE, 1996. doi:10.1109/auv.1996.532404.

[33] Xiaodong Ai, Keyou You, and Shiji Song. A source-seeking strategy for an autonomous underwater vehicle via on-line field estimation. In *2016 14th International Conference on Control, Automation, Robotics and Vision (ICARCV)*, pages 1–6. IEEE, 2016. doi:10.1109/ICARCV.2016.7838757.

[34] Jay A Farrell, Shuo Pang, and Wei Li. Chemical plume tracing via an autonomous underwater vehicle. *IEEE Journal of Oceanic Engineering*, 30(2):428–442, 2005. doi:10.1109/JOE.2004.838066.

[35] Joana Fonseca, Jieqiang Wei, Tor Arne Johansen, and Karl Henrik Johansson. Cooperative circumnavigation for a mobile target using adaptive estimation. In *Portuguese Conference on Automatic Control*, pages 33–48. Springer, 2020.

[36] Igor Belkin, Joao Borges De Sousa, Jose Pinto, Renato Mendes, and Francisco Lopez-Castejon. A new front-tracking algorithm for marine robots. In *AUV 2018 - 2018 IEEE/OES Autonomous Underwater Vehicle Workshop, Proceedings*. Institute of Electrical and Electronics Engineers Inc., nov 2018. doi:10.1109/AUV.2018.8729788.

[37] Y. Zhang, M. A. Godin, J. G. Bellingham, and J. P. Ryan. Using an autonomous underwater vehicle to track a coastal upwelling front. *IEEE Journal of Oceanic Engineering*, 37(3):338–347, 2012. doi:10.1109/JOE.2012.2197272.

[38] Christopher J Cannell, Aditya S Gadre, and Daniel J Stilwell. Boundary tracking and rapid mapping of a thermal plume using an autonomous vehicle. In *OCEANS 2006*, pages 1–6. IEEE, 2006. doi:10.1109/OCEANS.2006.306807.

[39] Christopher J Cannell and Daniel J Stilwell. A comparison of two approaches for adaptive sampling of environmental processes using autonomous underwater vehicles. In *Proceedings of OCEANS 2005 MTS/IEEE*, pages 1514–1521. IEEE, 2005. doi:10.1109/OCEANS.2005.1639970.

[40] Dhanushka Kularatne, Ryan N Smith, and M Ani Hsieh. Zig-zag wanderer: Towards adaptive tracking of time-varying coherent structures in the ocean. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3253–3258. IEEE, 2015. doi:10.1109/ICRA.2015.7139647.

[41] Ryan N Smith, Philip Cooksey, Frederic Py, Gaurav S Sukhatme, and Kanna Rajan. Adaptive path planning for tracking ocean fronts with an autonomous underwater vehicle. In *Experimental robotics*, pages 761–775. Springer, 2016. doi:10.1007/978-3-319-23778-7_50.

[42] J Scott Willcox, James G Bellingham, Yanwu Zhang, and Arthur B Baggeroer. Performance metrics for oceanographic surveys with autonomous underwater vehicles. *IEEE Journal of Oceanic Engineering*, 26(4):711–725, 2001. doi:10.1109/48.972114.

[43] Ricardo Martins, Paulo Sousa Dias, Eduardo RB Marques, José Pinto, Joao B Sousa, and Fernando L Pereira. Imc: A communication protocol for networked vehicles and sensors. In *Oceans 2009-Europe*, pages 1–6. IEEE, 2009. doi:10.1109/OCEANSE.2009.5278245.

[44] FL Pereira, J Pinto, JB Sousa, RMF Gomes, Gil Manuel Gonçalves, and PS Dias. Mission planning and specification in the neptus framework. In *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006.*, pages 3220–3225. IEEE, 2006.