Boundary tracking and source seeking of oceanic features using autonomous vehicles

Submitted by **Chiara Mellucci** to the University of Exeter as a thesis for the degree of Doctor of Philosophy in Mathematics in January 2019

This thesis is available for Library use on the understanding that it is copyright material and that no quotation from the thesis may be published without proper acknowledgement.

I certify that all material in this thesis which is not my own work has been identified and that no material has previously been submitted and approved for the award of a degree by this or any other University.

Signature:.....................................

Abstract

Boundary tracking and source seeking of oceanic features using autonomous vehicles

Chiara Mellucci

The thesis concerns the study and the development of boundary tracking and source seeking approaches for autonomous vehicles, specifically for marine autonomous systems. The underlying idea is that the characterization of most environmental features can be posed from either a boundary tracking or a source seeking perspective. The suboptimal sliding mode boundary tracking approach is considered and, as a first contribution, it is extended to the study of three dimensional features. The approach is aimed at controlling the movement of an underwater glider tracking a three-dimensional underwater feature and it is validated in a simulated environment. Subsequently, a source seeking approach based on sliding mode extremum seeking ideas is proposed. This approach is developed for the application to a single surface autonomous vehicle, seeking the source of a static or dynamic two dimensional spatial field. A sufficient condition which guarantees the finite time convergence to a neighbourhood of the source is introduced. Furthermore, a probabilistic learning boundary tracking approach is proposed, aimed at exploiting the available preliminary information relating to the spatial phenomenon of interest in the control strategy. As an additional contribution, the sliding mode boundary tracking approach is experimentally validated in a set of sea-trials with the deployment of a surface autonomous vehicle. Finally, an embedded system implementing the proposed boundary tracking strategy is developed for future installation on board of the autonomous vehicle. This work demonstrates the possibility to perform boundary tracking with a fully autonomous vehicle and to operate marine autonomous systems without remote control or pre-planning. Conclusions are drawn from the results of the research presented in this thesis and directions for future work are identified.

Acknowledgements

I would like to thank my supervisors, Dr. Prathyush P. Menon and Prof. Christopher Edwards for having supported and advised me throughout the years of my PhD. I would also like to thank Prof. Peter Challenor for his help and his advice. It was a privilege for me to collaborate with such inspirational people.

I would like to thank my house mates Hannah, Ashley, Pilar, Mirella and Chiara, for having made me feel at home and for having shared my passion for cooking. A special thank goes to Moira, Franci, Isa and Andrea BCP, who have supported me from Italy throughout this journey despite the distance. I would like to thank all the friends and colleagues I have met over the last few years, who have made my time in Exeter so enjoyable. Special mention goes to Marta, Francesca and Laura, for making me miss Italy a little less, and Celia, Elena and Michele, for being my second family here in Exeter. I am thankful to Michael for his patience, positivity and support.

Finally, the biggest thank goes to my parents, for their irreplaceable support and affection, and to my grandparents, who even if not here, have inspired and motivated me along the way.

Contents

List of Figures

List of Tables

Acronyms

2-SM Second Order Sliding Mode. 58, 227

AAOSN Adaptive Autonomous Ocean Sampling Networks. 16, 180

AIC Akaike Information Criterion. 155, 170, 171, 244

ASV Autonomous Surface Vehicle. 16, 17, 19, 21, 23, 24, 30, 31, 35, 49–51, 54, 69, 82, 111, 198, 201, 202, 204, 205, 207, 211–213

AUV Autonomous Underwater Vehicle. 17, 20, 30, 32, 49, 212, 213

BIC Bayesian Information Criterion. 155, 170, 171, 244

COG Course Over Ground. 182

DMM Decision Making Module. 206

DSTL Defence Science and Technology Laboratory. 16

FOAM Forecast Ocean Assimilation Model. 7, 8, 68, 69, 166, 167, 173, 176

GCS Ground Control Station. 196, 203

GP Gaussian Process. 24, 149–151, 158–160, 168–170, 239, 243, 246, 248

GSO Glowworm Swarm Optimization. 38

HAB Harmful Algae Bloom. 19, 29

MAS Marine Autonomous Systems. 11, 15, 17, 213

NERC Natural Environment Research Council. 16

QQ Quantile-Quantile. 9, 156, 172, 173

RMSE Root Mean Square Error. 156, 158, 167, 173, 174, 177, 245

ROMS Regional Ocean Model System. 147

ROS Robot Operating System. 181–186, 205–210

SE Squared Exponential. 150, 159, 170–174, 243–245

SMC Sliding Mode Control. 54–56, 58, 218

SOG Speed Over Ground. 182

SST Sea-Surface Temperature. 51, 69, 70

TRL Technology Readiness Level. 215

UAV Unmanned Aerial Vehicle. 20, 30, 32, 35, 51, 213

UGV Unmanned Ground Vehicle. 30, 36, 54, 213

VP Virtual Pilot. 185–187, 196, 205

VSC Variable Structure Control. 55, 217, 218, 221

WP Waypoint. 32, 33, 84, 147, 183, 185–187, 191–195

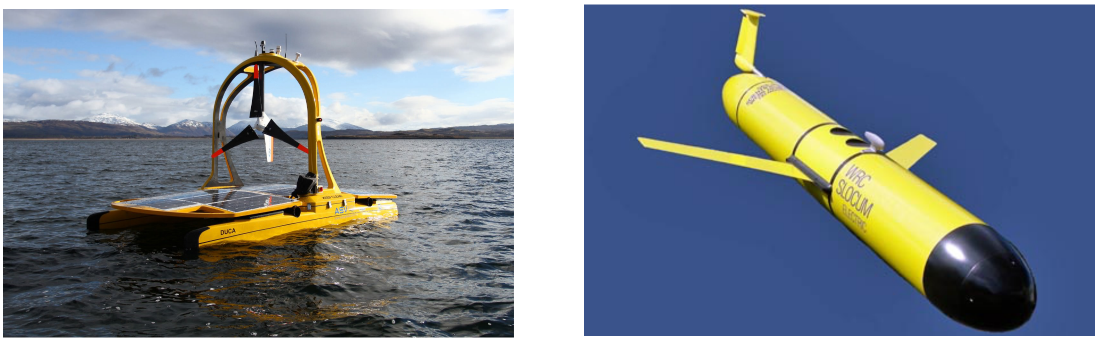
Chapter 1

Introduction

In the past decades, the fast development of autonomous vehicles has significantly modified the way in which environmental features are explored and monitored. Autonomous vehicles are increasingly being used for sampling environmental features, allowing the collection of data with high spatial and temporal resolution. The operational costs associated with these measurements are reduced with respect to traditional sampling techniques, based on static sensors or on human assisted monitoring [1]. In addition, the deployment of autonomous vehicles avoids risks for human operators when dealing with hazardous phenomena, such as oil spills or volcanic eruptions [2], [3]. The use of autonomous vehicles for environmental monitoring is particularly active in the oceanic environment: oceans constitute over 70% of the Earth surface and are the least precisely known part of the environment. This lack of knowledge, and the desire to explore the oceanic environment in a safe and cost effective manner, have promoted the development of maritime autonomous vehicles. Marine Autonomous Systems (MAS) have constituted the first examples of autonomous systems for environmental monitoring [1]. An overview of the main typologies of MAS is given in Section 1.1.

Beside introducing advantages from a scientific perspective, the use of autonomous vehicles for environmental monitoring can be of particular importance to control authorities, such as the Department for Environment Food and Rural Affairs (DEFRA) in the UK. The assessment of the spread of harmful spatial phenomena is crucial so that effective strategies for the protection of the environment and the reduction of hazardous phenomena can be devised. From an oceanic perspective, for example, the European Environment Agency (EEA) adopts a Driving force, Pressures, States, Impact and Response (DPSIR) model to understand the cause-effect relationships between interacting components of social, economic, and environmental systems [4]. In particular, the DPSIR model is used to determine the elements affecting Marine Protected Areas (MPAs), which are demarcated regions of the oceanic environment, and to develop management strategies. The assessment and the effective management of MPAs through autonomous vehicles is a desirable cost effective solution and it is the focus of the Marine Integrated Autonomous Observing Systems (MIAOS) project, launched in 2017 by the Natural Environment Research Council (NERC). The spread of contaminants due to leakage from point sources located in the ocean is also a very real threat. The tracer detection was identified as one of the five scenarios of interest during the Adaptive Autonomous Ocean Sampling Networks (AAOSN) project, promoted by NERC and the Defence Science and Technology Laboratory (DSTL) in 2015. This scenario was introduced to promote the development of techniques which use autonomous vehicles to determine the extension of a tracer patch and to identify the source position, representative of the leakage position.

The novel strategies presented in this thesis are classified as either source seeking, aimed at identifying the source of a spatial phenomenon, or boundary tracking, aimed at determining the extent of the phenomenon. Most of the work presented in this thesis is demonstrated in computer simulations, but the experimental validation of the considered boundary tracking strategy with an Autonomous Surface Vehicle (ASV) is also described.



(a) C-Enduro - ASV Limited

(b) Slocum glider - Webb Research Corporation

Figure 1.0.1: Marine Autonomous Systems

1.1 Marine autonomous systems (MAS)

MAS are under coming a fast development, with innovative design and control strategies being proposed at great pace. The two main classes of MAS are ASVs and Autonomous Underwater Vehicle (AUV)s. These are typically characterized by different designs, propelling systems and actuation technologies. ASVs can be monitored remotely, for instance through satellite observations, and can interact with on-shore control centres through many available communication technologies; consequently, the operation of ASVs is relatively safe. In order to sample three-dimensional underwater features, however, the deployment of underwater vehicles is preferred, even if their operation is associated with higher risks. The position and the status of AUVs while underwater, in fact, can not be monitored, as the communication link results unavailable.

An overview of the main typologies of MAS is given in Table 1.1.1. Propelled surface vehicles typically depend on a diesel engine and they are equipped with a battery pack as a secondary power source. Additionally, they are often equipped

Table 1.1.1: An overview of the main classes of MAS

Vehicle	**Reference**	**ASVs**	**AUVs**
Propelled surface vehicles	[5],	✓	
Wave propelled vehicles	[6], [7]	✓	
Propelled underwater vehicles	[8], [9]		✓
Underwater gliders	[10], [11], [12]		✓
Robotic fish	[13]		✓

with renewable energy sources, such as solar panels or wind turbines, for the battery recharge. Consequently, their endurance can be of the order of a few months. Wave propelled vehicles have a potentially longer endurance than the diesel engine propelled surface vehicles. Their peculiar structure, composed of a surfboard-like float attached to a submerged glider [6], allow them to use the ocean wave energy for propulsion. Their movement, however, is highly dependent on the ocean conditions and therefore the control possibilities are limited. Propelled underwater vehicles are an effective tool for exploring underwater features because of their low movement constraints. They are equipped with propellers which allow them to move freely in the three-dimensional space. Their endurance, however, is of the order of a few hours, as their movement relies entirely on the life of a set of batteries. This constitutes the main limitation associated with the deployment of this class of vehicles. Underwater gliders, whose movement is controlled through a buoyancy engine, have very long endurance, of the order of months. The buoyancy engine exploits the forces exerted by the surrounding fluid to propel the underwater glider, but the resulting movement is constrained to so-called saw-tooth trajectories. Finally, robotic fish mimic the movement of a real fish with a flapping and rotating tail for propulsion. Even this type of vehicles, relying on the power from a battery pack, has a relatively low endurance.

The autonomous vehicles considered in this thesis, which will be described in greater detail in the following chapters, are shown in Fig. 1.0.1: a propelled surface vehicle, C-Enduro [5], has been considered for monitoring two-dimensional features, and an underwater Slocum glider [11] has been considered for monitoring three-dimensional features.

1.2 Research scope and motivation

In recent years, the interest in using autonomous vehicles for monitoring and resolving hazardous spatial phenomena is increasing and significant research effort is being invested in this direction.

The Deepwater Horizon oil spill that took place in the Gulf of Mexico in 2010 is considered the largest marine oil spill in the history of the petroleum industry. The estimated amount of oil spilled during the leak, which lasted for approximately one month, is of 780000 m^3. This leak was also the cause of several fatalities and had a great impact on the marine environment, damaging the sea floor and affecting the survival of several species [14]. The oil spill also impacted the tourism industry, as well as the fish industry of the area, determining high economical costs [15]. In addition, the oil spill is believed to have affected the health of the people involved in the cleaning procedures [16]. The possibility of employing autonomous vehicles for the study and the resolution of catastrophic oceanic features of this sort is an active area of research, as it would avoid the direct involvement of human operators. An algorithm aimed at making an autonomous surface vehicle map the area interested by an oil spill is proposed in [3] and [17]. The method is based on the local and global navigation of the search space, which is divided into grid cells, and it deploys autonomous vehicles capable of absorbing the spilled oil while moving. Consequently, the efficient mapping of the oil spill coincides, in this case, with the efficient resolution of the problem.

Another example of hazardous oceanic feature is the spread of Harmful Algae Bloom (HAB). A recent example is the spread of toxic algal bloom which took place in Florida in July 2016. This was originated from a wetter than normal dry season (November - May) during 2015/2016. The Florida Department for Environmental Protection launched a project for the review of innovative algal bloom clean up technologies. Amongst the proposed technologies, several considered the deployment of autonomous vehicles for studying the algal bloom extension or for cleaning purposes. In [18], the results of an experimental work, deploying autonomous vehicles for monitoring HAB, are presented. In this work, autonomous vehicles were deployed for monitoring the spread of HAB in Singapore waters, where in 2009 a toxic bloom had great impact on the wildlife, leading to great economic losses. In the paper, ASVs and AUVs are deployed in order to collect high resolution measurements about the algal bloom. In order to design the pre-planned trajectory for the autonomous vehicles, some initial observations of the spatial phenomenon, collected through an Unmanned Aerial Vehicle (UAV), are exploited.

Different levels of autonomy, used to classify the types of operation of autonomous vehicles, are presented in [19]. Currently, autonomous vehicles are typically operated in a semi-autonomous way, as they are given a pre-planned target trajectory to follow. The dependence on a target trajectory can limit the effectiveness of the autonomous vehicles deployment, especially if the considered spatial feature is not perfectly known at the planning stage. Typically, rather conservative trajectories, such as lawnmower-type or zig-zag trajectories, are used, which can cause sampling of non significant areas and frequent manoeuvres, due to the swift changes in the desired direction of movement. Some preliminary works in which the initially designed target trajectory is adaptively updated during the vehicle operation have given promising results; see, for instance, [20].

The ultimate objective of autonomous vehicles deployment would be the fully autonomous operation of these vehicles. A fully autonomous vehicle would be equipped with the computation capabilities necessary to autonomously elaborate the collected measurements and to design the control actions necessary to achieve a pre-defined objective accordingly.

The work in this thesis is focussed on the development of techniques for monitoring and exploring unknown environmental features through the deployment of fully autonomous vehicles. The problems associated with environmental monitoring are posed from either a boundary tracking or a source seeking perspective. Novel source seeking and boundary tracking strategies for the application to two and three dimensional features are proposed. These make use of the local measurements collected by the vehicle, and do not require the a priori knowledge of the spatial features. Additionally, the proposed strategies do not require the pre-planning of a trajectory to be followed by the vehicle. The proposed algorithms, if equipped on board of an autonomous vehicle, would allow it to achieve the desired control objectives in a fully autonomous way. For this reason, an embedded system framework, with the potential of being installed on board of an ASV, has been developed. This would provide the ASV with the computation capabilities necessary to autonomously track the boundary of a two-dimensional spatial feature.

When additional information about the studied spatial phenomenon is available, however, this should be exploited in the control problem solution [1]. Thanks to the development of complex computer models of the environment, it is sometimes possible to obtain forecast information about the environmental feature under consideration prior to deploying the autonomous vehicles. This knowledge has the potential of being used not only to roughly determine the scope of the feature, but also at a high level planning stage. The advantages of making use of the available preliminary information are enumerated in [21]; these are the possibility of reducing operational costs and of minimizing the risks for the autonomous vehicles. This preliminary information has typically been used in the literature to design the preplanned trajectories [1], [22]. In this thesis, a probabilistic learning approach is proposed for the solution of the boundary tracking control problem for an imprecisely known environmental feature. Rather than planning the vehicle's trajectory on the basis of the available information, which is likely imprecise, a probabilistic model of the feature is built and iteratively updated through the inclusion of real world measurements. This iteratively updated model is shown to be an effective tool for the solution of the boundary tracking control problem, as it can be used in order to estimate the spatial gradient of the feature.

1.3 Contributions of the thesis

The main contributions of the thesis are summarized as follows:

1. A boundary tracking algorithm for three-dimensional underwater features, applied to an autonomous underwater glider: the approach is specifically de-

signed for controlling an underwater glider and tested on a complete dynamic model of a vehicle of this class in a set of computer based simulations. The direction of movement of the glider is defined through a suboptimal sliding mode control method, using only the measurement of the spatial feature at the glider's position. Knowledge of the vehicle position and of the gradient of the considered spatial feature is not required. The approach represents a preliminary step towards an enhanced autonomous operation of underwater gliders. Currently, in fact, underwater gliders are operated in a pre-planned way, as they are given a set of waypoints to reach.

2. A source seeking approach for an unknown static or dynamic two-dimensional spatial field: the approach makes use of a single autonomous vehicle, modelled as a kinematic unicycle and moving in the two-dimensional space. The vehicle is driven inside a neighbourhood of the source of the spatial field in finite time. The control strategy, based on the definition of a reference trajectory and on a second order sliding mode extremum seeking approach, makes use only the point measurement at the vehicle's position. The proposed approach is gradient-free and robust with respect to matched uncertainties. Bounds on the parameters defining the reference trajectory are introduced in order to guarantee the finite time convergence to a neighbourhood of the sought source.

3. A probabilistic learning boundary tracking approach: the exploration of an imprecisely known environmental feature with a formation of autonomous vehicles is considered. In contrast to more traditional approaches, the vehicles are not given a pre-planned trajectory, instead their control actions are designed online. The use of the forecast information is extensive: a probabilistic model of the feature is built initially and iteratively updated when real world measurements are made available by the autonomous vehicles. The vehicles, pursuing a formation boundary tracking objective, are controlled making use of the gradient of the spatial feature, estimated from the surrogate model.

4. The experimental validation of a boundary tracking algorithm with an ASV: the efficacy of the suboptimal sliding mode boundary tracking strategy is demonstrated through a staged validation procedure, deploying an autonomous surface vehicle, C-Enduro, in a set of sea trials. For the experimental validation, the seabed swath bathymetry is considered as a representative spatial feature and the ASV is controlled in order to track a constant depth contour.

5. An embedded system framework for the control of an autonomous vehicle: the boundary tracking and source seeking algorithms are implemented in an embedded system environment. This is developed for direct installation on board of an autonomous vehicle, particularly an ASV. The aim of the embedded system is to equip the autonomous vehicle with the on-board computation capabilities necessary to autonomously determine its trajectory in order to achieve a predefined objective.

1.4 Thesis organization

The thesis is organized in the following manner:

Chapter 2 begins with the definition of the boundary tracking and the source seeking control problems and briefly shows how the exploration of several oceanic features can be posed from one of these perspectives. The main boundary tracking and source seeking strategies available in the literature are then introduced, in order to define the scope of the work presented in this thesis.

Chapter 3 presents the suboptimal sliding mode boundary tracking algorithm, which represents the basis of the work in this thesis. The approach is applied to the study of two-dimensional static and dynamic oceanic features with an ASV. The approach is validated firstly on a numerical example and then on realistic datasets obtained from the Met Office, UK. Particularly, the results shown in Chapter 3 are focussed on the exploration of a sea surface temperature front in a tidal mixing area and on the determination of the extent of a tracer release patch.

Chapter 4 extends the boundary tracking algorithm presented in Chapter 3 to the exploration of three-dimensional oceanic features. The approach is applied to the control of an underwater glider, whose model, including its kinematics and dynamics, is introduced in the chapter. A guidance strategy is proposed, which influences the movement of the glider in the lateral plane, with the aim of tracking the boundary of a three-dimensional spatial feature. The results presented in the chapter are obtained in simulations, firstly on a numerical example and then using a realistic dataset, obtained from the Met Office, UK, relative to the water temperature in the Iceland Faroes front area.

Chapter 5 proposes a source seeking algorithm, aimed at solving the source seeking control problem with a single ASV. The approach is based on the generation of a reference trajectory for the measurements collected by the vehicle. A sufficient condition for the choice of the parameters defining the reference trajectory is introduced, which guarantees the finite time convergence of the vehicle to a neighbourhood of the sought source. The approach is shown valid when dealing with both static and dynamic spatial features. Even in this chapter, an initial set of numerical simulations is followed by a set of simulation results based on a realistic dataset, particularly relating to the tracer release scenario, also considered in Chapter 3.

Chapter 6 introduces the probabilistic learning boundary tracking approach applied to a formation of autonomous vehicles. The chapter describes the procedure followed to build a Gaussian Process (GP) model of the considered spatial feature and the chosen model validation strategy in detail. The fitted probabilistic model is then shown to be a useful tool in the estimation of the spatial gradient of the spatial feature, to be used in the boundary tracking control definition. The realistic dataset used in the chapter is the one describing the evolution of the sea surface temperature in a tidal mixing front area.

Chapter 7 describes the experimental validation of the boundary tracking control approach introduced in Chapter 3. The chapter begins with a detailed description of the used autonomous vehicle, the implementation strategy, and the experimental framework. The results of the sea trials demonstrate the efficacy of the approach. Chapter 7 finishes with an introduction to the structure of the embedded system, developed for the implementation of the proposed strategy and the possible future installation on board of the vehicle's control unit.

Finally, Chapter 8 provides some concluding remarks on the work presented in this thesis and highlights some of the possibilities for future extension.

1.5 Publications

The research work presented in this thesis has yielded the following publications:

1. Chiara Mellucci, Prathyush P. Menon, Christopher Edwards, Peter Challenor, *Source seeking with a single autonomous vehicle*, in Proceedings of the American Control Conference, Boston, 2016, pp. 6441-6446.

2. Chiara Mellucci, Prathyush P. Menon, Christopher Edwards, Peter Challenor, *Predictive oceanic features tracking with formations of autonomous vehicles*, in Proceedings of the Conference on Decision and Control, Las Vegas, 2016, pp. 6446-6451.

3. Chiara Mellucci, Prathyush P. Menon, Christopher Edwards, Peter Challenor, *Experimental validation of boundary tracking using the suboptimal sliding mode algorithm*, in Proceedings of the American Control Conference, Seattle, 2017, pp. 4878-4883.

4. Chiara Mellucci, Prathyush P. Menon, Christopher Edwards, Peter Challenor, *Oceanic Feature Boundary Mapping with an Autonomous Un-*

derwater Glider, in Proceedings of the American Control Conference, 2018, pp. 5338-5343.

5. Chiara Mellucci, Prathyush P. Menon, Christopher Edwards, Peter Challenor, *Environmental feature exploration with a single autonomous vehicle*, under review for publication to the IEEE Transactions on Control Systems Technology, 2018.

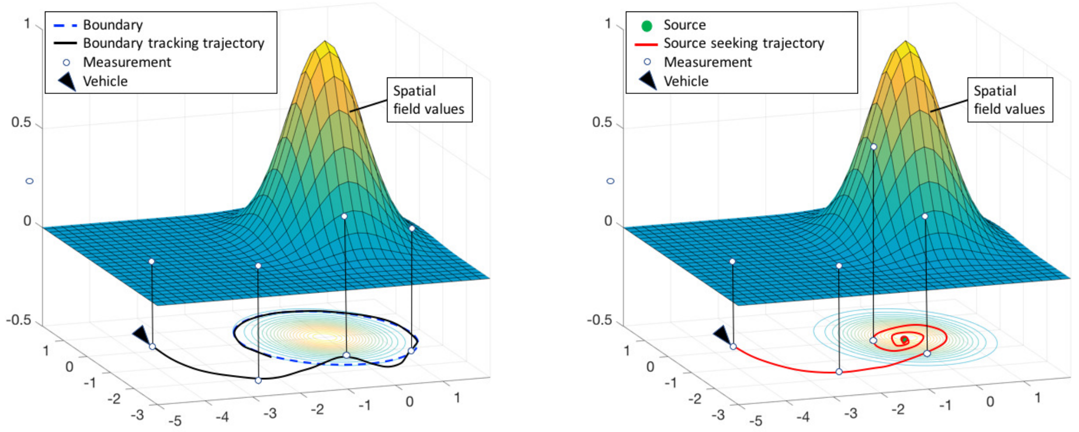
6. Chiara Mellucci, Prathyush P. Menon, Christopher Edwards, Peter Challenor, *Mapping an imprecisely known environmental feature with autonomous vehicles: a probabilistic learning approach*, submitted to the International Journal of Systems Science, 2018.

Chapter 2

Boundary tracking and source seeking methods - a review

2.1 Introduction

The exploration and monitoring of environmental features is being increasingly addressed through the deployment of autonomous vehicles, rather than using fixed sensors. The long endurance and low operational costs of autonomous vehicles allow sampling and monitoring of the regions of interest with high spatial and temporal resolution [1]. In addition, autonomous vehicles have the potential to efficiently perform demanding tasks in harsh environments without human supervision [23]. Current commercially available autonomous systems, however, possess little or no on-board intelligence and they are traditionally pre-programmed to follow predefined trajectories. These predefined trajectories are suboptimal, as they are planned a-priori, using the available preliminary information about the feature of interest, likely imprecise. The most common types of preplanned trajectories belong to the lawnmower or zig-zag typologies [24]. The wish to enhance the autonomy of these vehicles is modifying the way in which environmental features are explored: from pure surveys, completed following pre-programmed trajectories, more recent works consider adaptive control strategies [20] and the online definition of the vehicle trajectory.



(a) Boundary tracking

(b) Source seeking

Figure 2.1.1: Schematic of the boundary tracking and source seeking control probelms

The problems addressed in this thesis are the boundary tracking and the source seeking problems in environmental monitoring using autonomous vehicles.

In the *boundary tracking* problem, the autonomous vehicle is required to move along the boundary of the feature of interest. The boundary is identified as the locus characterized by a constant value of the monitored quantity, typically representing a safe level, such as a threshold contamination level associated with the mixing of pollutants in a medium [25]. A schematic visualization of the boundary tracking control objective is given in Fig. 2.1.1a.

In the *source seeking* problem, the autonomous vehicle is required to move towards a neighbourhood of the source, where the environmental feature has its local minimum or maximum. Neither the value of the spatial feature at the source location, nor the location of the source are known a-priori. A schematic visualization of the source seeking control objective is depicted in Fig. 2.1.1b.

Specific problems of interest in ocean sampling may be posed as either boundary tracking or source seeking problems.

The mapping of the seabed, which is aimed at gaining knowledge about its conformation, can be posed as a boundary tracking or a source seeking control problem.

As an example, in [26], the boundary separating areas of the seabed with different composition and habitats is tracked. The swath bathymetry study is aimed at identifying characteristic features of the seabed, such as underwater canyons [27]. This type of survey can be posed as a boundary tracking problem, in which the vehicle is requested to follow a trajectory characterized by a constant value of the water depth. Alternatively, the seabed swath bathymetry can be approached from a source seeking perspective when the point of maximum or minimum depth in a certain area is searched [28].

The monitoring of pollution phenomena, such as oil spills [3], [17] or the outflow plumes of industrial plants [20] can also be posed as either a boundary tracking or a source seeking control problem. If the extension of the polluted area needs to be determined, the study of the pollution phenomenon can be posed as a boundary tracking problem. In [20], as an example, the tracked boundary separates clean waters from waters affected by a nuclear plant thermal plume. Alternatively, if the source of the pollution phenomenon is unknown, this can be identified by following a source seeking approach; in [29], as an example, the source of a chemical plume is identified through a source seeking algorithm.

An HAB is an area of natural waters in which possibly harmful algae grow in a higher than normal concentration. The requirement to monitor the extension of the area affected by HAB can be interpreted as a boundary tracking problem. Choosing the threshold that identifies an harmful level of algal concentration (typically a known value), a boundary tracking algorithm can be used to enable an autonomous vehicle to track the boundary of the phenomenon, in order to isolate the interested area [30].

The dye release scenario, in which an inert tracer is released in the ocean and the evolution of its patch is studied, is usually aimed at determining marine currents [31], [32]. The tracer release may be monitored with autonomous vehicles from a boundary tracking perspective, if the extension of the patch and the changes in its shape are of interest, or from a source seeking perspective, if the point of maximum concentration of the tracer is sought.

Finally, the tidal mixing front exploration may be posed as a boundary tracking problem. A tidal mixing front is the area of the ocean where tidally mixed and seasonally stratified shelf waters, having different physical and chemical properties, encounter [33]. In this case the boundary can be identified as an isoline of the spatial field, that is the locus where a quantity of interest, such as the water temperature or salinity, has a constant value.

The recent literature, see for instance [26], [29], [20], [34], [35], [23], is rich of different, yet not optimal in all aspects, boundary tracking and source seeking methodologies which are proposed to study different environmental features using autonomous vehicles. These examples span all areas of environmental monitoring, exploiting ASVs or AUVs, UAVs or Unmanned Ground Vehicle (UGV)s. The most significant boundary tracking and source seeking methodologies are reviewed in Section 2.2 and Section 2.3 respectively.

2.2 Boundary tracking methodologies

A non-comprehensive review of the main boundary tracking methodologies is listed in Table 2.2.1. The criteria for the classification of the boundary tracking methodologies are the number of autonomous vehicles used and the static or dynamic nature of the environmental boundary.

Table 2.2.1: Principal boundary tracking methodologies

Method	**Reference**	**Single**	**Multiple**	**Static**	**Dynamic**
Adaptive lawnmower	[20]	✓			✓
Image processing	[26], [36]	✓		✓	
Reactive control	[37], [38]	✓		✓	✓
Sliding mode	[35], [39]	✓		✓	
Bang-bang	[40], [41], [42], [34]		✓	✓	
Hybrid coordination	[43], [44]		✓	✓	✓
Glowworm swarm	[45], [46]		✓	✓	
Boundary approximation	[30], [47], [48]		✓	✓	✓
Rigid formation	[49], [50]		✓	✓	

2.2.1 Boundary tracking with a single autonomous vehicle

Obviously, boundary tracking algorithms requiring the deployment of a single autonomous vehicle are economical from the perspective of the costs associated with the sensors, the communication requirements and the deployment and recovery [37]. One possible drawback associated with the deployment of a single autonomous vehicle is the difficulty in collecting distributed measurements, which may render the solution of the control problem more complex.

Adaptive lawnmower trajectory method

In [20], the study of the thermal outflow from the Calvert Cliffs Nuclear Power Plant in Lusby, MD, USA is considered from a boundary tracking perspective, using a single ASV. The proposed strategy is based on a plume indicator function, which is a function of the water temperature, salinity, dissolved oxygen, and flow magnitude measurements, and determines the likelihood of whether the vehicle's position belongs to the outflow plume. Traditional oceanic sampling is based on lawnmower-type trajectories [24], composed of a series of transects in the presumed area of the oceanic feature. Each transect has the same length and direction, and the distance between any two subsequent transects is constant. The boundary tracking approach in [20] is based on the design of adaptive transects. The length and the direction of each transect, as well as the distance between consecutive transects, are designed in real time in order to make each transect orthogonal to the main direction of the plume and crossing the plume boundary. The resulting trajectory, which is generated in real-time, is similar to a lawnmower type one, but tracking accuracy is enhanced as only the region around the boundary is sampled. An example visualization of a traditional lawnmower trajectory in comparison with an example adaptive trajectory is shown in Fig. 2.2.1. In [20], the results from a set of field trials, conducted with a small ASV, are presented. The main advantage of this strategy over traditional lawnmower approaches is the swift mapping of the feature, and consequently the reduced operational costs. However, the approach lacks tight tracking of the boundary, which is often preferred.

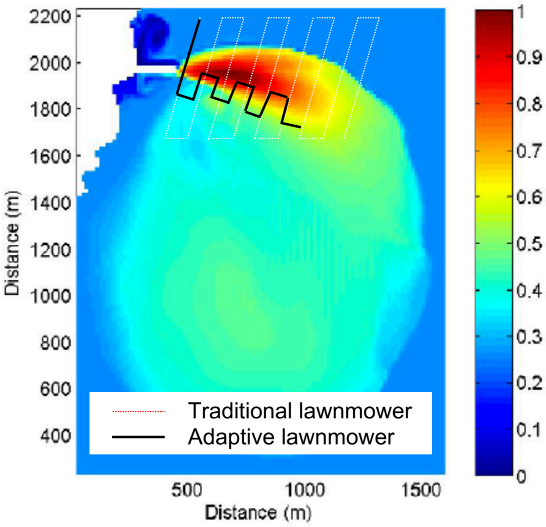


Figure 2.2.1: Examples of a traditional and an adaptive lawnmower trajectory - adapted from [20]

Image processing methods

A different class of boundary tracking methods applies image processing techniques to the boundary observations collected by the autonomous vehicles. In [26] a solution which deploys a remotely controlled AUV, moving at a constant depth, is proposed to track the boundary between different sea bed habitats and validated in simulations. The vehicle is equipped with a sonar profiler, whose measurements are processed through an image segmentation algorithm. The segmentation algorithm is used to classify the location of the vehicle as inside or outside the tracked boundary. The sonar profiles, in fact, have different shapes depending on the material composing the sea floor. This information is used to subsequently apply the appropriate steering control.

In [36] the possibility to track the boundary of an oceanic surface feature, for instance a petrol stain on the sea surface, with a single UAV equipped with a camera is investigated in simulations. The approach is based on image processing techniques, which process the incomplete boundary images collected through the on-board camera in order to design the next Waypoint (WP), a latitude and longitude stamp to be reached by the vehicle.

The main disadvantage of the boundary tracking methods based on partial observations is the dependence on complex image processing techniques, which require the vehicle to be equipped with a high level of computation capability. In addition, these approaches have only been validated in computer based simulations. Practical difficulties, such as the time required by the image processing techniques to design the steering law or the WP and the battery requirements, should be addressed in experimental frameworks. The measurements collected with these strategies, anyway, allow a deep understanding of the considered features.

Reactive control

The reactive boundary tracking control proposed in [37] is characterized by a low information requirement. The spatial field measurement at the vehicle's position and the time derivative of the spatial field measurements collected along the vehicle's trajectory are required. The time derivative is estimated using the collected measurements. The autonomous vehicle is considered as a reactive agent, which is controlled through its angular velocity in reaction to the characteristics of the field. The angular velocity of the vehicle is modified through a proportional controller, making use of the measurement at the vehicle's position, the value of the spatial field on the tracked boundary and the estimated rate of change of the collected measurements. The proposed controller ensures that the vehicle tracks the desired boundary, provided certain constraints on its curvature are verified. The ultimate objective of the proposed approach is to obtain a vehicle trajectory having the same curvature of the tracked boundary. One limitation of the approach, which has been relaxed in [38], is the hypothesis on the initial orientation of the vehicle, which should be sufficiently aligned with the tangent to the tracked isoline. Additionally, the performances of the approach are higher in the ideal case, when the curvature of the tracked isoline is constant. In the case of a non constant curvature, the tracking error is shown to be directly proportional to the difference between the actual curvature of the isoline and the ideal one. The hypothesis of a constant curvature, however, is unrealistic when considering real-world spatial fields. The approach is validated in simulation firstly on the ideal case of a completely radial spatial function and secondly on a more realistic type of spatial field. Boundary tracking is achieved in both cases, but the tracking accuracy is significantly reduced if a spatial field whose isolines are characterized by a non-constant curvature is considered. As a final set of results, tracking of the boundary of a diffusing field is considered, but the consequences of having a time-varying spatial field on the design parameters are not formally evaluated.

Sliding mode control methods

A boundary tracking method based on sliding mode control ideas is proposed in [35] for the control of a single autonomous vehicle modelled as a kinematic unicycle [51]. A single vehicle, moving at constant speed, is considered and its angular velocity is controlled. The angular velocity controller obeys to a switching logic which sets the vehicle's angular velocity to either its minimum or maximum value. The information requirement for the control action definition is limited to the measurement of the spatial feature at the vehicle's position and the rate of change of the collected measurements, which is estimated. The approach is inspired by sliding mode control techniques, as the difference between the measured and the tracked value of the spatial field is treated as the sliding variable, to be reduced to zero in finite time - the reader is referred to Appendix A for an introduction to sliding mode control. The main advantages of this method are the non-local convergence to the tracked contour and the low computation requirements. The non-local convergence is the capability to reach the tracked contour independently of the initial position of the vehicle relative to the tracked contour itself. The approach, however, lacks robustness to measurement noise, which may significantly worsen the tracking performances. This limitation can be partly overcome by averaging the collected measurements. Additionally, the approach is characterized by chattering, which is the high frequency switching in the control signal. This dangerous phenomenon is showed to be successfully reduced through the boundary layer technique [52]. Additionally, other chattering avoidance techniques, such as higher order sliding mode techniques [53], may be considered.

Another sliding mode based boundary tracking algorithm is proposed in [39]. The proposed suboptimal sliding mode controller modifies the angular velocity of an UAV moving at a specific height, in order to track the boundary of a spatial phenomenon. The main advantage of this approach is the reduced information requirement, with respect to other boundary tracking methods as well as the work in [35]: the local knowledge of the spatial field is the only information required for the control law definition. The approach is validated in simulations on a realistic dataset relating to the ash cloud resulting from the Eyjafyallajökull volcano eruption in 2010. Part of the work in this thesis builds on the boundary tracking algorithm originally proposed in [39], which is presented in detail in Chapter 3 and used in a novel application. Particularly, the movement of an ASV for the ocean exploration is controlled through the proposed method for the study of a set of oceanic features. The experimental validation of the approach with an ASV, additionally, constitutes one of the contributions of this thesis and it is described in Chapter 7.

2.2.2 Boundary tracking with multiple autonomous vehicles

The deployment of a formation of vehicles for the environmental boundary tracking is typically considered for two main reasons: the collection of distributed measurements and the distribution of the agents along the boundary. The distributed measurements, in fact, are useful for the estimation of higher order information about the spatial field, such as the spatial gradient [49]. When the vehicles are distributed along the boundary, instead, a set of simultaneous observations is available to be used for the real time boundary estimation [41], [34], [48]. The deployment of a formation of agents, however, have disadvantages in terms of operational costs and communication constraints [35].

Bang-bang controller

The bang-bang like boundary tracking strategy is "*perhaps the simplest tracking al-*

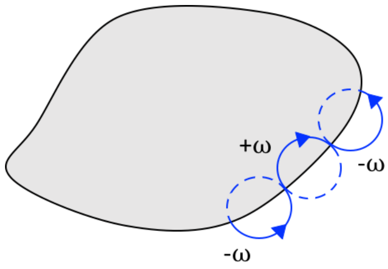


Figure 2.2.2: Boundary tracking using a bang-bang control scheme - adapted from [40]

gorithm" [41] and certainly one of the few which has been validated throughout. This gradient-free approach has been initially proposed in [40], where a two-dimensional static boundary is tracked with a formation of vehicles. Each vehicle performs boundary tracking independently and it coordinates with the others, by modifying its speed, in order to avoid collisions. The sign of the angular velocity of each vehicle, and hence the direction of movement along a circular path, is changed every time the tracked boundary is crossed, through a bang-bang control logic:

$$\dot{\theta} = \begin{cases} +\omega & \text{when inside the boundary} \\ -\omega & \text{when outside the boundary} \end{cases} \tag{2.2.1}$$

where θ is the heading of the vehicle and ω is the angular velocity. This approach is schematically represented in Fig. 2.2.2. In addition to the bang-bang control, the coordination logic modifies each vehicle's speed as a function of the proximity to neighbouring vehicles. The introduced interaction mechanism has the effect of avoiding collisions and spreading the agents evenly along the boundary [40]. This approach has been experimentally validated in [42], where UGVs have been deployed in order to track a virtual geometric boundary. Virtual sensors are employed, which use the position measurement of the vehicle in order to determine if the vehicle is inside or outside the virtual boundary, and command the angular velocity accordingly. One limitation of the approach in [40], as highlighted in [42], is the lack of robustness with respect to measurement noise. In the presence of a high level of measurement noise, in fact, the evaluation of the boundary crossing may result delayed and false crossings may be detected. The approach in [40] is further developed in [41], where cumulative sum (CUSUM) filters [54] are employed to render the boundary crossing evaluation robust to measurement noise. The aims of CUSUM filters are to minimize the difference between the estimated and real crossing time, and to minimize the number of false crossing reports. The approach, enriched with the CUSUM filters, is validated in simulations on a time-varying ellipse-shaped boundary, which can be successfully tracked assuming that the vehicle's speed is higher than the boundary speed. The algorithm is further tested in [34] through an experiment deploying autonomous wheeled vehicles following a static boundary, created through a tape path on the floor. The main advantage of this strategy is the simplicity of the control definition. This approach, however, fails if the turning radius of the vehicle is greater than the minimum radius of curvature of the tracked contour. This, when tracking an a-priori unknown boundary, makes the choice of the vehicle's speed and angular velocity critical.

Hybrid coordination algorithm

In [43] and [44], a decentralized hybrid coordination algorithm is proposed, which allows a dynamic boundary to be monitored through a network of mobile agents. The approach is decentralized, as each vehicle defines its control action independently of the others in the formation. It is assumed that the tracked boundary is moving slower than the vehicles' maximum speed. Each vehicle has a limited field of view, in which it is able to sense the presence of other vehicles, and a limited communication range. Boundary tracking is achieved exploiting a hybrid controller, composed of: (i) the *random coverage controller*; (ii) the *potential field controller*; (iii) the *tracking controller*. The underlying switching logic is depicted in Fig. 2.2.3. The *random coverage controller* is aimed at covering as large an area as possible while searching the tracked boundary. Each vehicle performs a logarithmic spiral search until the tracked contour is detected. After the contour detection, the vehicle switches to the *tracking controller* and sends its location to the vehicles in its communication range. The tracking controller regulates the angular velocity of

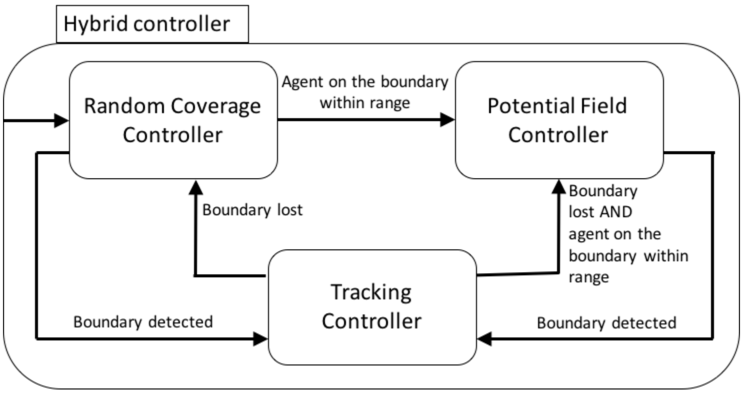

Hybrid controller
Random Coverage Controller
Agent on the boundary within range
Potential Field Controller
Boundary lost
Boundary lost AND agent on the boundary within range
Tracking Controller
Boundary detected
Boundary detected

Figure 2.2.3: Hybrid boundary tracking switching logic - adapted from [44]

the vehicle through a bang-bang strategy, similarly to [40]. In addition, the linear velocity is modified in order to avoid collisions with other vehicles on the boundary. When the i-th vehicle is within a certain range from a vehicle on the boundary, it receives the location of the vehicle on the boundary through a communication link and it is controlled through the *potential field controller*. In particular, it is subject to an attractive force, function of its distance from the position of the vehicle on the boundary. One drawback of this strategy is the use of the random coverage control, which is a suboptimal strategy to find the tracked boundary, characterized by unnecessary manoeuvres and long distance movement. In addition, being the tracking controller based on a bang-bang strategy, tracking is lost if the vehicle radius of curvature is greater than the minimum radius of curvature of the boundary, as in [40]. In this case, however, the random coverage controller is restored and tracking is recovered.

Glowworm swarm algorithm

In [45], a modification of the Glowworm Swarm Optimization (GSO) algorithm [55] is proposed to simultaneously achieve boundary tracking and source seeking over a two-dimensional region. The GSO, in fact, was originally developed to detect multiple optima of a function with multiple robots. In this application a *swarm* composed of a large number of agents is subdivided with the tasks of seeking the local sources, according to the classical GSO algorithm [55], and of tracking the boundary, according to a modification of the GSO algorithm. Each boundary tracking agent moves depending on the information exchanged among its neighbours, which are the agents located in the sensing/communication range. At each location where measurements are taken, each agent has a suitably scaled value of the measurement, the so-called luciferin value. Each agent communicates its luciferin value with the neighbouring agents. Based on this information, following a probabilistic rule, each agent identifies a leader, typically the neighbour with the highest luciferin value, and then moves towards it with a step movement. Once the boundary tracking agents reach the tracked boundary, their luciferin value is set to zero and they start acting as a reference for other agents approaching the boundary. When a boundary tracking agent identifies as the leader an agent already on the boundary, its movement results from a trade-off between the attraction towards the leader and the repulsion, necessary to maintain a minimum distance between agents. In [46] the algorithm presented in [45] is extended to track a static three-dimensional boundary. The main advantage of this method is the possibility of achieving both the boundary tracking and the source seeking control objectives simultaneously, while one limitation of the approach is the required number of agent to be deployed.

Boundary approximation

The boundary tracking methods proposed in [30] and [47] adapt the snake algorithm for image processing [56], with the aim of moving a formation of agents onto the tracked boundary. A *snake* is defined as a curve which quickly adapts around the boundary of the considered shape. The snake algorithm is aimed at minimizing an energy function, whose minimum corresponds to the snake which better approximates the unknown boundary. The snake is described through a number of points equal to the number of deployed agents, whose positions identify the desired positions of the agents. The energy is defined in terms of a potential function, which, in the case of a boundary tracking problem, is a negative function of the spatial gradient of the feature. Minimizing the energy function therefore drives the snake towards sharp gradients and consequently towards the tracked boundary. For the solution of the minimization problem, knowledge of the spatial gradient at each agent position is required and hence each vehicle is assumed equipped with a sensor for the spatial gradient estimation. In addition, global communication is required. Despite the approach is proven successful in simulations, even in the presence of communication losses, the hypothesis on the availability of gradient estimates is strong. This hypothesis is relaxed in [47], where the definition of the energy function is modified. The main disadvantage of these algorithms, however, remain the high requirement in terms of both computation capabilities and communication between agents.

A similar approach is proposed in [48], where an algorithm to optimally approximate a slowly-moving boundary with a polygon of interpolation points, located by a formation of vehicles, is proposed. Through the proposed *estimate, update and pursuit algorithm*, the interpolation points and the agents uniformly distribute on the boundary. Each agent, while moving along the tracked boundary, estimates the shape of the boundary, i.e. its tangent and its curvature. In addition, each vehicle updates its position and that of the interpolation points, whose number is assumed higher than the number of agents. The boundary can then be reconstructed by linear interpolation of the interpolation points. Finally, each vehicle estimates the arc length distance between himself and the closest neighbours and modifies its speed accordingly, in order to avoid collisions and uniformly distribute along the boundary. The hypothesis on the availability of an initial estimation of the boundary configuration and on the capability of each vehicle of estimating both the boundary tangent and curvature may be considered as a drawback of this approach.

Rigid formation based approach

In [49] the vehicles are assumed to move in a rigid formation. The collected measurements, whose relative positions are known, are used to build an estimate of the spatial field gradient and Hessian. The gradient information is used to determine the direction of maximum variation of the field, while the Hessian information is used to estimate the contour curvature. The boundary tracking strategy is implemented to control the movement of the centroid of the vehicle formation. The angular velocity of the centroid is controlled through a steering law defined according to the estimated spatial field gradient and curvature. The vehicles in the formation follow the movement of the centroid, in order to move along the tracked contour, while maintaining the desired rigid formation around the centroid itself. The approach is validated in simulations considering a static two-dimensional field and it is extended to three-dimensional fields in [50]. The necessity to estimate both the spatial gradient and the Hessian may be seen as a limitation of this approach, as high computation capabilities and a stable communication link between the agents are required.

2.3 Source seeking methodologies

Most of the source seeking algorithms in the literature exploit the knowledge, or an estimate, of the gradient of the monitored quantity: in order to reach the maximum of the spatial field, in fact, the most immediate approach is to follow the direction of the gradient of the spatial field itself. A non-comprehensive review of the main source seeking methodologies in the literature is given in Table 2.3.1. The methodologies are classified in terms of the gradient dependency, the number of vehicles and the static or dynamic nature of the considered spatial feature.

2.3.1 Source seeking with a single autonomous vehicle

The advantages of deploying a single autonomous vehicle highlighted in Section 2.2.1 remain true for source seeking applications. Gradient-based algorithms deploying

Table 2.3.1: Principal source seeking methodologies

Method	**Reference**	**Gradient**	**Single**	**Multiple**	**Static**	**Dynamic**
Hybrid control	[57], [58]		✓		✓	
Extremum seeking	[59], [60], [61], [62], [63], [64]	✓	✓		✓	
Sliding mode	[65], [66], [67]		✓		✓	✓
Leader centred	[68]	✓		✓	✓	
VBAP	[69], [70], [71]	✓	✓	✓	✓	
Stable formation	[72]	✓		✓	✓	
Glowworm swarm	[55], [45]			✓	✓	

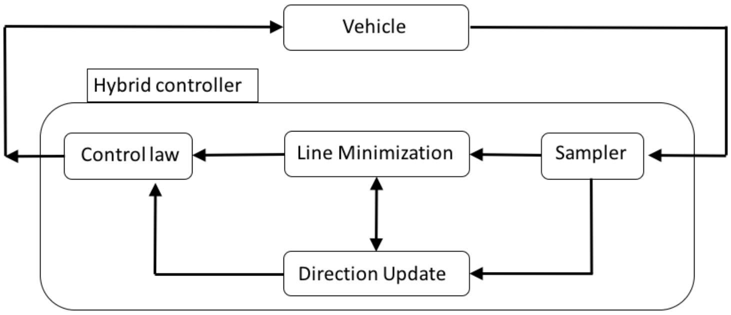


Figure 2.3.1: Hybrid source seeking switching logic - adapted from [57]

a single autonomous vehicle, however, are often associated with complex and frequent manoeuvres, required to collect spatially distributed measurements, necessary for the gradient estimation. Gradient-free approaches, consequently, are subject to high research interest.

Hybrid control

A single vehicle gradient-free source seeking algorithm, based on a series of line minimizations, is proposed in [57], [58]. The approach is based on a hybrid system, which ensures convergence to a neighbourhood of the sought source; a schematic of the proposed hybrid system is given in Fig. 2.3.1. The vehicle is assumed to move in a certain direction until a minimum/maximum of the considered spatial field along that direction is determined. This is achieved isolating a bracket, which is an interval known to contain a local extremum, and then finding the minimum/maximum belonging to the bracket. Once this is identified, the direction of movement is changed and the new direction is chosen amongst the conjugate directions to the current direction [73]. The main advantage of this method is the low information requirement, as the control law is designed using only the measurements collected by the vehicle. One limitation of the approach is the requirement of abrupt changes in the direction of movement of the vehicle, which follow each update in the direction of movement. In addition, complete turning manoeuvres are necessary in order to identify the minimum inside a certain bracket.

Extremum seeking control

"*Extremum seeking is a non-model based real-time optimization approach for dynamic problems where only limited knowledge of a system is available*" [74]. In extremum seeking, the maximum/minimum point of an unknown function is sought. In the case of environmental features, this corresponds to the point where the measured quantity reaches its extremum. The quickest way to reach the extremum of an unknown feature is by following the direction of the spatial gradient [59], which indicates the direction of maximum intensity variation of the field. This, however, is often unknown and needs to be estimated. Consequently, much of the extremum seeking literature is based on gradient estimation techniques. Gradient estimates are built by collecting spatially distributed measurements along the vehicle's trajectory. These are obtained by deliberately altering the vehicle's speed or direction of movement or by dithering the position of the sensor. A basic schematic of an extremum seeking controller is given in Fig. 2.3.2, where $\gamma(x, y)$ is the measurement at the vehicle's position and $\eta(t)$ is the introduced perturbation signal. The washout filter is a high-pass filter used to isolate the variation in the output. The introduced perturbations usually belong to two main categories: slow periodic sinusoidal perturbations [59], [61], or stochastic terms [62], [63], [64]. As an example, in [61] the speed of the vehicle is modified through sinusoidal perturbations, while a similar approach is used in [59] to control the angular velocity of the vehicle; this approach is also extended to the three-dimensional space in [60]. Sinusoidal perturbations have the limitation of being uniformly bounded, and this may restrict the region probed by the autonomous vehicle and therefore the region of attraction of the approach [74]. Consequently, the possibility to control the angular velocity or the speed of the vehicle through a stochastic extremum seeking approach has been considered in [63] and [64] respectively. In these approaches, the perturbation signals are typically white noise signals processed through a low-pass filter.

The main advantage of extremum seeking techniques is that they are "*truly model free*" [75] and have proven to be robust and effective in a high number of different application fields. One limitation of these techniques is their local behaviour: in the

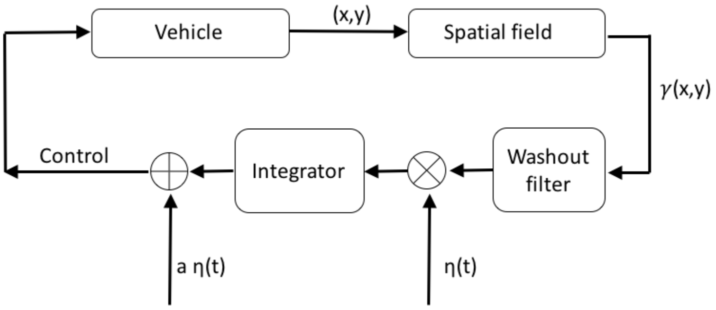


Figure 2.3.2: Schematic of a basic extremum seeking control - adapted from [74]

presence of multiple local extrema, the global extremum may not be found. More importantly, when controlling an autonomous vehicle, extremum seeking techniques are associated with frequent and fast manoeuvres, which may result in high operational costs.

A gradient-free extremum seeking approach, based on sliding mode ideas, is proposed in [65]. During sliding, the measurements collected by the vehicle follow a monotonically increasing reference trajectory $\gamma_{ref}(t)$, until the neighbourhood of the sought source is entered. The reference trajectory defines the desired evolution of the measurements at the vehicle's position, and it is defined as a strictly monotonically increasing function in [65]. Sliding mode techniques are used in order to reduce the *sliding variable* to zero in finite time and to subsequently maintain the sliding motion. By defining the sliding variable as the difference between the desired measurements, given by the reference trajectory, and the actual measurements at the vehicle's position, source seeking is achieved in a controlled way. The main advantage of sliding mode based extremum seeking techniques is the reduced information requirement, limited to the measurements at the vehicle's position. Gradient information is not required and this highly reduces the operational costs associated with these methods, as the deliberately introduced manoeuvres necessary in traditional extremum seeking techniques are avoided. The main difficulty associated with these techniques is the definition of the reference trajectory: badly tuned reference trajectories may cause the failure of the source seeking method.

The source seeking approach proposed in Chapter 5 is based on sliding mode extremum seeking techniques and on the definition of a reference trajectory. Explicit bounds on the parameters of the reference trajectory are defined in order to ensure convergence to a neighbourhood of the source of a static or dynamic two-dimensional spatial field.

Sliding mode source seeking control

A single vehicle, gradient-free source seeking approach is proposed in [66]. This approach is based on sliding mode ideas and guarantees the finite time convergence of the vehicle to a neighbourhood of the spatial field source. The main advantage of this approach is the low information requirement, as only the collected measurements at the vehicle's position and the estimated rate of change of the measurements are used. The approach controls the angular velocity of the vehicle through a bang-bang like controller. The sign of the angular velocity is varied every time the difference between the estimated rate of change of the vehicle's measurements and the desired growth rate changes sign. The desired growth rate defines the desired rate of change of the collected measurements in the movement towards the source. The choice of this design parameter is critical: an excessively big value can make the approach fail. An upper bound on this parameter, function of an estimate of the spatial field derivatives, is proposed. The main limitations of this approach, as highlighted in [67], are the dependence on the measurements' derivative estimate, which may results unreliable in the case of noisy measurements, and the switching nature of the used controller. The approach proposed in [67], inspired by the work in [38], overcomes these limitations. The proposed controller acts directly on the vehicle's heading and reduces the information requirements, as only the point measurement at the vehicle's position is required, without necessity to estimate the rate of change of the measurements. The advantages of the proposed approach, based on a proportional controller, are the design simplicity and the capability to tackle both static and dynamic fields.

2.3.2 Source seeking with multiple autonomous vehicles

Most of the source seeking approaches deploying a formation of vehicles aim at collecting spatially distributed measurements for the estimation of the spatial gradient of the feature. The main disadvantage of these strategies is the communication requirement; consequently, gradient-free approaches are being developed.

Leader centred approach

In [68], a group of vehicles is steered towards the extremum of an unknown spatial field while moving in a prescribed formation. The leader of the formation performs source seeking independently, through an extremum seeking technique. The leader collects spatial field measurements along a dithered trajectory, particularly around a closed triangular path. The leader then constructs an estimate of the field gradient and Hessian information through the finite difference method and uses this information in order to determine the next desired position, closer to the sought extremum. Its velocity is then oriented towards the desired position. The remaining vehicles apply passivity-based coordination rules in order to follow the leader, while maintaining the desired formation. The coordination control strategy proposed in [76] is exploited in order to maintain the desired formation, while following the position of the leader by estimating its velocity [77]. In the approach in [68], only the designated leader needs to have sensing capability and the collected measurements are not communicated to other vehicles. Consequently, this approach is advantageous in applications where the sampling and communication processes are expensive. The main disadvantage of this approach is the total dependence on the leader's capability to accurately collect measurements, estimate the gradient direction and move accordingly. In addition, as the leader is capable of performing source seeking autonomously, the formation could be omitted.

Virtual body and artificial potential approach

In [70], each vehicle in the formation is assumed capable of estimating the spatial gradient at its position and it is controlled accordingly, in order to move towards the sought source. In addition each vehicle is subject to control forces derived from the artificial potentials, which maintain the vehicle in a desired formation. Particularly, vehicles are attracted when they are far from each other and they are repelled if they are very close [69]. The force applied to each vehicle depends on the interactions with all the other vehicles inside a certain range. One advantage of this approach is the robustness with respect to a vehicle failure: the artificial potential strategy is in fact independent of the number of vehicles. This approach is extended in [71], where the concepts of virtual body and artificial potentials are combined. The virtual body is a collection of linked, moving reference points, called virtual leaders, that can perform translations and rotations. The formation moves with the virtual body: the driving forces acting on each vehicle are derived from the artificial potentials interactions with the other vehicles in the formation and the virtual leaders. Consequently, the artificial potentials are exploited in order to maintain the desired formation and to make the formation follow the movement of the virtual body. The movement of the virtual body is controlled through a centralized computation logic, whose aim is to perform gradient climbing. The centralized computation logic uses the measurements of the vehicles in the formation to estimate the spatial gradient of the field through a least-square optimization method and accordingly updates the state of the virtual body, which is then communicated back to each vehicle. This approach, despite maintaining the advantages of the approach [70] in terms of sensing simplicity and robustness to vehicle failure, results centralized and highly dependent on communication, which is assumed synchronized and continuous.

Stable formation approach

In [72], the measurements of a group of vehicles in a stable circular formation is exploited for the gradient estimation. The formation is enforced through the decentralized control approach proposed in [78], which uniformly distributes the vehicles in a circular formation and allows them to follow the movement of the centroid. The proposed source seeking approach assumes the stable formation is successfully maintained and defines an outer loop source seeking control. The outer loop control acts on the position of the centroid of the formation and moves it in the direction of the estimated gradient. The main advantage of this approach is the simplicity of the centroid control definition. One possible drawback of this approach is its centralized nature: in order to control the movement of the centroid, the measurement of each vehicle in the formation is required and this may be costly in terms of communication requirements.

The strategy proposed in Chapter 6 to perform cooperative boundary tracking is based on similar concepts. In particular, the stable formation is assumed maintained through the approach proposed in [78] and the centroid of the formation is controlled exploiting the estimated gradient. Differently from [72], the approach proposed in this thesis estimates the spatial gradient of the feature by fitting a probabilistic model to the feature and uses the measurements collected by the vehicles in the formation to enhance the accuracy of the fitted model.

Glowworm swarm optimization approach

The glowworm swarm approach previously discussed in Section 2.2.2 is used in order to achieve source seeking with a swarm of agents in [55] and [45]. The movement of each agent associated with a source seeking task is based purely on local information and, in particular, a representative scaled value of the actual measurement (the luciferin value), which the agents inside a sensing radius communicate to each other. At each update instant, each vehicle determines a leader, likely an agent with a higher luciferin value, through a probabilistic mechanism. Each agent then updates its position moving towards the selected leader. One advantage of this source seeking approach is the possibility to simultaneously locate several local maxima of the considered spatial fields. As previously mentioned, the main disadvantage of this method is the requirement of several agents.

2.4 Concluding remarks

A review of the principal boundary tracking and source seeking algorithms in the literature has been given in this chapter. Particular attention has been dedicated to algorithms aimed at the exploration of environmental features with autonomous vehicles. The reviewed methodologies have been classified through different criteria, such as the dependence on the spatial gradient, the deployment of a single vehicle or a formation of vehicles and the applicability to the exploration of static or dynamic spatial features. The algorithms based on the deployment of a single vehicle are advantageous because of the reduced operational and communication costs. These, however, may require longer operational times and complex vehicle manoeuvres to fully explore the features of interest. When a high resolution sampling of the considered feature is required, the deployment of a formation of autonomous vehicles may result preferable.

The general trend in developing boundary tracking and source seeking algorithms aims at reducing the measurements required in the definition of the control actions for the autonomous vehicles. Additionally, the desire to embed the computational capability directly on-board of the autonomous vehicles encourages the development of control strategies based on simple computations.

In this thesis, both boundary tracking and source seeking approaches are proposed for the exploration of oceanic features. Most of the work in this thesis focuses on the single vehicle application, considering an ASV when dealing with two-dimensional features, and an AUV when dealing with three-dimensional features. In addition, a formation based boundary tracking approach is proposed in Chapter 6.

Chapter 3

Boundary tracking using a single autonomous vehicle

3.1 Introduction

In this chapter, an algorithm for tracking the boundary of a spatial phenomenon using a single autonomous vehicle is discussed. The methodology is applied to make an ASV track the boundary of oceanic features. A review of the literature on single vehicle boundary tracking methods was presented in Section 2.2.1. It has been highlighted how the deployment of a single vehicle reduces the operational and communication costs involved in performing boundary tracking. Consequently, the problem of tracking the boundary of oceanic features with a single autonomous vehicle is considered in this chapter. The boundary tracking algorithm discussed in this chapter has been selected amongst those described in Chapter 2 due to the reduced requirements in terms of sensing and computation.

The suboptimal sliding mode boundary tracking algorithm, originally proposed in [39], is considered in this chapter. The algorithm is based on the instantaneous measurement of the spatial field at the position of the vehicle and two previous measurements. Higher order information about the spatial field, such as the spatial gradient and the Hessian, is not required. The suboptimal sliding mode boundary tracking algorithm acts basically as a *steering control* for the vehicle.

In [39], the approach is applied to the control of an UAV flying at a constant height, with the aim of tracking the boundary of an ash cloud resulting from a volcanic eruption. A realistic dataset relating to the Eyjafyallajökull volcano eruption in 2010, obtained from the Met Office, is used for the validation of the algorithm in a simulated environment. However, the validation considers only a simple, static case of the problem.

In this chapter, the algorithm is used to track the boundary of two-dimensional oceanic features with a single ASV. The oceanic features considered in this chapter are a Sea-Surface Temperature (SST) front in a tidal-mixing area and a tracer release patch. Realistic datasets, obtained from the Met Office ocean models, are used in the simulations. Comparing to the basic sets of results in [39], the effect of external disturbances, typically in the form of drift, and the dynamic nature of the spatial field are addressed in this chapter.

The chapter is organized as follows: the considered problem is defined in Section 3.2; particularly, the considered two-dimensional spatial field is defined in Section 3.2.1, the kinematic model used for the ASV is introduced in Section 3.2.2, while the control design objective is formulated in Section 3.2.3. The suboptimal sliding mode boundary tracking approach is presented for static fields in Section 3.3. The proposed approach is validated in simulations: firstly, a numerical example is considered in Section 3.4. Secondly, the results obtained considering a tidal mixing temperature front are presented in Section 3.5, while those obtained considering a tracer release scenario are shown in Section 3.6. The possibility to track the boundary of a dynamic two-dimensional field is also introduced. Concluding remarks are given in Section 3.7, while an introduction to sliding mode control is given in Appendix A.

3.2 Problem definition

3.2.1 Two-dimensional spatial field

Consider an unknown two-dimensional time-invariant spatial phenomenon described as:

$$\gamma(x, y) : \mathcal{D} \to \mathbb{R} \tag{3.2.1}$$

where $\mathcal{D} \subset \mathbb{R}^2$ is a compact two-dimensional region containing the spatial feature, as shown in Fig. 3.2.1. The spatial mapping $\gamma(x, y)$ associates a numerical value of the spatial phenomenon with every position $(x, y) \in \mathcal{D}$, for instance with every latitude and longitude stamp. The spatial field is assumed to be continuous and smooth; this requires the expression of the spatial field $\gamma(x, y)$ and its derivatives, hence the spatial gradient $\nabla\gamma$, to be continuous everywhere in $\mathcal{D}$. In addition, in order to allow the solution of the boundary tracking problem, it is assumed that the spatial gradient $\nabla\gamma$ and the time derivative of the spatial gradient $\frac{d||\nabla\gamma||}{dt}$ are bounded. The knowledge of the bounds is not required; in fact, neither the explicit expression of the spatial mapping $\gamma(x, y)$ nor the gradient information are available, or estimated. As a final assumption, the diffusion and advection rates of the spatial mapping are assumed to be sufficiently slow such that the scenario can be considered as static. In other words, it is assumed that the vehicle can move fast enough to be able to complete the boundary tracking task before any significant change to the spatial map occurs. A remark on the applicability of the presented approach to dynamic fields is given in Section 3.6.2.

A graphical representation of a static two-dimensional spatial field is given in Fig. 3.2.1. The objective of the boundary tracking algorithm is to make the vehicle track the boundary of the two-dimensional feature autonomously. The boundary of the spatial phenomenon in (3.2.1) is assumed to be a smooth simple contour, defined as the compact level set:

$$\Gamma := \{(x, y) \in \mathcal{D} \mid \gamma(x, y) = \gamma^*\}, \tag{3.2.2}$$

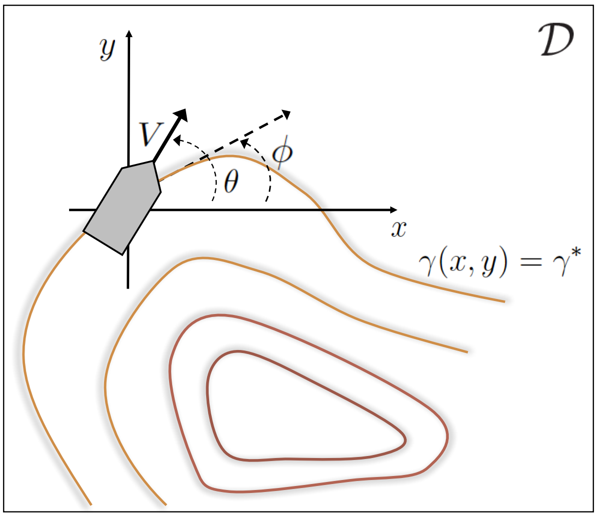


Figure 3.2.1: Spatial field and tracked boundary characterization

where γ^* is the tracked value of the spatial phenomenon; it is assumed here that $\gamma^* > 0$. The tracked value may be seen as a threshold, corresponding for instance to a safe contamination level. Alternatively, the tracked value may be chosen according to particular scientific interest, as for instance a certain value of water temperature in a tidal-mixing front area, or a value of water depth associated with specific bathymetry features. The tracked contour is assumed to be smooth; this justifies the introduced hypothesis on the spatial field gradient, which is assumed continuous, with continuous spatial derivatives.

3.2.2 Vehicle model

The kinematic of the vehicle is modelled as a nonholonomic unicycle [51]:

$$\begin{cases} \dot{x}(t) = V\cos\theta(t) \\ \dot{y}(t) = V\sin\theta(t) \\ \dot{\theta}(t) = u(t) \end{cases} \tag{3.2.3}$$

where $x(t)$ and $y(t)$ are the x-axis and y-axis positions of the vehicle respectively (e.g. longitude and latitude), and $\theta(t)$ is the heading angle, at an instant of time t. The heading angle $\theta(t)$ is conventionally measured starting from the horizontal axis in anticlockwise direction [79], as shown in Fig. 3.2.1. The speed of the vehicle V is assumed to be constant, while the angular velocity $\dot{\theta}(t)$ can be directly manipulated through the control input $u(t)$. Hence, the control input is a steering control, and it only affects the direction of movement of the vehicle.

The nonholonomic constraint associated with the model in (3.2.3) is [80]:

$$\dot{y}(t)\cos\theta(t) - \dot{x}(t)\sin\theta(t) = 0 \tag{3.2.4}$$

This constraint limits the sideways movement of the vehicle, as it imposes a zero lateral velocity to the vehicle: the vehicle can only move in the direction of the speed V. In addition, the assumption that the speed of the vehicle is constant prevents the vehicle from stopping at a desired goal position.

Though the dynamics of the autonomous vehicle is not represented here, the kinematic unicycle model is an effective and simple representation of its movement and can be used at the guidance level [61]. This model has been previously used to represent ASVs in [81], where a path following guidance strategy is developed on the basis of the kinematic model in (3.2.3) and experimentally validated using a small catamaran-like vehicle. Furthermore, the representation in (3.2.3) has been used in [39], for an unmanned aerial vehicle moving at a specified height, and in [82], where UGVs are modelled through (3.2.3) in order to design a formation navigation strategy.

When using the kinematic model in (3.2.3), it is assumed (Assumption 1) that there exists a low level inner loop control scheme for the vehicle, which addresses the motion constraints emanating from the vehicle's dynamics, which have not been precisely accounted for. Additionally, it is assumed that the vehicle is equipped with the suitable sensors for measuring $\gamma(x(t), y(t))$, the value of the spatial field at its current position $(x(t), y(t))$ at time t. This measurement, in fact, is required for the boundary tracking algorithm presented in this chapter.

3.2.3 Sliding mode control objective

The objective is to design a suitable steering control law $\dot{\theta}(t) = u(t)$, so that the vehicle modelled as in (3.2.3) can identify and track the boundary set Γ, defined in (3.2.2). This control objective is posed from a Sliding Mode Control (SMC) perspective.

SMC, a class of Variable Structure Control (VSC), is a popular nonlinear robust control methodology [83], [84]. The underlying philosophy of sliding mode control is to first define a manifold, the so-called *sliding surface* [83], and then to drive the states of the dynamical system onto that manifold, using an external forcing term. The sliding surface definition introduces a set of constraints on the system's states. These are typically defined as a function of a *sliding variable.* If these constraints are verified by the system's states, then the desired behaviour of the system is achieved, and a *sliding motion* is obtained. During sliding, the system shows robustness with respect to the so-called *matched uncertainties*, which are uncertainties entering the input channel. The behaviour of the closed loop system controlled through SMC techniques can be divided into two phases. The phase in which the system states are driven towards the sliding surface is referred to as the *reaching phase.* Since the states of the system reach the sliding surface, they are constrained to remain on it for all the subsequent time, obtaining the *sliding phase.* A more detailed introduction to SMC is given in Appendix A.

The definition of the sliding surface needs to mirror the introduced control objective. For the boundary tracking control objective, the sliding surface is defined as the locus:

$$S := \{(x(t), y(t)) \in \mathcal{D} : \gamma(x(t), y(t)) - \gamma^* = 0\} \qquad (3.2.5)$$

where the spatial field measurement at the vehicle's position $\gamma(x(t), y(t))$ coincides with the value on the tracked boundary γ^*. Defining the sliding variable as the difference between the measurement at the vehicle's position and the tracked value:

$$\sigma(t) := \gamma(x(t), y(t)) - \gamma^* \ , \tag{3.2.6}$$

the sliding surface S results the locus where $\sigma(t) = 0$. If a sliding motion is obtained, the vehicle's measurements coincide with the value of the spatial field on the tracked contour; consequently, boundary tracking is successfully achieved.

It is assumed in [39], while developing the boundary tracking scheme, that the vehicle is initially deployed *sufficiently close* to the tracked contour, that is in a vicinity of the tracked contour. Despite a vicinity of the tracked contour always exists, the correct deployment of the vehicle may be difficult in the case of an imprecisely known spatial feature. Consequently, this assumption will be relaxed in Chapter 5.

3.3 Suboptimal sliding mode boundary tracking

3.3.1 Relative degree and sliding order

In order to choose a suitable SMC design for the steering control $u(t)$, the relative degree of the system between the sliding variable $\sigma(t)$ and the control action $u(t)$ has been determined.

It can be shown that the system between the sliding variable $\sigma(t)$ defined in (3.2.6), treated here as the output, and the control input $u(t)$ has a constant relative degree two dynamics (for a definition of relative degree, see for instance [85]). This implies that two subsequent differentiations of (3.2.6) are necessary in order to obtain an explicit term dependent on the control action.

Differentiating the expression of $\sigma(t)$ in (3.2.6) along the system's trajectory in

(3.2.3), yields:

$$\begin{aligned}
\dot{\sigma}(t) &= \frac{\partial\gamma}{\partial x}(V\cos\theta) + \frac{\partial\gamma}{\partial y}(V\sin\theta) && (3.3.1)\\
&= V||\nabla\gamma||\left[\cos\theta\cos\left(\phi + \frac{\pi}{2}\right) + \sin\theta\sin\left(\phi + \frac{\pi}{2}\right)\right] && (3.3.2)\\
&= V||\nabla\gamma||\left[-\cos\theta\sin(\phi) + \sin\theta\cos(\phi)\right] && (3.3.3)\\
&= ||\nabla\gamma||V\sin(\theta - \phi) && (3.3.4)
\end{aligned}$$

where $||\nabla\gamma||$ is the magnitude of the field spatial gradient, defined as

$$\nabla\gamma = \left[\frac{\partial\gamma}{\partial x}, \frac{\partial\gamma}{\partial y}\right], \qquad (3.3.5)$$

and ϕ is the angle between the tangent line to the tracked contour and the horizontal direction, as shown in Fig. 3.2.1. The projection of the spatial field gradient components onto the x and y axis yields $\frac{\partial\gamma}{\partial x} = -||\nabla\gamma||\sin(\phi)$ and $\frac{\partial\gamma}{\partial y} = ||\nabla\gamma||\cos(\phi)$.

As the expression in (3.3.4) is independent of the control action, a second differentiation step is performed, obtaining:

$$\ddot{\sigma}(t) = \underbrace{V\sin(\theta - \phi)\frac{d||\nabla\gamma||}{dt}}_{\xi(t)} + \underbrace{||\nabla\gamma||V\cos(\theta - \phi)}_{b(t)}(\dot{\theta} - \dot{\phi}) \qquad (3.3.6)$$

Equation (3.3.6) involves a term depending on the vehicle steering law $\dot{\theta}(t) = u(t)$. Consequently, the system between the sliding variable and the control action has relative degree two.

The expression in (3.3.6) can be rewritten in the following generic form, as in [84]:

$$\ddot{\sigma}(t) = \xi(t) + b(t)(u(t) - \dot{\phi}(t)) \qquad (3.3.7)$$

where the uncertain terms $\xi(t)$ and $b(t)$ are introduced. The imprecisely known drift term $\xi(t)$ and the time varying multiplicative uncertainty $b(t)$ depend on the spatial gradient information $\nabla\gamma$, which is unknown. These uncertainty terms are assumed to be bounded. As the expression of $\ddot{\sigma}(t)$ in (3.3.6) has an identical structure to the system considered in [84], the following classical assumptions on the uncertain terms are introduced:

$$|\xi(t)| < C, \quad 0 < K_m \leq b(t) \leq K_M \tag{3.3.8}$$

where C, K_m, K_M are positive constants. Additionally, the term $\dot{\phi}(t)$ is treated as a *matched uncertainty*. The term *matched* is associated with an uncertainty which affects the input channel.

The relative degree of the system to be controlled influences the choice of the SMC structure [84]. As $u(t)$ only affects $\ddot{\sigma}(t)$ in the case of a relative degree two system, the boundary tracking control objective can be achieved only imposing a second order sliding motion, characterized as:

$$\sigma(t) = \dot{\sigma}(t) = 0 \tag{3.3.9}$$

Consequently, for a relative degree two system, a Second Order Sliding Mode (2-SM) controller should be chosen, which imposes a second order sliding motion in finite time. A review of the main 2-SM controllers is also included in Appendix A.

3.3.2 Suboptimal sliding mode control

Standard r-th order sliding mode controllers as in [84] require the knowledge of $\sigma(t), \dot{\sigma}(t), \ldots, \sigma^{(r-1)}(t)$, where $\sigma^{(r-1)}(t)$ is the derivative of order $(r-1)$ of $\sigma(t)$. Therefore, a standard second order sliding mode controller as in [86] requires knowledge of $\sigma(t)$ and $\dot{\sigma}(t)$. This is also true for the quasi-continuous and the twisting controllers [84]. In the current application, however, $\dot{\sigma}(t)$ is unknown because the spatial field gradient information, which appears in (3.3.4), is not available. Consequently, these approaches result inapplicable.

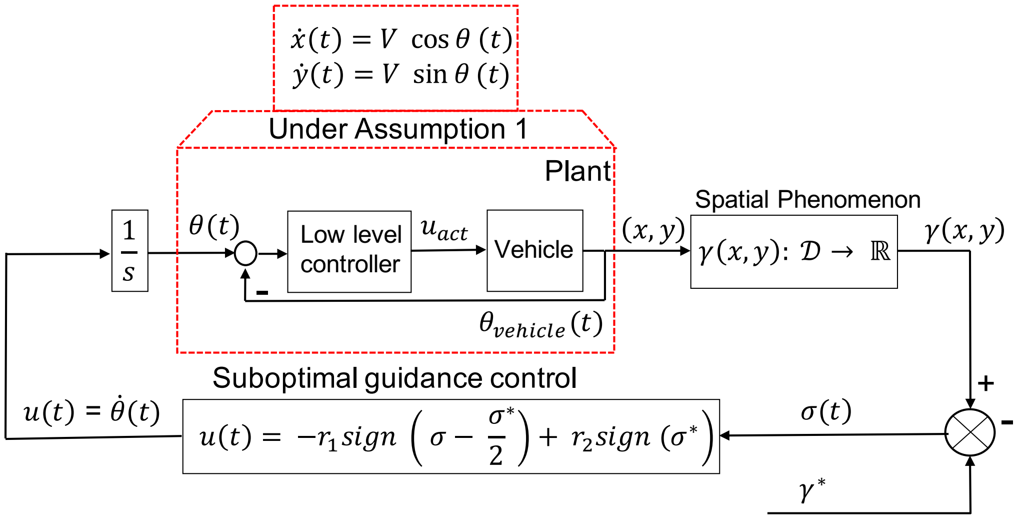

$\dot{x}(t) = V \cos\theta(t)$
$\dot{y}(t) = V \sin\theta(t)$
Under Assumption 1
Plant
$\frac{1}{s}$
$\theta(t)$
Low level controller
u_{act}
Vehicle
$\theta_{vehicle}(t)$
(x, y)
Spatial Phenomenon
$\gamma(x,y)\colon \mathcal{D} \to \mathbb{R}$
$\gamma(x,y)$
+
-
Suboptimal guidance control
$u(t) = \dot{\theta}(t)$
$u(t) = -r_1 sign\left(\sigma - \frac{\sigma^*}{2}\right) + r_2 sign\left(\sigma^*\right)$
$\sigma(t)$
γ^*

Figure 3.3.1: Schematic of the proposed boundary tracking approach

The *suboptimal sliding mode* controller, originally proposed in [53], imposes a second order sliding motion without requiring the knowledge nor the estimation of $\dot{\sigma}(t)$. This control design only requires the knowledge of the sliding variable $\sigma(t)$ at different time instances [53], [84], [87]. The suboptimal sliding mode controller is therefore chosen because of its low information requirement. The steering control results:

$$u(t) = -r_1 \operatorname{sign}\left(\sigma - \frac{\sigma^*}{2}\right) + r_2 \operatorname{sign}(\sigma^*), \quad r_1 > r_2 > 0 \tag{3.3.10}$$

where the controller gains r_1 and r_2 are positive design constants, and σ^* is the value of the sliding variable when the condition $\dot{\sigma}(t) = 0$ was last verified. A schematic of the boundary tracking control approach is shown in Fig. 3.3.1.

Being the first derivative of the sliding variable unknown, the occurrence of a zero-crossing for $\dot{\sigma}(t)$ is determined through a digital peak detector, originally proposed in [88]. The digital peak detector is based on the knowledge of the sliding variable at different time instants, specifically $\sigma(t)$, $\sigma(t-1)$ and $\sigma(t-2)$, and it determines the occurrence of a zero-crossing for $\dot{\sigma}(t)$ by monitoring the sign of:

$$(\sigma(t-2) - \sigma(t-1))\,(\sigma(t-1) - \sigma(t)) \tag{3.3.11}$$

If (3.3.11) is negative, then zero-crossing is verified and $\sigma^* = \sigma(t-1)$ is set. If σ^* is estimated through the digital peak detector, the control law in (3.3.10) does not require the knowledge or the estimation of the gradient of the spatial field at the vehicle's position. The algorithm can easily be implemented using the current measurement $\sigma(t)$ and the last two point measurements at the location of the vehicle $(\sigma(t-1), \sigma(t-2))$.

A set of design constraints on the gains r_1 and r_2 of the controller has been introduced in [84]. Specifically:

$$r_1 - r_2 > \frac{C}{K_m} \tag{3.3.12}$$

$$r_1 + r_2 > \frac{4C + K_M(r_1 - r_2)}{3K_m} \tag{3.3.13}$$

A set of values for r_1 and r_2 satisfying these constraints always exists. Equation (A.2.11), in fact, imposes a positive lower limit on the value of $(r_1 - r_2)$, which is positive by definition as it is required that $r_1 > r_2 > 0$. Consequently, the right-hand side of (3.3.13) is always positive and values of r_1 and r_2 satisfying this constraint always exist.

To impose the turning radius of the vehicle, an additional condition is introduced. Considering the movement of the vehicle, modelled as in (3.2.3), along a circumference of radius R, the relation between the vehicle's speed V and the angular velocity $\dot{\theta}(t)$, which is the steering control input, is:

$$V = \dot{\theta}(t)R = u(t)R \tag{3.3.14}$$

Consequently, the minimum radius of curvature of the vehicle's trajectory is:

$$R_{min} = \frac{V}{u_{max}} \tag{3.3.15}$$

where u_{max} is the maximum value of the controller in (3.3.10), i.e. $u_{max} = |r_1| + |r_2|$. If the tracked contour is characterized by turns with a certain minimum radius of

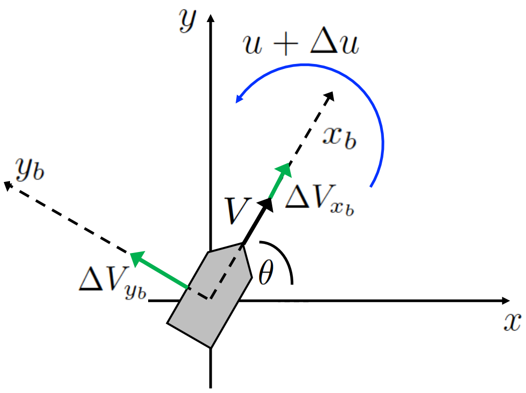


Figure 3.3.2: Disturbances affecting the vehicle's movement

curvature R, the controller gains should be chosen accordingly, in order to ensure that $R_{min} < R$. If this condition is not verified, the vehicle can not turn fast enough to track the sharp features of the tracked contour and therefore tracking is temporarily worsened.

Considering the additional turning constraint in (3.3.15), the controller's gains need to be chosen from the set:

$$\Big\{ r_1 > r_2 > 0 \in \mathbb{R}^+ : (r_1 - r_2) \; > \; \frac{C}{K_m}, \qquad (3.3.16)$$
$$(r_1 + r_2) \; > \; \max\left\{\frac{4C + K_M(r_1 - r_2)}{3K_m}, \frac{V}{R_{min}}\right\}\Big\}$$

When the autonomous vehicle is on the tracked contour, it is desirable to know a priori in which direction it will track it (clockwise or anticlockwise). The direction of movement along the tracked contour can be influenced through the definition of the sliding variable in (3.2.6). With the definition given in (3.2.6), in particular, the movement happens in *anticlockwise* direction. Changing the definition of $\sigma(t)$ to $\sigma(t) = \gamma^* - \gamma(x(t), y(t))$ affects the sign of the control law and therefore the steering direction, resulting in the movement along the tracked contour in an opposite direction.

3.3.3 Robustness to external disturbances

In the presence of disturbances, which can affect the speed and the angular velocity of the vehicle, the kinematic movement of the vehicle in (3.2.3), can be written as

[89]:

$$\begin{cases} \dot{x}(t) = (V + \Delta V_{x_b})\cos\theta(t) - \Delta V_{y_b}\sin\theta(t) \\ \dot{y}(t) = (V + \Delta V_{x_b})\sin\theta(t) + \Delta V_{y_b}\cos\theta(t) \\ \dot{\theta}(t) = u(t) + \Delta u \end{cases} \tag{3.3.17}$$

The drift terms in the x_b and y_b directions in the body reference frame are modelled here as $\Delta V_{x_b}(t)$ and $\Delta V_{y_b}(t)$ and the disturbance on the angular velocity term is modelled as Δu. The disturbances are shown in Fig. 3.3.2. In the presence of the disturbance terms, (3.3.6) can be rewritten as:

$$\begin{aligned} \ddot{\sigma}(t) = \underbrace{\left[(V + \Delta V_{x_b})\sin(\theta - \phi) + \Delta V_{y_b}\cos(\theta - \phi)\right]\frac{d||\nabla\gamma||}{dt} + ||\nabla\gamma||\left(\frac{d\Delta V_{x_b}}{dt}\sin(\theta - \phi) + \frac{d\Delta V_{y_b}}{dt}\cos(\theta - \phi)\right)}_{\xi(t)} \\ + \underbrace{||\nabla\gamma||\left[(V + \Delta V_{x_b})\cos(\theta - \phi) - (\Delta V_{y_b})\sin(\theta - \phi)\right]}_{b(t)}\left(u + \Delta u - \dot{\phi}\right) \end{aligned}$$

The suboptimal sliding mode control law is robust with respect to matched uncertainties [84], and thus to Δu and the drift terms affecting $b(t)$. The drift terms can be assumed bounded as $|\Delta V_{x_b}|, |\Delta V_{y_b}| < \overline{\Delta V} < V$, where the upper bound is introduced in order to guarantee the capability of the vehicle to move forward. Additionally, a bound $|\frac{d\Delta V_{x_b}}{dt}|, |\frac{d\Delta V_{y_b}}{dt}| \leq \widetilde{\Delta V}$ can be introduced. Under these assumptions, the bound on $\xi(t)$ in (3.3.8) can be rewritten as

$$|\xi(t)| < C + 2\overline{\Delta V}\frac{d||\nabla\gamma||}{dt} + 2\widetilde{\Delta v}||\nabla\gamma|| \tag{3.3.18}$$

A design satisfying this increased bound on the uncertainty term $\xi(t)$ can perform robustly in the presence of both matched and bounded unmatched uncertainties.

It is evident that the implication of the modified bound in (3.3.18) is a larger value for $(r_1 - r_2)$ and $(r_1 + r_2)$, and hence of the controller gains. Typically, an upper bound for $|r_1| + |r_2|$ emanates from the practical minimum turning radius of the vehicle, and this has to be accounted for during the design process. In practice, the

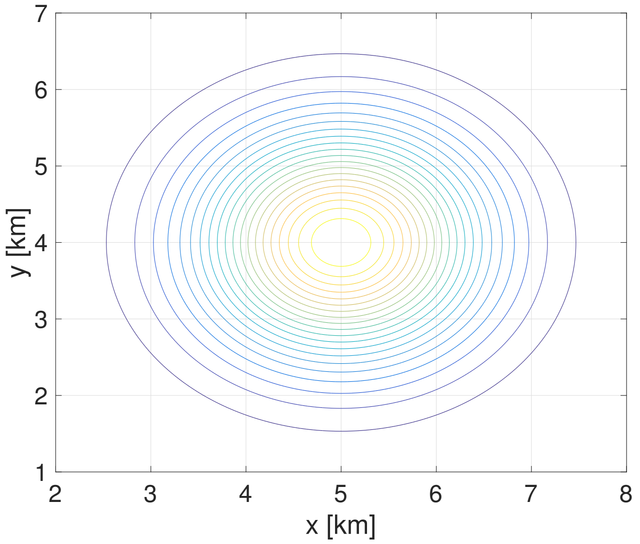


Figure 3.3.3: Numerical boundary tracking results - spatial function

disturbance components correspond to environmental forces acting on the movement of the vehicle, such as winds or oceanic currents. The forecast relating to these quantities can be exploited in order to define a reasonably conservative value of the upper bounds.

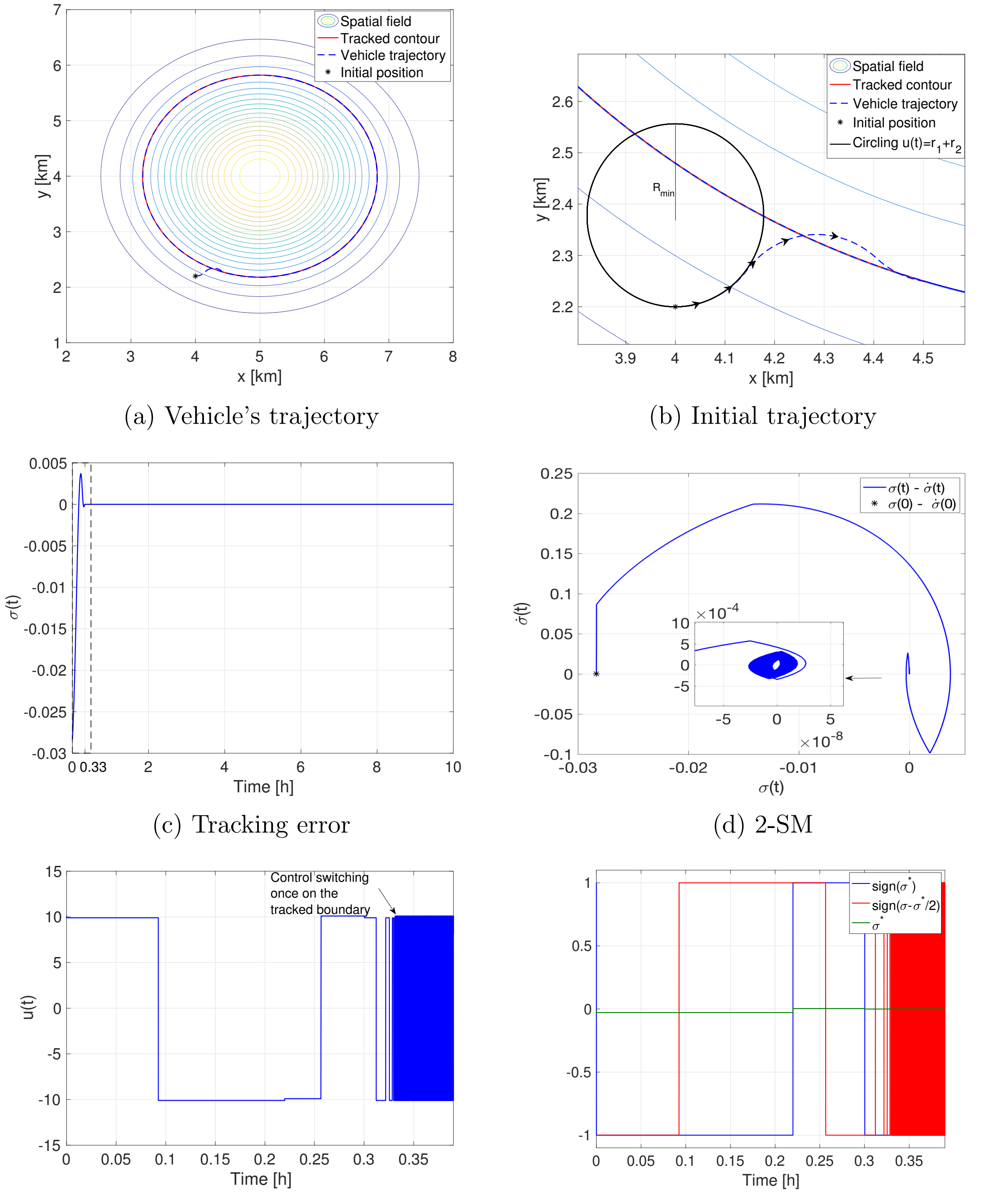
3.4 Numerical example

In this section, a numerical example is considered, in order to demonstrate the effectiveness of the boundary tracking approach in simulation. Consider the Gaussian distribution:

$$f(x,y) = \frac{1}{\sqrt{2\pi\sigma_G^2}} e^{-\frac{(x-\mu_x)^2+(y-\mu_y)^2}{2\sigma_G^2}}, \tag{3.4.1}$$

centred at (μ_x, μ_y) with variance σ_G. The spatial field shown in Fig. 3.3.3 has been obtained through the function in (3.4.1), with $\sigma_G = 1$, $\mu_x = 5$ and $\mu_y = 4$.

The simulations have been run in a Matlab/Simulink environment (version R2016b), exploiting the fixed step Euler integration method, with a step size of 0.1 s. A vehicle with velocity 0.5 m/s $\approx$ 1.8 km/h is modelled through (3.2.3) and its angular velocity is controlled through the suboptimal sliding mode approach in (3.3.10), with the design parameters set to $r_1 = 10$ and $r_2 = 0.1$. With these choices, uncertainties



(a) Vehicle's trajectory

(b) Initial trajectory

(c) Tracking error

(d) 2-SM

(e) Control action before sliding occurrence

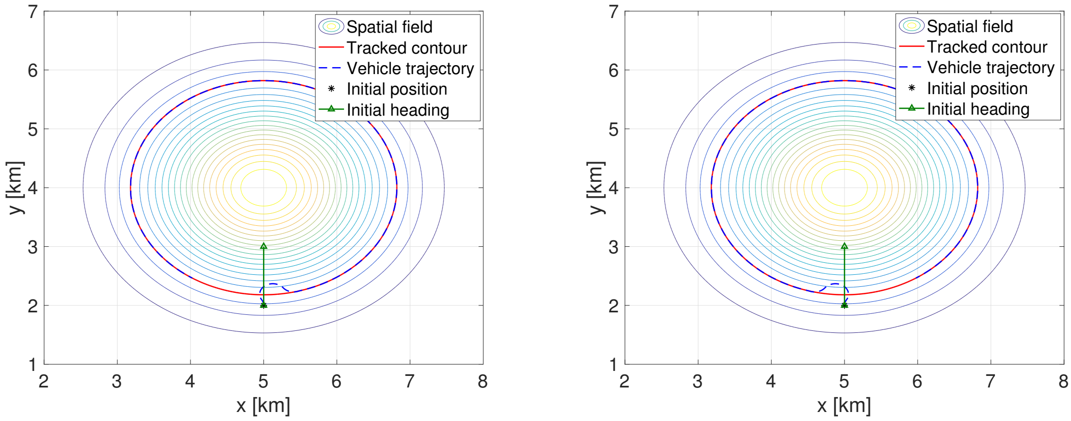
(f) Behaviour of $sign(\sigma^*)$, $sign(\sigma - \frac{\sigma^*}{2})$, σ^*

Figure 3.4.1: Numerical boundary tracking results

bounded as in (3.3.8) with $K_m = 15$, $K_M = 50$, $C = 10$ can be tackled by the proposed approach and the minimum turning radius of the autonomous vehicle is limited to approximately $R_{min} = 0.178$ km.

The results obtained in simulation are shown in Fig. 3.4.1. The vehicle is initially deployed at position $(x(0), y(0)) = (4, 2.2)$, with $\theta(0) = 0$, in the vicinity of the tracked contour, characterized by $\gamma^* = 0.075$. The initial position of the vehicle is shown as a black star in Fig. 3.4.1a, where the tracked contour is highlighted in red and the vehicle's trajectory is shown in blue. From its initial deployment, the vehicle moves with angular velocity $\dot{\theta} = r_1 + r_2$ along a circumference of radius R_{min}. After approximately 0.33 h, the vehicle successfully intercepts the tracked contour. The tracking error is shown in Fig. 3.4.1c, and highlights the effective tracking of the desired contour. As soon as the tracking error is reduced to zero, in approximately 0.33 h, a sliding motion is obtained. As highlighted in Section 3.3.2, the chosen controller in (3.3.10) enforces a second order sliding mode, characterized by $\sigma(t) = \dot{\sigma}(t) = 0$ in finite time (0.33 h). A typical suboptimal second order sliding mode is shown in the phase plane trajectory of $\sigma(t)$ and $\dot{\sigma}(t)$, shown in Fig. 3.4.1d. This behaviour of the suboptimal sliding mode controller is highlighted in Appendix A. The evolution of the suboptimal control action $u(t)$ up to 0.37 h is shown in Fig. 3.4.1e. As soon as sliding is obtained, after approximately 0.33h, the control action is characterized by fast switching, which is necessary to maintain sliding. In practice, a filtered version of the control action is used. Fig. 3.4.1f shows the behaviour of the components of the suboptimal control action in (3.3.10) and the time evolution of σ^*. The switch in the sign of $(\sigma - \frac{\sigma^*}{2})$, which happens after approximately 0.09 hours, allows the vehicle to invert the sign of its angular velocity and to start moving along the tracked contour.

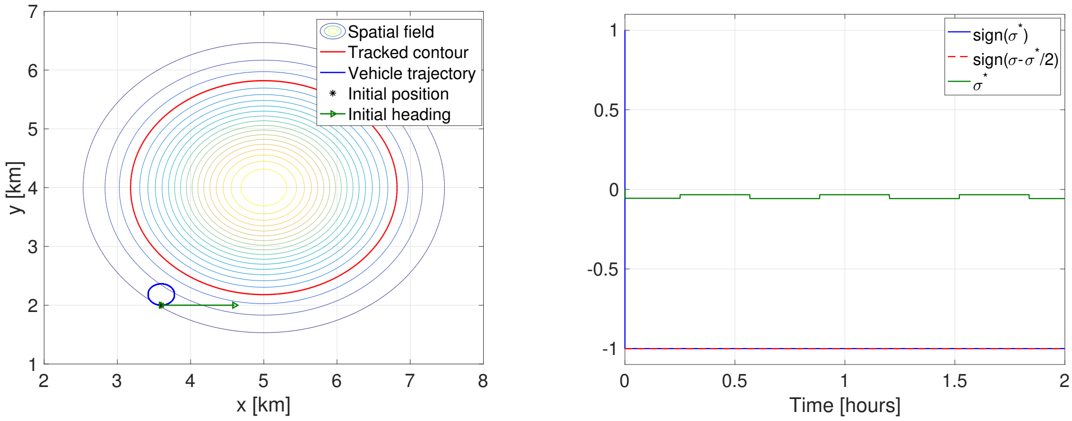
To demonstrate how the sliding variable definition affects the direction in which the vehicle traverses the boundary while tracking it, additional simulations have been carried out, and the obtained results are shown in Fig. 3.4.2. In this case, the vehicle is initially deployed at $(x(0), y(0)) = (5, 2)$, with $\theta(0) = \pi/2$. In two different simulations, the sliding variable is defined according to the strategy discussed in Section 3.3.2. The results in Fig. 3.4.2a, where the tracked contour is followed in anticlockwise direction, are obtained defining the sliding variable as in (3.2.6). The results in Fig. 3.4.2b, in which the boundary is followed by the vehicle in clockwise direction, are obtained defining the sliding variable as $\sigma(t) = \gamma^* - \gamma(x(t), y(t))$. In



(a) Anticlockwise boundary tracking $\sigma(t) = \gamma(x(t), y(t)) - \gamma^*$

(b) Clockwise boundary tracking $\sigma(t) = \gamma^* - \gamma(x(t), y(t))$

Figure 3.4.2: Numerical boundary tracking results - boundary tracking in clockwise and anticlockwise direction



(a) Vehicle's trajectory

(b) Behaviour of $sign(\sigma^*)$, $sign(\sigma - \frac{\sigma^*}{2})$, σ^*

Figure 3.4.3: Numerical boundary tracking results - initial position outside the vicinity of the tracked contour

conclusion, the strategy discussed in Section 3.3.2 is shown to be an effective way to a-priori determine the direction in which the vehicle will move along the tracked contour.

An additional simulation is run to demonstrate the local nature of the boundary tracking algorithm. The vehicle is initially deployed outside the vicinity of the tracked contour, at $(x(0), y(0)) = (3.6, 2)$, $\theta(0) = 0$. Fig. 3.4.3 shows the obtained results. As it can be seen, the vehicle moves on a circumference for the entire simulation time, and it never crosses the tracked contour. The evolution of the components

of the control action in (3.3.10) is shown in Fig. 3.4.3b. It can be seen how the sign of $(\sigma - \frac{\sigma^*}{2})$ never changes and hence the vehicle moves along a circumference of radius $R = \frac{V}{r_1 - r_2}$ for the entire length of the simulation.
This result motivates the requirement for a boundary *seeking strategy*, that allows the vehicle to move towards the tracked contour, even when it is initially deployed outside the vicinity of the contour. A possible solution, aimed at relaxing the assumption on the vehicle initial position relative to the tracked contour, will be introduced in Chapter 5.

As a final comment on the local behaviour of the approach, a more complex numerical example with two local maxima, shown in Fig. 3.4.4, is considered. The spatial field is obtained considering the cumulative effect of two Gaussian distributions as in (3.4.1), characterized by $\sigma_{Gi} = 1$, $i = 1, 2$ and centred at $(\mu_{x,1}, \mu_{y,1}) = (3, 4)$ and $(\mu_{x,2}, \mu_{y,2}) = (7, 4)$ respectively. The tracked contour, with $\gamma^* = 0.115$, is highlighted in red in Fig. 3.4.4. As it can be seen, this is composed of two distinct closed contours characterized by $\gamma(x, y) = \gamma^*$. In this case, the vehicle will track the contour in which vicinity it will be initially deployed. If the vehicle's initial position is equidistant from the two contours, then the boundary tracking algorithm makes the vehicle move along the contour which is firstly crossed by the vehicle when moving from its initial deployment. In the two cases shown in Fig. 3.4.5, the initial deployment of the vehicle is chosen as $(x(0), y(0)) = (5, 4)$ and distinct initial orientations $\theta(0) = 0$ and $\theta(0) = \pi$ respectively are chosen. The different initial heading of the vehicle determines which contour is first crossed by the vehicle and therefore it determines which of the two contours is tracked.

3.5 Tidal mixing front application

The possibility of collecting measurements in tidal mixing areas through the deployment of autonomous vehicles has been introduced in [33], where some preliminary experimental results are shown. Traditional surveying techniques are based on lawn-

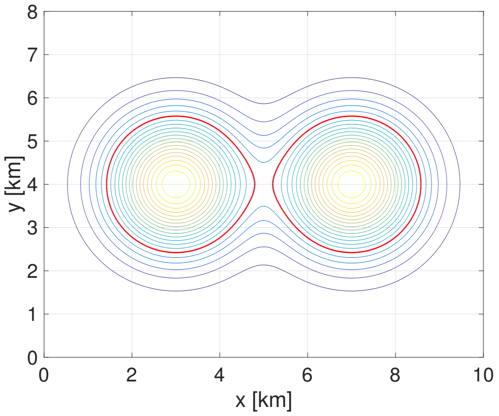
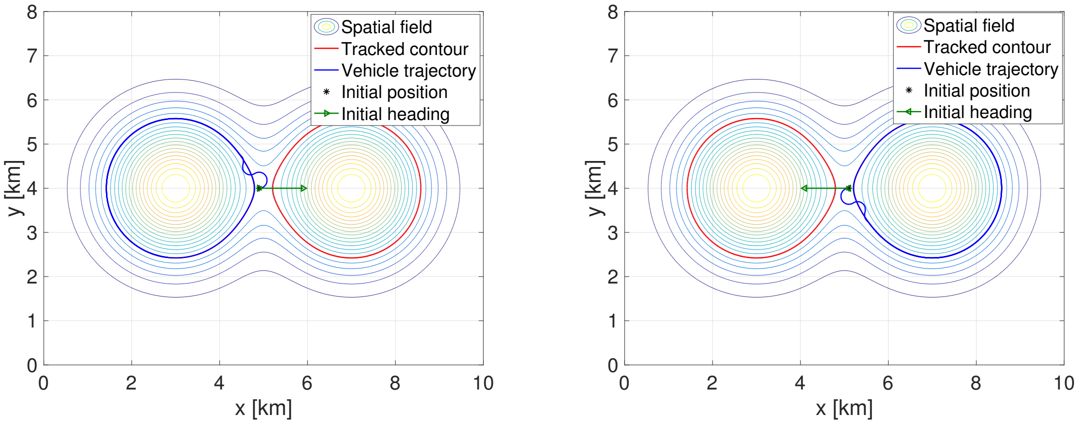


Figure 3.4.4: Numerical boundary tracking results - spatial function obtained from the cumulative effect of two Gaussian distributions



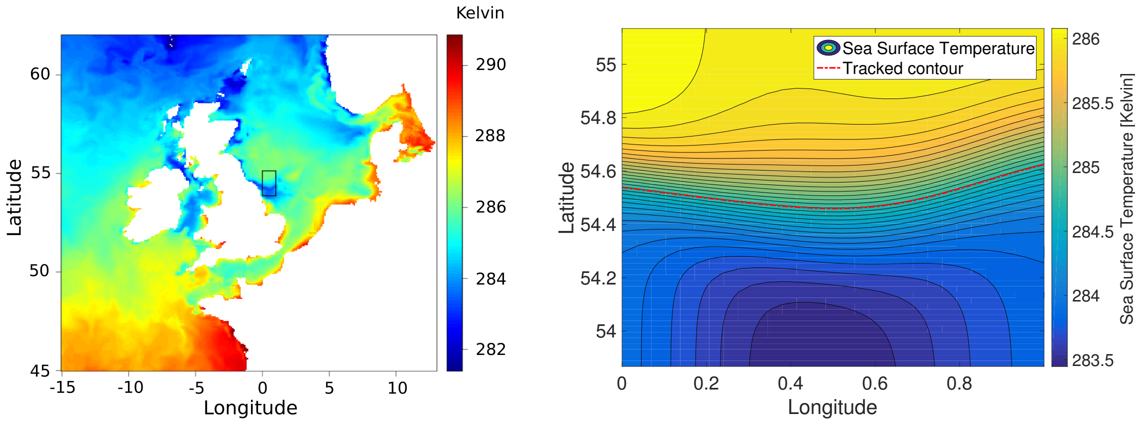
(a) Boundary tracking with $\theta(0) = 0$

(b) Boundary tracking with $\theta(0) = \pi$

Figure 3.4.5: Numerical boundary tracking results - local behaviour

mower or zig-zag type preplanned trajectories [33], to have a good coverage of the area. This is required in order to determine the front by post-processing the gathered measurements. Here, the collection of measurements in a tidal mixing front area is posed as a boundary tracking problem. The boundary in this case is defined as the locus where the water temperature has a constant value.

The problem is considered using a dataset, obtained from the Met Office, UK, and generated from the FOAM 7 km Atlantic Margin Model (AMM7) [90]. The available dataset is the sea-surface temperature in the European North West continental shelf, and it is shown in Fig. 3.5.1a. An area characterized by a tidal mixing front, for the application of the boundary tracking algorithm, is highlighted with a black



(a) Full dataset and area of interest

(b) Smoothed dataset in the area of interest

Figure 3.5.1: FOAM sea-surface temperature in the European North West continental shelf

box in Fig. 3.5.1a, and it extends 0° – 1° E, 53.9° – 55.1° N. Applying a flat approximation, the area corresponds to a rectangular region of approximately 65×140 km. The considered tidal mixing front is the Flamborough Head front, which has been identified in [91]. The dataset, with a resolution of approximately 2.5 km, is obtained through a smoothing procedure based on a probabilistic model, as in [92]. The dataset, with smooth variations of the sea surface temperature in the range $[283.5 - 286]$ Kelvin degrees, is shown in Fig. 3.5.1b.

3.5.1 Simulation results

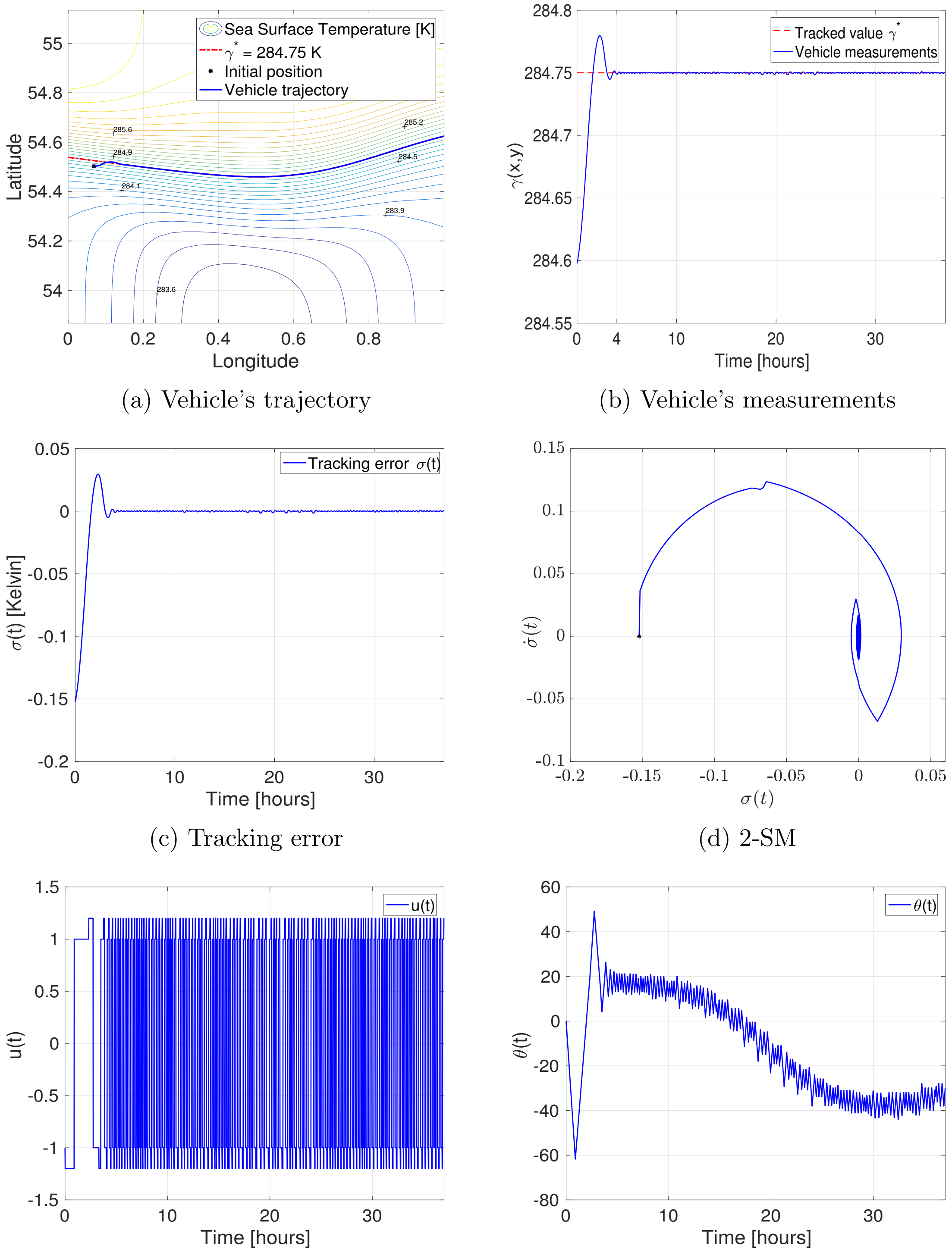
The kinematic model in (3.2.3) is used to model an ASV moving at a constant speed of 1.8 km/h ≈ 0.5 m/s. The vehicle is assumed capable of measuring the spatial field at its position through a designated sensor. The Euler integration method, with a fixed step size of 1 minute is used in the simulation. The larger update rate is aimed at mimicking a realistic application, in which the control action update rate is slow. The simulation length is set to 37 hours, in order to allow the vehicle to fully explore the tracked contour. The boundary tracking steering control is designed to allow the vehicle to track a constant SST contour in the central area of the temperature front, characterized by $\gamma^* = 284.75$ K and shown as a red dotted line in Fig. 3.5.1b. The vehicle is initially deployed in the vicinity of the tracked contour, at the initial longitude/latitude stamp $(x(0), y(0)) = (0.0688°, 54.5028°)$, with $\theta(0) = 0$.

The vehicle's initial position is shown in Fig. 3.5.2a as a black star. The suboptimal sliding mode steering control in (3.3.10) is implemented with $r_1 = 1.1$ and $r_2 = 0.1$. With these choices of the controller gains, specific bounds for the uncertain terms in (3.3.8) can be determined. The respective minimum turning radius of the vehicle in (3.3.15) is thereby limited to approximately 1.5 km. The minimum turning radius is therefore quite large, but this choice is motivated by two reasons: firstly, having a bigger turning radius facilitates the vehicle to detect the tracked contour from its initial deployment; secondly, the tracked contour does not show sharp features, and, consequently, the vehicle does not require high turning capabilities to track it.

The results obtained are shown in Fig. 3.5.2. Fig. 3.5.2a shows the SST contour plot, the tracked contour in red and the vehicle's trajectory in blue. The vehicle successfully detects the tracked contour and it moves along it, after achieving a sliding motion. Fig. 3.5.2b shows the tracked contour value γ^*, and the time evolution of the measurements collected by the vehicle. The measurement at the vehicle's initial position is $\gamma(x(0), y(0)) \approx 284.6$ K. The tracked contour is crossed after approximately 1.5 h and the vehicle then performs a series of oscillations around it, having decreasing amplitudes. After approximately 4 h, the vehicle starts sliding on the tracked contour, achieving a sliding motion with $\gamma(x(t), y(t)) \approx \gamma^*$ for the subsequent time. The initial sequence of oscillations is visible also in Fig. 3.5.2c, where the tracking error $\sigma(t)$ is shown. This typical feature of the suboptimal sliding mode controller is in this case accentuated by the slow control action update frequency and by the small values of the controller's gains, which limit the vehicle's turning capability. The achievement of a second order sliding mode in finite time is confirmed by the phase plane trajectory in Fig. 3.5.2d, whose initial point is highlighted with a black star. The obtained trajectory is typical of the suboptimal sliding mode controller - see Appendix A. In conclusion, with the proposed set up, accurate contour tracking is obtained, as the vehicle precisely follows the tracked contour. The tracking error, after sliding occurs, remains in the range ±0.005 K. This is guaranteed by the suboptimal sliding mode control in (3.3.10), whose behaviour is shown in Fig. 3.5.2e. Once the tracked contour is crossed, after approximately 1.5 hours, fast switching, aimed at maintaining the tracking error as small as possible, occurs. This causes the zig-zag behaviour of the vehicle's heading, shown in Fig. 3.5.2f, which is adjusted in order to follow the tracked contour.

Additional simulations are run to demonstrate the effects of the control action update frequency and the vehicle's speed on the tracking accuracy. In the first simulation, the control action update rate is further reduced from 1 minute to 5 minutes, while in the second simulation the vehicle's speed is increased from 0.5 m/s to 2 m/s. The results obtained are shown in Fig. 3.5.3. The trajectories of the vehicle in the two cases are shown in Fig. 3.5.3a; in both cases, the vehicle successfully detects the tracked contour in finite time and moves along it. Additionally, the tracking error relative to the two cases is shown in Fig. 3.5.3b, in comparison to the tracking error from the original simulation. The control action update frequency highly influences the obtained results in terms of tracking accuracy: with a 5 minutes update rate, in fact, the amplitude of the initial overshoot is increased and the overall tracking accuracy is significantly reduced. The tracking error oscillates around zero for the entire length of the simulation in a range of ±0.01 K. From a control perspective, having a slower control update frequency worsens the tracking performance of the algorithm, as the vehicle is unable to tightly track the boundary, but it oscillates around it. These oscillations, however, may be considered advantageous from a scientific perspective in some applications. In the tidal mixing front area studies, if the vehicle oscillates around the tracked contour, it is able to collect more distributed measurements across the front area. Consequently, a deeper sampling of the area, not focussed only on the tracked contour, is obtained. When choosing the control parameters, a trade-off between tracking accuracy and sampling requirements should be sought.

The effect of the vehicle's speed is different: it can be observed from Fig. 3.5.3a how the vehicle initially moves on a circumference having a greater radius; this is due to the definition of the minimum turning radius in (3.3.15), which introduces a



(a) Vehicle's trajectory

(b) Vehicle's measurements

(c) Tracking error

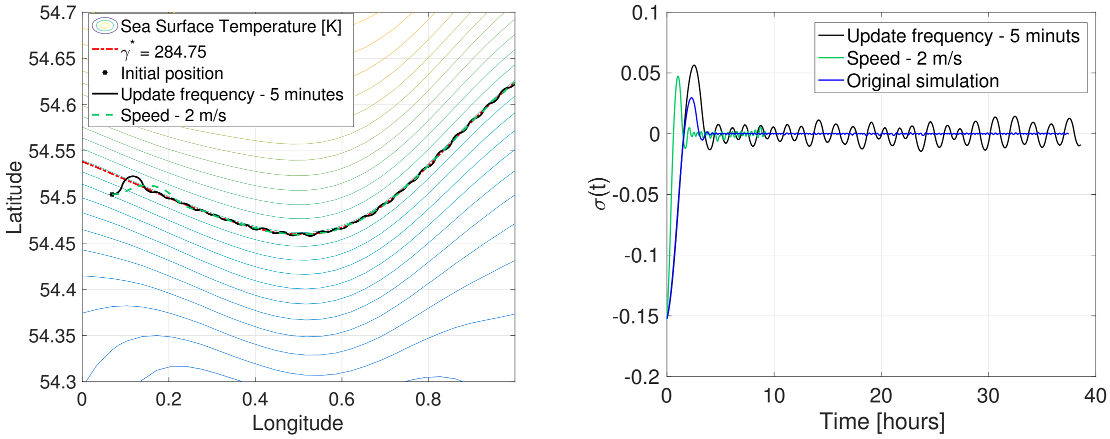
(d) 2-SM

(e) Steering control

(f) Vehicle's heading

Figure 3.5.2: Boundary tracking results - sea surface temperature front

direct proportionality between the vehicle's speed and the minimum turning radius. The tracking error remains of the same order of magnitude of the original simulation, but both the time taken to firstly cross the tracked contour and the time required for sliding to occur are reduced, as the tracked contour is firstly crossed



(a) Vehicle's trajectory

(b) Tracking error

Figure 3.5.3: Boundary tracking results - factors influencing tracking accuracy

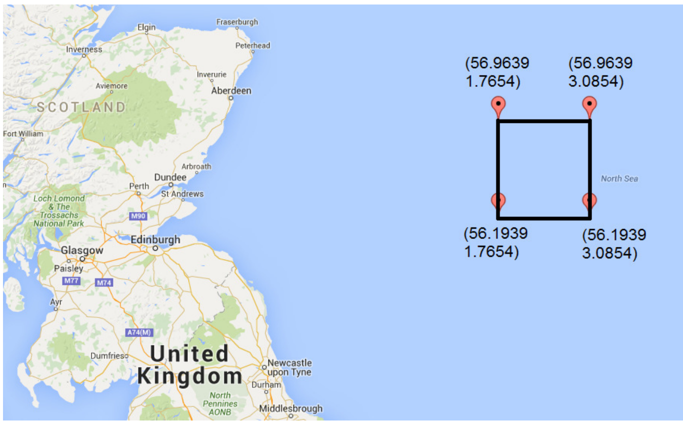


Figure 3.5.4: Boundary tracking results - tracer release considered region

after approximately 0.5 h and sliding occurs after approximately 2.5 h.

3.6 Tracer release application

In a tracer release experiment, the evolution of the shape and the position of the tracer patch in the ocean is monitored in order to study the oceanic currents affecting the movement of the tracer particles. A dataset obtained from the Met Office, UK, representing a tracer release over an approximately 18×16 km region in the Southern North Sea is considered, as shown in Fig. 3.5.4. The dataset is useful to mimic a polluting phenomenon in the ocean, such as an oil spill.

The dataset has been created at the Met Office mimicking the movement of a definite number of particles, released at a specific position, in the chosen region over time.

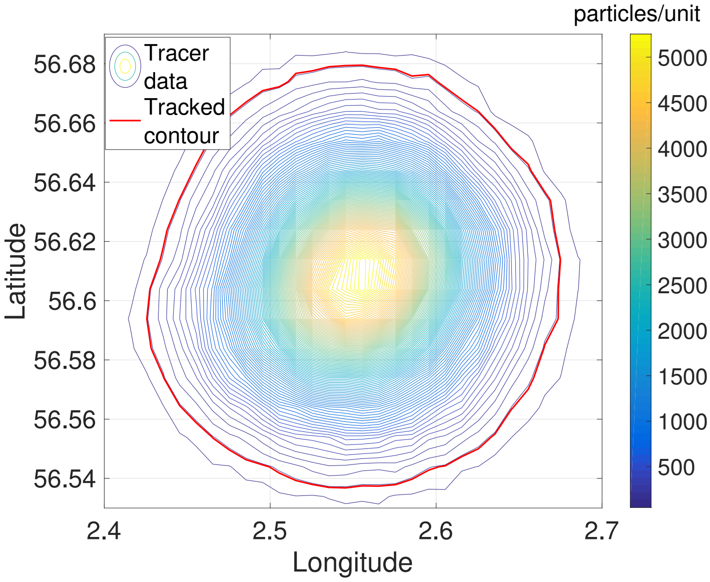


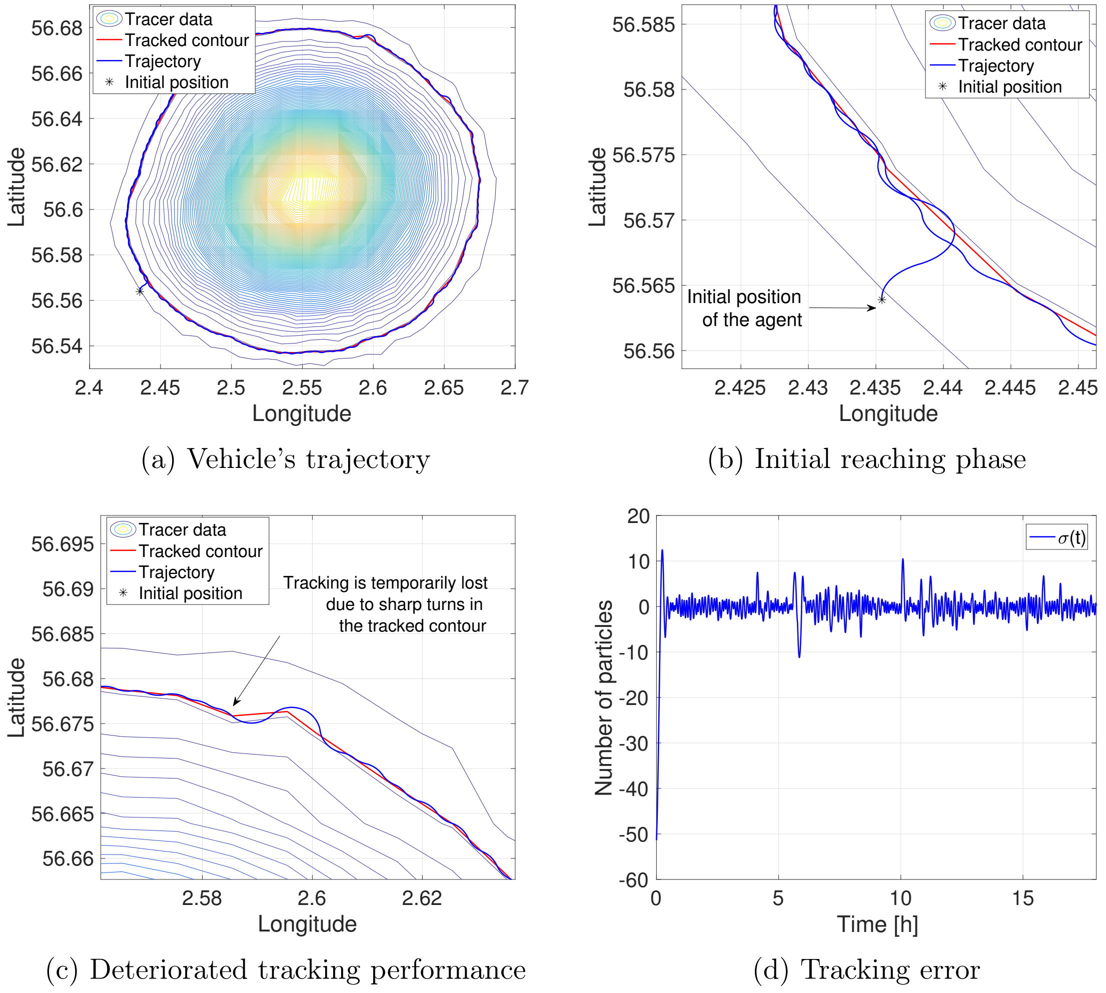
Figure 3.6.1: Boundary tracking results - static tracer release spatial field and tracked contour

The available data are the concentration of the tracer and the water currents in both the North and the East directions. The dataset has a spatial resolution of approximately 1 km and a time resolution of 5 minutes, and describes the evolution of the patch for a period of 15 days. The dataset can hence be used for applying the suboptimal sliding mode boundary tracking approach to a static spatial feature, obtained selecting a specific instant of time, and to a dynamic feature, obtained using the time-varying dataset.

3.6.1 Simulation results - static case

In the static scenario, the vehicle is assumed to move fast enough to complete the exploration of the boundary before the occurrence of any significant change. Furthermore, the effect of the sea-surface currents is neglected. The static scenario is based on the conformation of the tracer patch at the beginning of day 10, shown in Fig. 3.6.1. The tracked boundary, highlighted in red in Fig. 3.6.1, is characterized by a constant value of the tracer concentration of $\gamma^* = 100$ particles/unit.

The vehicle's speed is fixed to $V = 3.6$ km/h $= 1$ m/s and the steering control gains in (3.3.10) are set to $r_1 = 11$, $r_2 = 0.4$. With these choices, the minimum turning radius of the vehicle results $R_{min} \approx 0.3$ km, according to (3.3.15). The vehicle



(a) Vehicle's trajectory

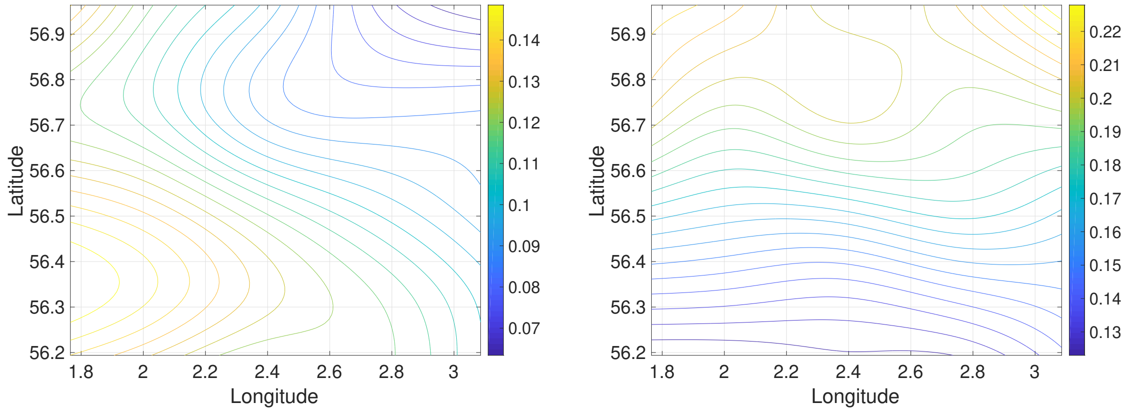
(b) Initial reaching phase

(c) Deteriorated tracking performance

(d) Tracking error

Figure 3.6.2: Boundary tracking results - static tracer release

is initially deployed sufficiently close to the tracked contour, with $(x(0), y(0)) = (2.4354, 56.5639)$ and $\theta(0) = \pi/2$. In order to allow the vehicle to fully explore the tracked boundary, an 18 hours long simulation is performed, with a control action update rate of 30 seconds. The obtained results are shown in Fig. 3.6.2. Fig. 3.6.2a shows the vehicle's trajectory in black and the vehicle's initial position as a black star. The vehicle successfully crosses the tracked contour in approximately 10 minutes, after which a sliding motion is obtained. The enlargement in Fig. 3.6.2b, relative to the initial instances of the simulation, shows how the trajectory of the vehicle is smoother than the tracked contour; this is due to the limiting effect of the controller gains r_1, r_2 on the vehicle's turning capabilities. Consequently, where the contour shows sharp features, sliding accuracy is reduced, as visible in more detail in Fig. 3.6.2c. The tracking error, that is the sliding variable in (3.2.6), is shown in



(a) Eastward currents

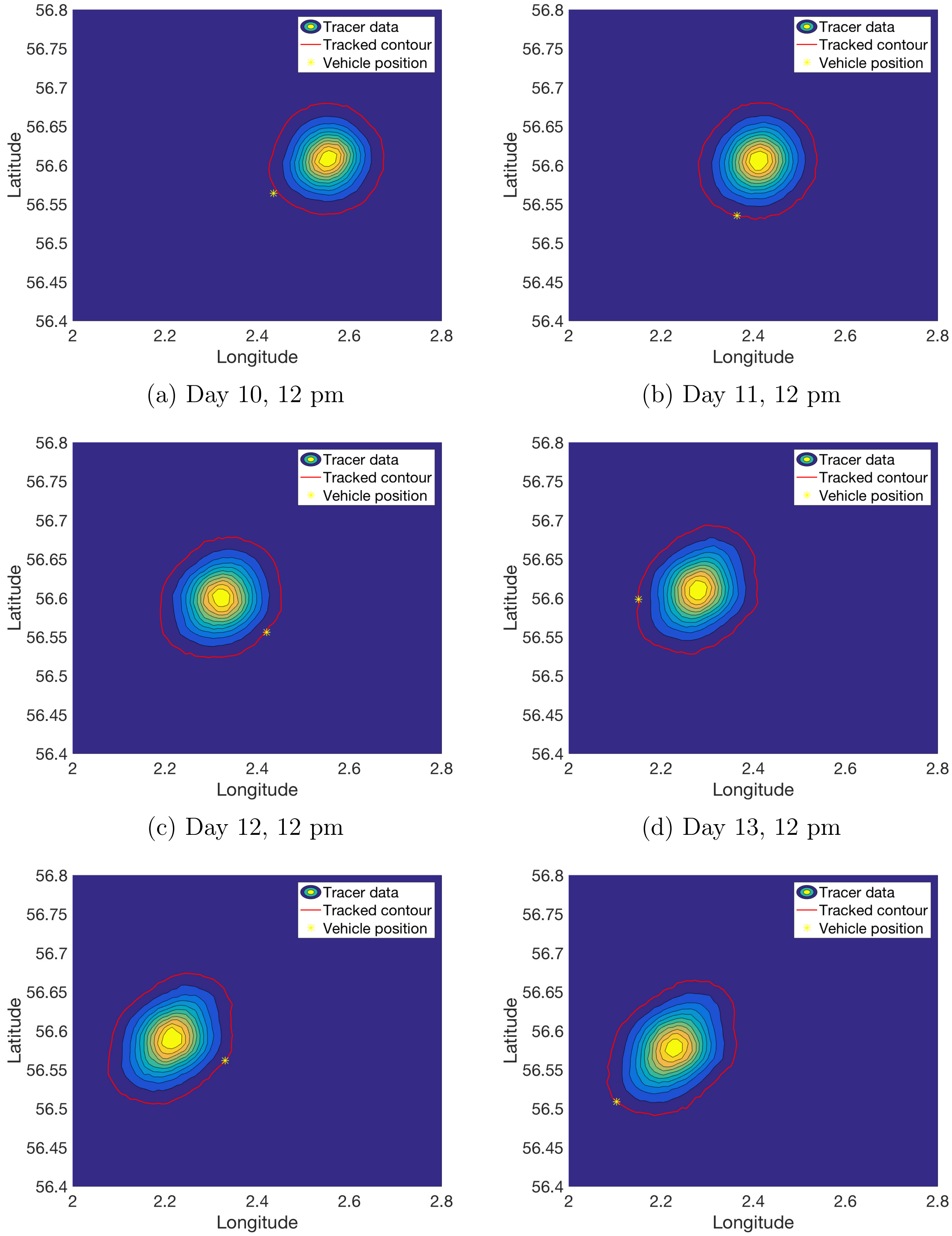
(b) Northward current

Figure 3.6.3: Water current distribution at the beginning of day 10

Fig. 3.6.2d. The tracking error is within the range of ±5 particles for most of the simulation, with exceptions corresponding to the initial phase of the simulation and to sharp turns in the tracked contour. The initially big tracking error, of the order of 50 particles/unit in absolute value, is quickly reduced through a short series of oscillations. When sharp turns are encountered, the tracking error increases quickly, but it is always successfully reduced through a similar series of decreasing amplitude oscillations. After approximately 6 hours from the beginning of the simulation, the oscillations due to the tracked contour's sharp feature highlighted in Fig. 3.6.2c are visible in Fig. 3.6.2d. Even in this case, after a few oscillations, tracking is recovered. This highlights the capability of the suboptimal sliding mode boundary tracking algorithm to recover when sliding is temporarily lost.

3.6.2 Simulation results - dynamic case

In this section, the suboptimal sliding mode boundary tracking approach is applied to the dynamic dataset describing the evolution of the tracer patch. The algorithm is capable of imposing dynamic boundary tracking, provided certain assumption on the spatial field evolution hold. Specifically, the two-dimensional boundary needs to be sufficiently slow moving with respect to the vehicle, as also mentioned in [39] and [41]. If this condition is satisfied, the vehicle successfully tracks the moving boundary.



(a) Day 10, 12 pm

(b) Day 11, 12 pm

(c) Day 12, 12 pm

(d) Day 13, 12 pm

(e) Day 14, 12 pm

(f) Day 15, 12 pm

Figure 3.6.4: Dynamic tracer release boundary tracking - screen shots showing the spatial field conformation and the vehicle's position

The considered dynamic dataset describes the evolution of the tracer patch between day 10 and day 16, with a temporal resolution of 5 minutes. The simulation set up is the same as in Section 3.6.1, the only exception being the simulation time,

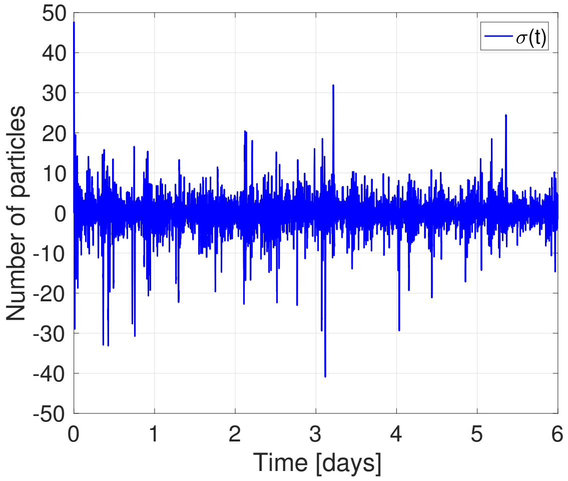


Figure 3.6.5: Dynamic tracer release boundary tracking - tracking error

extended to 6 days. The kinematic model of the vehicle in (3.3.17), affected by the drift terms ΔV_{x_b} and ΔV_{y_b}, is considered. The dynamic dataset, in fact, accounts for the presence of the sea surface water currents at every position of the vehicle in the East and North directions. The contour levels of the Eastward and the Northward currents at the beginning of day 10 are shown in Fig. 3.6.3a and Fig. 3.6.3b respectively. The sea surface water currents, v_x and v_y in the East and North directions respectively, are upper bounded within the six days of the dataset, with $v_{x,max} = 0.2996$ m/s and $v_{y,max} = 0.2273$ m/s. Being this bounds smaller than the speed of the vehicle ($V = 1$ m/s), the spatial field can be assumed to be sufficiently slow moving with respect to the vehicle, verifying the assumption in [39]. Hence, the vehicle is expected to successfully track the time-varying tracked contour.

A sequence of screenshots of the results from the simulation based on the dynamic dataset is shown in Fig. 3.6.4. Each screenshot corresponds to the initial instant of one of the six days of the simulation. It is visible how the shape of the tracked contour changes over the simulation time, as well as its location in the considered area. Particularly, the tracked contour changes from being almost perfectly circular to an ellipsoid shape, while it moves towards West. In each frame, the position of the vehicle is highlighted with a yellow star and the tracked contour is shown in red. Despite not being on the tracked contour at the beginning of the simulation (Fig. 3.6.4a), the vehicle appears on the tracked contour for all the subsequent time instants. This suggest a good tracking performance, which is confirmed by the tracking error, shown in Fig. 3.6.5. Despite the presence of oceanic currents affecting the vehicle's movement, the movement of the patch itself and the changes in the shape of the tracked contour, the tracking error is contained in the range of ±10 particles for the whole length of the simulation, with the exception of a few peaks. These, as previously mentioned, are due to the limited turning capability of the vehicle when the tracked contour shows sharp features, and to the relatively slow control action update rate. These results demonstrate the applicability of the suboptimal sliding mode boundary tracking approach to dynamic fields. In addition, they motivate the requirement of an explicit bound on the vehicle's speed, aimed at guaranteeing the capability of the vehicle to track the boundary of a dynamic spatial field.

3.7 Concluding remarks

The suboptimal sliding mode boundary tracking algorithm, originally proposed in [39], is presented in this chapter. This boundary tracking algorithm has been chosen amongst the single vehicle boundary tracking strategies presented in Section 2.2.1 because of its low information requirement and its reduced computational load. In this chapter, the algorithm is applied to a single autonomous vehicle tracking the boundary of a two-dimensional oceanic feature. Differently from [39], where the algorithm was applied to a simple and static scenario, the approach is applied in this chapter in the presence of external disturbances and in the presence of a time-varying spatial field. The approach is applied in simulations to the exploration of at tidal-mixing front area, in which the vehicle is required to track a contour characterized by a constant value of the water temperature, and in the exploration of the boundary of a tracer release. The obtained results practically demonstrate the efficacy of the suboptimal sliding mode boundary tracking algorithm in these scenarios. Additionally, the approach results robust in the presence of external disturbances, the water currents acting on the vehicle, and in the presence of a time-varying spatial field. The original approach in [39] is based on the assumption that the vehicle is initially deployed in a vicinity of the tracked contour. It is shown in the chapter that, if this hypothesis is not verified, then the approach fails to drive the vehicle towards and along the tracked contour. This highlights one possible development of the approach, that is the introduction of an initial seeking strategy, to drive the vehicle from an arbitrary initial deployment to a vicinity of the tracked contour.

Chapter 4

Three dimensional boundary tracking with an underwater glider

4.1 Introduction

Often spatial phenomena appear in three dimensions, as it is manifest considering underwater oceanic features or atmospheric phenomena. In this chapter, an algorithm to explore a three-dimensional feature with a single autonomous vehicle is proposed. As highlighted in the literature review in Section 2.2.1, there are only few algorithms which are applicable to track the boundaries of three-dimensional features. The suboptimal sliding mode boundary tracking strategy, proposed in [39] and presented in detail in Chapter 3, is extended in this chapter and its applicability to the exploration of three-dimensional oceanic features is investigated. Particularly, the algorithm is employed to control the movement of an underwater glider performing boundary tracking.

The chapter is organized as follows: an introduction to the working principle and the movement possibilities of underwater gliders is given in Section 4.2. A kinematic model and a complete dynamical model of an underwater glider are presented in Section 4.3. Details about the model construction, used for the validation of the approach in a simulated environment, are given in Section 4.4. The proposed three-

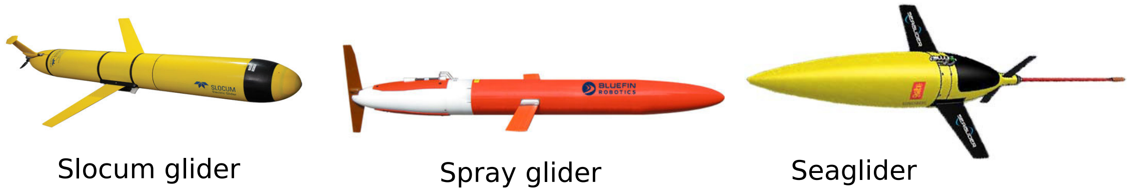


Figure 4.1.1: A Slocum glider, a Spray glider and a Seaglider

dimensional boundary tracking strategy is introduced in Section 4.5. The approach is validated firstly on a numerical example, in Section 4.6, and then on a realistic dataset, relating to the water temperature in the Iceland Faroes Front area, in Section 4.7. Concluding remarks are given in Section 4.8, while complementary details relating to the model components are given in Appendix B.

4.2 Underwater gliders

Underwater gliders are long endurance, buoyancy propelled autonomous vehicles for the ocean exploration [93]. The original prototype, the Slocum glider, was named after Joshua Slocum, the first man to sail single-handed around the world on a small vessel [11]. Slocum gliders are characterized by a long endurance, of the order of several months [94], as their movement is mainly generated by natural forces, exerted by the surrounding fluid. Therefore, Slocum gliders do not relay on a propeller. This distinguishes them from electrically driven underwater vehicles, such as Autosub [9], which can operate for a maximum of a few tens of hours at a time [95]. Different types of underwater gliders have been subsequently designed, such as the Spray glider [12] and the Seaglider [10]. These different typologies of gliders, shown in Fig. 4.1.1, are based on the same working principles, but are characterized by slightly different designs and actuator strategies, as it will be highlighted in the following.

The main drawbacks of the operation of underwater gliders are the movement constraints and the limited operational maximum velocities - of the order of $20 - 30$ cm/s in the horizontal plane [95]. Additionally, differently from ASVs, underwater gliders have limited communication capabilities. Gliders can communicate only during the time they spend on the surface, as underwater communication strategies, such as acoustic techniques, are constrained in terms of range and bandwidth. Due to the limited communication possibilities, the risks associated with the operation of this type of vehicles are generally high, as it is not possible to remotely monitor their position and state while they are underwater [96].

As previously mentioned, underwater gliders are non-propelled, and their motion is generated exploiting the forces exerted by the fluid in which they are immersed. The *net buoyancy mass* is defined as the difference between the mass of a body immersed in a fluid and the mass of the fluid displaced by the body itself. When the net buoyancy mass is positive, the body sinks, while it moves towards the surface when the net buoyancy mass is negative. Gliders have the capability of modifying their net buoyancy mass through a buoyancy engine, which therefore is used to influence their direction of movement in the vertical plane. The change in buoyancy can be achieved with different methods, which modify the density of the vehicle with respect to the density of the surrounding fluid. Electric Slocum gliders make use of a pumping system, which modifies the quantity of water inside a ballast tank [97]. The buoyancy modifications for Thermal Slocum gliders are achieved through the phase change of an internal working fluid [98]. Spray gliders modify their volume, pumping an operational fluid between a reservoir internal to the body of the glider and an external bladder; a similar principle is exploited by Seagliders.

The movement originated by the buoyancy changes is purely in the vertical plane. The horizontal component of the glider's velocity is originated through the lift force exerted by the vehicle's wings, which allows gliders to move forward. The resulting movement is typically along so-called *sawtooth* trajectories [11].

The direction of movement of the gliders in the horizontal plane can be influenced in different ways: Seagliders and Spray gliders exploit an internal moving mass, such as a set of batteries, whose movement influences the mass distribution of the vehicle, while Slocum gliders regulate the direction of movement in the horizontal plane through a movable *rudder*.

Gliders are increasingly being used in ocean sampling and monitoring because of their reduced operational costs. Usually, gliders are operated in a *semi-autonomous* way [19], with a pre-planned mission, composed of a sequence of WPs. Gliders autonomously attain the pre-planned mission through their actuators actions. Specifically, underwater gliders make use of the available measurements to estimate their current position. The position estimate is used to design the required control actions aimed at sequentially reaching the given WPs.

When sampling an a-priori unknown feature, the set of waypoints is defined in order to achieve a trade-off between the coverage of the area of interest and the operational costs. Typically, suboptimal zig-zag or lawnmower like trajectories of WPs are used for this purpose [99]. The study of techniques for the optimization of the pre-planned trajectories, in terms of glider safety and power consumption, is ongoing [21]. In [100] the WPs for a fleet of gliders are designed in order to make the gliders move in a predefined formation, while studying the upwelling phenomenon in Monterey Bay. In [101], the preplanned trajectory is iteratively updated using the collected data about an harmful algae bloom phenomenon and the available prediction capability.

Another critical aspect of the waypoint based operation of underwater gliders is the fact that, while underwater, gliders rely on an *estimate* of their actual position. This estimate is built using the last position stamp on the surface, the measured depth and the measured orientation of the glider. This estimation procedure is known as *dead-reckoning* [100], [102]. Inaccurate dead-reckoning may cause major discrepancies between the goal mission (the trajectory created by the waypoints) and the actual trajectory taken by the glider. Consequently, research aimed at improving the dead-reckoning methodologies is active [103].

In this chapter, the possibility of overcoming the difficulties associated with the

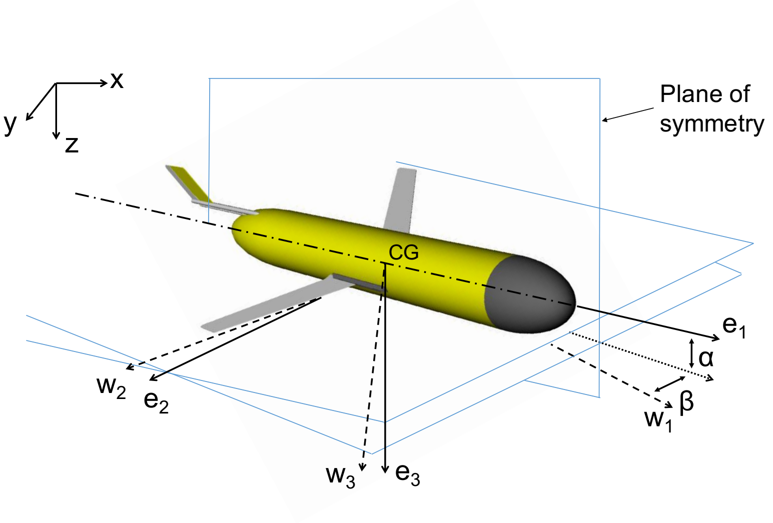


Figure 4.3.1: Reference frames: $\{I\} = \{x, y, z\}$, $\{B\} = \{e_1, e_2, e_3\}$, $\{W\} = \{w_1, w_2, w_3\}$

path planning and the dead reckoning techniques is considered by operating an underwater glider in a fully autonomous way.

4.3 Model of an underwater glider

A complete eight degrees of freedom model of the underwater glider is presented in this section, which includes both kinematics and dynamics. This model describes the movement and the working principle of an Electric Slocum underwater glider. The reference frames required for the model construction are defined in Section 4.3.1. The kinematic model of the glider movement is introduced in Section 4.3.2. The glider working principle is described in detail in Section 4.3.3, while the complete model of the glider is given in Section 4.3.4. Additional details, relating to the glider model components, are given in Appendix B.

4.3.1 Reference frames

The three reference frames used in modelling the glider are shown in Fig. 4.3.1: (i) the *inertial* reference frame $\{I\} = \{x, y, z\}$; (ii) the *body* reference frame $\{B\} = \{e_1, e_2, e_3\}$; and (iii) the *wind* reference frame $\{W\} = \{w_1, w_2, w_3\}$.

The *inertial frame* $\{I\} = \{x, y, z\}$ is defined with the x and y axis lying in the horizontal plane, perpendicular to the gravity vector, and with the z axis parallel to the gravity vector, pointing downwards [104].

The vehicle's *body frame* $\{B\} = \{e_1, e_2, e_3\}$ has its origin at the centre of buoyancy of the glider, which is the centroid of the displaced volume of fluid. The body frame is defined with the e_1 axis parallel to the principal axis of the glider, the e_2 axis parallel to the wing plane and the e_3 axis orthogonal to both e_1 and e_2, to form a right-handed coordinate system. The change of coordinates from the body frame $\{B\}$ to the inertial frame $\{I\}$ is achieved through the rotation matrix $R \in SO(3)$. The rotation group $SO(3)$ is defined as:

$$SO(3) = \{R \in \mathbb{R}^{3\times3} : RR^\top = I_d, \quad \det(R) = 1\} \tag{4.3.1}$$

where $I_d \in \mathbb{R}^{3\times3}$ is the identity matrix and $\det(R)$ is the determinant of matrix R. The rotation matrix R is defined as a function of a triple of Euler angles: the *pitch* angle (θ), the *yaw* angle (ψ) and the *roll* angle (ϕ), which are shown in Fig. 4.3.2. Particularly, the pitch, yaw and roll angles are obtained from a rotation along the e_2, e_3 and e_1 axis respectively. The Euler angles describe the orientation of $\{B\}$ relative to $\{I\}$. Particularly, implying with the notation $R_y(k)$ a rotation about axis y of k degrees, the rotation matrix results:

$$R = R_{e_3}(\psi)R_{e_2}(\theta)R_{e_1}(\phi) \tag{4.3.2}$$

The *wind frame* $\{W\} = \{w_1, w_2, w_3\}$ is also centred at the vehicle's centre of buoyancy. The orientation of the wind frame with respect to the body frame is described through two aerodynamic angles: the *angle of attack* α and the *sideslip angle* β. Being $v = [v_1 \quad v_2 \quad v_3]^\top$ the glider's velocity vector expressed in $\{B\}$, the aerodynamic angles are defined as:

$$\alpha := \tan^{-1}\left(\frac{v_3}{v_1}\right), \quad \beta := \sin^{-1}\left(\frac{v_2}{\|v\|}\right) \tag{4.3.3}$$

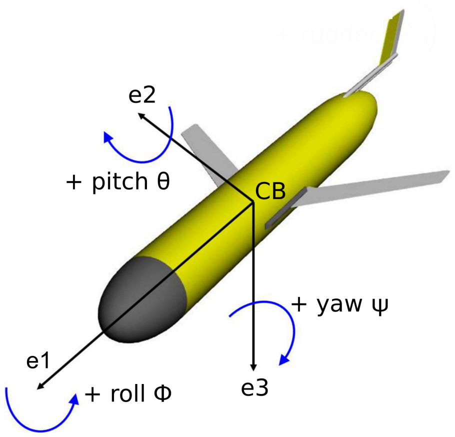


Figure 4.3.2: Euler angles definitions: pitch (θ), roll (ϕ) and yaw (ψ)

where $\|\cdot\|$ represents the Euclidean norm. The wind reference frame is obtained from $\{B\}$ by a rotation of α around w_2, described by $R_{w_2}(\alpha)$, and a rotation of β around w_3, given by $R_{w_3}(\beta)$. These rotations are fully described by the rotation matrix $R_{WB} \in SO(3)$, which performs the change of coordinates from the body frame $\{B\}$ to the wind frame $\{W\}$ [105]:

$$R_{BW} = R_{w_3}(\beta)R_{w_2}(\alpha) \tag{4.3.4}$$

4.3.2 Kinematic model

A kinematic model of the glider's movement in the three-dimensional space has been introduced in [104]. A similar kinematic model has been used in [106], where a guidance control for the homing and docking of an autonomous underwater vehicle is developed, and in [107], where a path-following scheme is proposed. The glider is considered as a point mass particle and its movement is described neglecting the underlying dynamics. The obtained kinematic model is:

$$\begin{cases} \dot{R} = R\hat{\Omega} \\ \dot{b} = Rv \end{cases} \tag{4.3.5}$$

where R is the rotation matrix defined in (4.3.2), $b = [x \quad y \quad z]^\top$ is the position vector of the glider in $\{I\}$ and v, $\Omega \in \mathbb{R}^3$ are the vectors of the longitudinal and angular velocities of the glider in $\{B\}$. The operator $\hat{\ }$, maps the vector $\Omega \in \mathbb{R}^3$ to the skew symmetric matrix, as follows[1]:

$$\hat{\Omega} = \begin{bmatrix} 0 & -\Omega_3 & \Omega_2 \\ \Omega_3 & 0 & -\Omega_1 \\ -\Omega_2 & \Omega_1 & 0 \end{bmatrix} \tag{4.3.6}$$

The kinematic model has been defined neglecting the presence of oceanic currents influencing the movement of the glider. Oceanic currents may affect both the direction of movement of the glider and its speed [107] and may be included in the glider model modifying (4.3.5) as [104]:

$$\dot{b} = Rv + v_{\text{currents}} \tag{4.3.7}$$

where $v_{\text{currents}} = [v_x \quad v_y \quad v_z]^\top$ is the velocity vector of the water currents, expressed in $\{I\}$.

The glider is assumed to move at *flight-level* [108], that is with a null roll angle ϕ. This, in [109], is considered as a nominal equilibrium condition of the glider movement, and any deviation from the flight-level movement is treated as a perturbation. Considering the full expression of the rotation matrix in (4.3.2):

$$R = \begin{bmatrix} \cos\psi\cos\theta & -\sin\psi\cos\phi + \cos\psi\sin\theta\sin\phi & \sin\psi\sin\phi + \cos\psi\cos\phi\sin\theta \\ \sin\psi\cos\theta & \cos\psi\cos\phi + \sin\psi\sin\theta\sin\phi & -\cos\psi\sin\phi + \sin\theta\sin\psi\cos\phi \\ -\sin\theta & \cos\theta\sin\phi & \cos\theta\cos\phi \end{bmatrix},$$

and assuming that the glider moves at flight level ($\phi = 0$), the position of the glider

[1] As a consequence, $\hat{a}b = a \times b$, $\forall\ a, b \in \mathbb{R}^3$.

evolves according to the following kinematic equations:

$$\begin{aligned}
\dot{x} &= v_1 \cos\psi \cos\theta - v_2 \sin\psi + v_3 \cos\psi \sin\theta \\
\dot{y} &= v_1 \sin\psi \cos\theta + v_2 \cos\psi + v_3 \sin\psi \sin\theta \\
\dot{z} &= -v_1 \sin\theta + v_3 \cos\theta
\end{aligned} \tag{4.3.8}$$

This model describes the evolution of the glider's position in the inertial frame as a function of its velocity vector v and its Euler angles. Similarly to Chapter 3, the kinematic model is used at a *guidance* level, in order to define the glider's desired yaw angle (ψ^*), which is the guidance control to be applied to the vehicle. The guidance control is subsequently applied in this chapter to a complete nonlinear model of the underwater glider.

4.3.3 Working principle

The movement of underwater gliders in the three-dimensional space is controlled through the ballast mass (m_B), the movable mass position along the e_1 axis (r_{p1}) and the rudder deflection (δ_R). These controlled variables are shown in Fig. 4.3.3.

The movement of underwater gliders in the vertical plane is determined by the effect of buoyancy forces [110]. The *net buoyancy mass* is defined as

$$m_0 := m_v - m \ , \tag{4.3.9}$$

where m_v is the total mass of the glider and m is the mass of the displaced fluid. Particularly, the mass of the glider is obtained as:

$$m_v = m_H + m_B + \bar{m} + m_w \tag{4.3.10}$$

where m_H is the hull mass, which is assumed to be uniformly distributed along the body of the glider, m_B is the ballast point mass, $\bar{m}$ is the moving internal mass

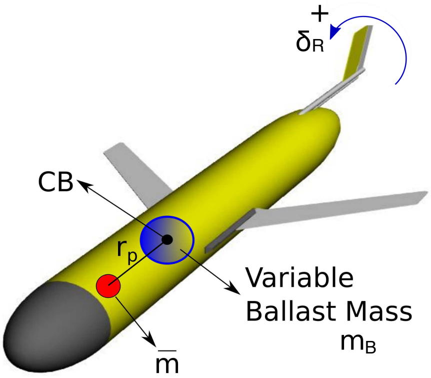


Figure 4.3.3: Ballast mass (m_B), movable mass ($\bar{m}$) and rudder angle (δ_R) definitions

and m_w is a fixed offset mass with respect to the centre of buoyancy. The glider mass, and hence the net buoyancy mass, are controlled by modifying the *ballast mass* m_B. If the net buoyancy mass in (4.3.9) is positive, the glider is heavier than the displaced fluid and it sinks; and vice versa.

The buoyancy control determines the movement along the z axis. Through the lift force introduced by the vehicle's wings, the vehicle is capable of moving forward. The pitch angle θ, shown in Fig. 4.3.2 is controlled through the position of the movable internal mass $\bar{m}$. The movable mass position in the glider's body frame is $r_p = [r_{p1} \quad r_{p2} \quad r_{p3}]^\top$. Typically r_{p2} and r_{p3} are fixed [104]; the controlled variable is the position of the movable mass $\bar{m}$ along the e_1 axis, which is identified as r_{p1}. The movement of $\bar{m}$ along the e_1 axis modifies the position of the glider centre of gravity with respect to the centre of buoyancy, and the obtained offset between the centre of gravity and the centre of buoyancy causes a gravitational torque [104].

Finally, the direction of movement of the glider in the lateral plane is controlled through a movable rudder. The controlled variable, which is the rudder deflection angle δ_r, is shown in Fig. 4.3.3. The rudder deflection affects the hydrodynamic forces influencing the movement of the vehicle and therefore determines its rotation with respect to the e_3 axis, thus affecting the vehicle's yaw angle ψ.

4.3.4 Complete model

The complete model of the glider is composed of a kinematic and a dynamic component. The dynamic model of the glider is based on Newton's second law in the inertial coordinates system and is derived in [104]. The following simplifying hypothesis, proposed in [104] and [109] are imposed: (i) the glider is assumed to be a rigid elliptical body with fixed wings and tail, symmetric with respect to both the (e_1, e_3) and the (e_1, e_2) planes; (ii) the centre of buoyancy of the gliders is assumed to be fixed at the centre of the ellipsoid representing the vehicle body; (iii) the buoyancy mass m_B is assumed to be located at the position of the centre of buoyancy; (iv) the movable mass $\bar{m}$ is constrained to move along the principal axis of the glider (e_1); (v) the offset mass m_w is assumed null. The resulting complete model is [104]:

$$\begin{bmatrix} \dot{R} \\ \dot{b} \\ \dot{\Omega} \\ \dot{v} \\ \dot{r}_p \\ \ddot{r}_p \\ \dot{m}_B \\ \dot{\delta}_R \end{bmatrix} = \begin{bmatrix} R\hat{\Omega} \\ Rv \\ J^{-1}T \\ M^{-1}F \\ \dot{r}_p \\ \omega_p \\ u_{BR} \\ u_{\delta_R} \end{bmatrix} \tag{4.3.11}$$

where M and J are the total mass and inertia matrices of the system including the glider and the added mass. The terms T and F in (4.3.11) are the total torque and force in the simplified mass system, obtained applying the introduced hypothesis. These include the contributions of the control actions and the hydrodynamical forces F_{ext} and torques T_{ext}.

The control actions affecting the dynamics of the glider are the rate of variation of the ballast mass, the acceleration applied to the movable mass and the rudder deflection. As reported in the literature in [107] [108] and [111], the rudder control can be designed independently of the buoyancy and pitch control, and vice versa.

Consequently, the glider control in the lateral plane and in the vertical plane can be assumed as decoupled.

In Electric Slocum gliders, buoyancy control is performed through an electrically driven pumping system, which modifies the quantity of water inside the ballast tank [11]. Because of the operation of the electric pump, which relies on a set of batteries, buoyancy control is costly. Consequently, the traditional operation of electric Slocum gliders does not account for frequent variations of the vehicle buoyancy, as the endurance of the non-propelled vehicles needs to be maximized. Traditionally, gliders are equipped with a depth sensor and they are constrained to operate in an allowed depth range: $z \in [z_{min}, z_{max}]$. If $z < z_{min}$, then the vehicle is near the surface and, in order to continue the mission, it needs to revert its direction of movement in the vertical plane and start sinking. Consequently, $m_0 > 0$ is desired and the ballast tank is partly or fully filled. When $z > z_{max}$, on the contrary, $m_0 < 0$ is required and hence the ballast tank is partly or fully emptied. The ballast mass control is thus activated only when one of the two depth thresholds is exceeded. This generates the typical *saw-tooth* trajectory of the glider in the longitudinal plane, an example of which is depicted in Fig. 4.3.4. The ballast mass control is constrained by the maximum flow rate $\dot{m}_{B,max}$, and by the maximum net buoyancy mass $m_{0,max}$.

The pitch control modifies the position of the movable mass $\bar{m}$ along the e_1 axis and makes the glider move with the desired orientation in the vertical plane. In Slocum gliders, the movable mass is typically controlled through a proportional acceleration input as [112]:

$$\omega_p = K_p(\theta - \theta^*) \tag{4.3.12}$$

where θ is the measured pitch angle, θ^* is the desired pitch angle, and K_p is the proportional gain. The proportional pitch controller is limited by the maximum allowed acceleration applicable to the movable mass $\omega_{p,max}$ and by the range of movement of the movable mass $\bar{m}$ along the e_1 axis of the glider body frame. In fact, it is

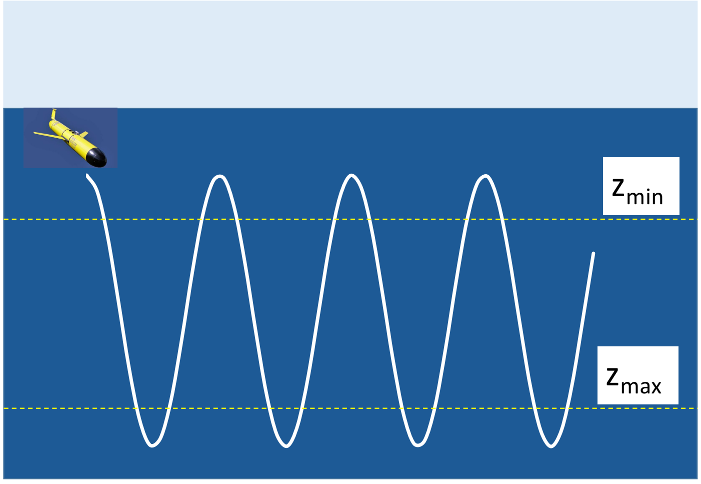


Figure 4.3.4: Example of a glider sawtooth path - z_{min} and z_{max} are the predefined depth thresholds.

required that $r_{p1} \in [r_{p1,min}, r_{p1,max}]$.

The direction of movement of the glider in the lateral plane is controlled through the deflection of the rudder δ_R. Tracking of a desired yaw angle ψ^* in the lateral plane is achieved by controlling the rudder deflection through a proportional controller similar to (4.3.12) [112]:

$$u_{\delta_R} = K_{P\delta_R}(\psi - \psi^*) \tag{4.3.13}$$

where ψ is the measured yaw angle of the vehicle and $K_{P\delta_R}$ is the proportional controller gain. The rudder controller is limited by the maximum rudder deflection $\delta_{R,max}$ and angular acceleration $\dot{\delta}_{R,max}$.

Additional details on the complete model in (4.3.11) are given in Appendix B.

4.4 Model construction

The parameters of the nonlinear model of a typical Slocum glider are given in Table 4.4.1 and have been obtained from [107] and [113]. The complete nonlinear model of the glider in (4.3.11) is implemented in a Matlab/Simulink environment (version R2016b). The amplitudes and the rate limits associated with the control actions have been accounted for. The maximum amplitude of the ballast mass controller, u_{BR}, for Slocum gliders, is $\dot{m}_{B,max} = 1\ g/\mathrm{s}^2$, while the maximum net buoyancy is

Table 4.4.1: Slocum glider parameters used for the dynamic model construction

Parameter	**Symbol**	**Value**	**Unit**
Hull mass	m_H	40	kg
Ballast tank mass	m_B	$[0.75, 1.25]$	kg
Movable mass	$\bar{m}$	9	kg
Displaced fluid mass	M_f	$\begin{bmatrix} 5 & 0 & 0 \\ 0 & 60 & 0 \\ 0 & 0 & 70 \end{bmatrix}$	kg
Displaced fluid inertia	J_f	$\begin{bmatrix} 4 & 0 & 0 \\ 0 & 12 & 0 \\ 0 & 0 & 11 \end{bmatrix}$	kg $\cdot$m^2
Hydrodynamic coefficients	K_{D0}	2	kg/m
	K_D	45	kg/m/rad^2
	K_β	20	kg/m/rad
	K_{L0}	0	kg/m
	K_L	135	kg/m/rad
	K_{MR}	-60	kg/rad
	K_{Q1}	-20	kg $\cdot$ s/rad
	K_{M0}	0	kg
	K_M	-50	kg/rad
	K_{Q2}	-60	kg $\cdot$ s/rad
	K_{My}	100	kg/rad
	K_{Q3}	-20	kg $\cdot$ s/rad
	$K_{\delta_R\beta}$	10	kg/m/rad
	K_{δ_R}	2	kg/m/rad
	$K_{\delta_R M}$	2	kg/m/rad

$m_{0,max} = 0.25$ kg [104]. The maximum amplitude of the pitch controller ω_p is, for Slocum gliders, $\omega_{p,max} = 0.001$ m/s^2, while the movement of mass $\bar{m}$ is limited within $[r_{p1,min}, r_{p1,max}] = [-0.05; 0.05]$m [104]. Finally, the maximum allowed rudder deflection is $\delta_{R,max} = \pm 20°$, and the maximum angular acceleration of the rudder is limited by $\dot{\delta}_{R,max} = \pm 1°/\text{s}$ [107].

4.4.1 Trim algorithm

In order to obtain an equilibrium glider dynamics, the choice of the initial conditions for the model in (4.3.11) is crucial. A trim algorithm has been used in order to determine the glider set up that guarantees a steady glide. Trim conditions are often used as initial conditions for the simulations, in both glider and aircraft literature, as they represent a stable movement set up [105], [114]. The solutions of the trim

Table 4.4.2: Upper and lower bounds on the trim parameters for the longitudinal trim algorithm

	v_1	v_2	v_3	θ	r_{p1}	m_B
unit	m/s	m/s	m/s	rad	m	kg
l_b	0	0	0	$-\pi/2$	-0.05	0.75
u_b	0.5	0.5	0.5	$\pi/2$	0.05	1.25

algorithm are the *equilibrium glides*, in which the glider can stay indefinitely, without the effect of any control action [113]. The equilibrium glides can be divided into two categories: (i) *straight glides*, in which the glider moves on a straight line with constant yaw, pitch and roll angles; (ii) *helix glides*, in which the glider moves with constant pitch and roll and it varies its heading at a constant rate, moving on an helix trajectory. The equilibria are obtained through the solution of a constrained optimization problem based on a cost function and on a set of constraints, which account for the physical limits previously introduced.

The following cost function has been chosen as in [115]:

$$F(\xi) = \dot{v}_1^2 + \dot{v}_2^2 + \dot{v}_3^2 + \dot{\Omega}_1^2 + \dot{\Omega}_2^2 + \dot{\Omega}_3^2 + \dot{\theta}^2 + \dot{\phi}^2 + \dot{\psi}^2 \tag{4.4.1}$$

where ξ is the vector of states and control actions that can be modified in order to find the trim solution. The complete vector ξ has been chosen as:

$$\xi = \begin{bmatrix} v_1 & v_2 & v_3 & \Omega_1 & \Omega_2 & \Omega_3 & \theta & \phi & \psi & r_{p1} & m_B & \delta_R \end{bmatrix}^\top \tag{4.4.2}$$

The minimization of the cost function in (4.4.1) is performed by acting on the parameters in (4.4.2), some of which may be considered fixed at specific values. Consequently, the components of the speed and angular velocity of the vehicle, the Euler angles and the controlled variables can be modified in order to minimize the cost function in (4.4.1). The minimization of the cost function $F(\cdot)$ is performed numerically and this is done in Matlab through the routine `fmincon` [116], which allows the specification of bounds and linear and nonlinear constraints on the tunable parameters ξ.

Table 4.4.3: Longitudinal trim algorithm solution with $r_{p3} = 0.05$ m and $\bar{V} = 0.4$ m/s. For each desired pitch angle θ, the trim algorithm is used to determine the movable mass position along the e_1 axis, the components of the vehicle's velocity in the body frame (v_1, v_2, v_3) and the buoyancy mass (m_B). The value of the cost function $F(\xi)$ at the end of the trim algorithm solution is given.

θ	r_{p1}	v_1	v_2	v_3	m_b	$F(\xi)$
°	m	m/s	m/s	m/s	kg	-
-5	0.0069	0.3982	0.0001	0.0378	1.2117	$1.7 \cdot 10^{-7}$
-10	0.0106	0.3991	0.0001	0.0265	1.1500	$1.6 \cdot 10^{-7}$
-15	0.0148	0.3995	0.0001	0.0195	1.1128	$1.6 \cdot 10^{-7}$
-20	0.0193	0.3997	0.0001	0.0151	1.0897	$1.6 \cdot 10^{-7}$
-25	0.0245	0.3998	0.0001	0.0121	1.0744	$1.6 \cdot 10^{-7}$
-30	0.0296	0.3999	0.0001	0.0100	1.0638	$1.6 \cdot 10^{-7}$
-35	0.0357	0.3999	0.0001	0.0083	1.0561	$1.6 \cdot 10^{-7}$
-40	0.0426	0.3999	0.0001	0.0070	1.0504	$1.6 \cdot 10^{-7}$
-45	0.0500	0.4000	0.0001	0.0020	1.0393	$3.6 \cdot 10^{-6}$
-26	0.0253	0.3998	0.0001	0.0117	1.0720	$1.6 \cdot 10^{-7}$

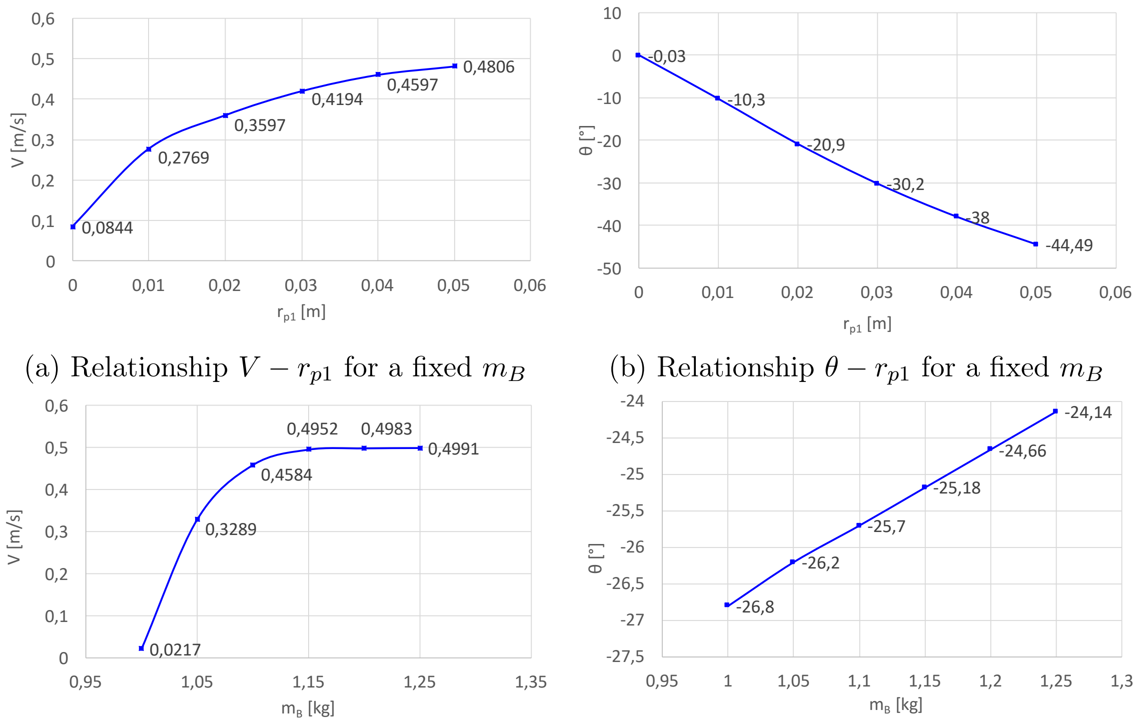
4.4.2 Trim algorithm solutions

Firstly, the movement of the glider has been constrained to the longitudinal plane. In the longitudinal plane, the glider is assumed to move at flight level ($\phi = 0$), with constant Euler angles ($\Omega_1 = \Omega_2 = \Omega_3 = 0$), null yaw angle ($\psi = 0$) and null rudder deflection ($\delta_R = 0$). Hence, the number of free parameters in (4.4.2) is reduced to six, which are $[v_1 \quad v_2 \quad v_3 \quad \theta \quad r_{p1} \quad m_B]^\top$. The upper and lower bounds, u_b and l_b respectively, for each of these six parameters are given in Table 4.4.2. These bounds become amplitude constraints for the free parameters in the optimization problem solution. Additionally, if the speed of the glider is fixed to a desired level ($\bar{V}$), then a nonlinear constraint needs to be specified; in detail: $\sqrt{v_1^2 + v_2^2 + v_3^2} - \bar{V} = 0$ has been used.

Tables 4.4.3 and 4.4.4 collect a set of solutions for the trim algorithm. The flight level equilibria with a fixed gliding velocity of $\bar{V} = 0.4$ m/s are given in Table 4.4.3. The desired value of the glider pitch angle θ is varied between $-5°$ and $-45°$. In Table 4.4.4, the desired pitch angle is set to $\theta = -26°$, which is the nominal glide angle for Slocum gliders [104], and the desired glide velocity $\bar{V}$ is varied from 0.1 m/s to 0.5

Table 4.4.4: Longitudinal trim algorithm solution with $r_{p3} = 0.05$ and $\theta = -26°$. For each desired glide speed $\bar{V}$, the trim algorithm is used to determine the movable mass position along the e_1 axis, the components of the vehicle's velocity in the body frame (v_1, v_2, v_3) and the buoyancy mass (m_B). The value of the cost function $F(\xi)$ at the end of the trim algorithm solution is given.

$\bar{V}$	r_{p1}	v_1	v_2	v_3	m_B	F(ξ)
m/s	m	m/s	m/s	m/s	kg	–
0.1	0.0245	0.0998	0.0002	0.0062	1.0072	$1.9 \cdot 10^{-7}$
0.2	0.0246	0.1999	0.0001	0.0068	1.0196	$1.8 \cdot 10^{-7}$
0.3	0.0249	0.2999	0.0001	0.0090	1.0411	$1.65 \cdot 10^{-7}$
0.4	0.0253	0.3998	0.0001	0.0117	1.0720	$1.6 \cdot 10^{-7}$
0.5	0.0257	0.4998	0.0001	0.0145	1.1117	$1.6 \cdot 10^{-7}$



(a) Relationship $V - r_{p1}$ for a fixed m_B

(b) Relationship $\theta - r_{p1}$ for a fixed m_B

(c) Relationship $V - m_B$ for a fixed r_{p1}

(d) Relationship $\theta - m_B$ for a fixed r_{p1}

Figure 4.4.1: Trim solutions - relationships between parameters

m/s. Fig. 4.4.1a and Fig. 4.4.1b consider the case when m_B is fixed at the value obtained for the glider nominal pitch angle $\theta = 26°$, specifically $m_B = 1.0720$ kg, and various fixed positions of $\bar{m}$ along the e_1 axis (r_{p1}) are considered. For each value of r_{p1}, the obtained glide velocity V and pitch angle θ are shown. If the movable mass shifts towards the nose of the glider, thus increasing r_{p1}, both the glider speed and the pitch angle are increased. The growth of the glider speed is saturated by the maximum speed imposed through an additional nonlinear constraint, specifically: $V = \sqrt{v_1^2 + v_2^2 + v_3^2}$ m/s ≤ 0.5. The results in Fig. 4.4.1c and Fig. 4.4.1d, consider

Table 4.4.5: Upper and lower bounds on the trim parameters for the three-dimensional trim algorithm

	v_1	v_2	v_3	Ω_1	Ω_2	Ω_3	θ	ϕ	ψ	r_{p1}	m_B	δ_R
unit	m/s	m/s	m/s	rad/s	rad/s	rad/s	rad	rad	rad	m	kg	rad
l_b	0	0	0	0	0	0	$-\pi/2$	$-\pi/2$	$-\pi/2$	-0.05	0.75	$-\pi/4$
u_b	0.5	0.5	0.5	0.5	0.5	0.5	$\pi/2$	$\pi/2$	$\pi/2$	0.05	1.25	$\pi/4$

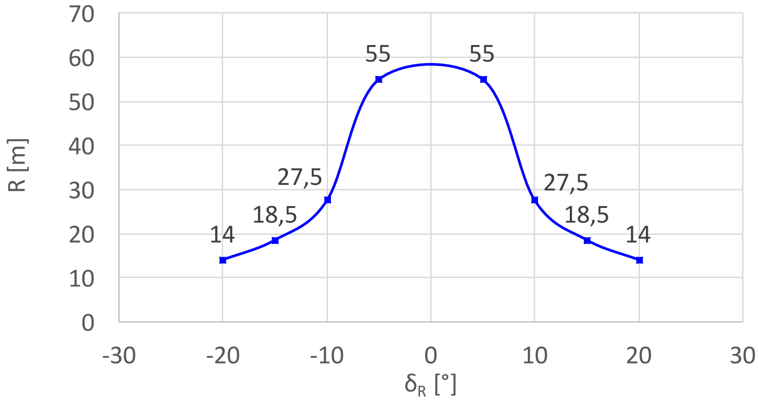
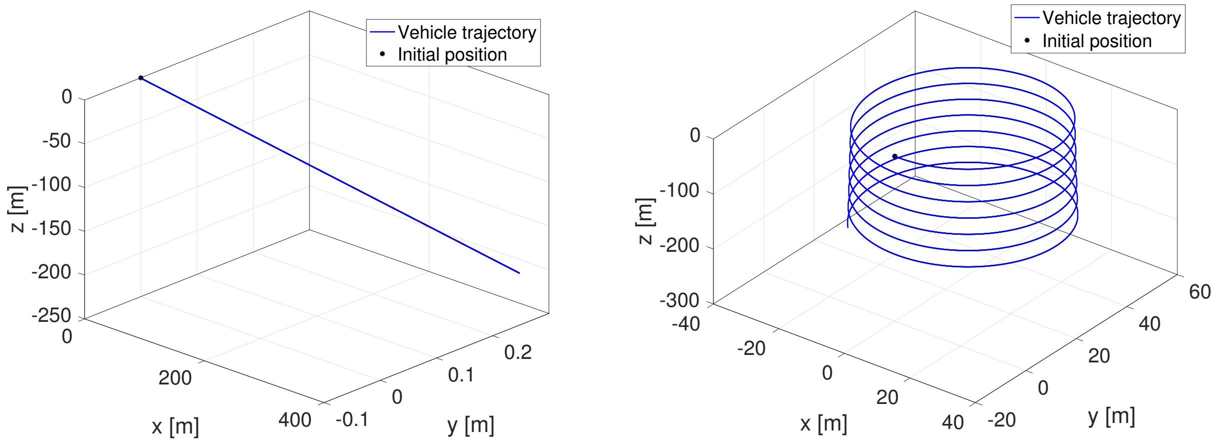


Figure 4.4.2: Trim solutions - relationship between the radius of the helix glide and the rudder deflection

the case when $r_p = 0.0253$ m is fixed and m_B is gradually increased. Increasing the ballast mass, with a fixed movable mass, increases the glider speed, but decreases the glider pitch angle.

When the constraint on the movement of the glider in the longitudinal plane is relaxed, the full vector of free parameters in (4.4.2) is considered. The lower and upper bound values l_b and u_b in this case are given as in Table 4.4.5. The trim algorithm is used to determine the initial conditions for the three-dimensional simulations. One particular set of solutions corresponds to the helix glides [115]. These solutions are obtained imposing the flight level movement ($\phi = 0$) and $\Omega_1 = \Omega_2 = 0$. The relationship between the rudder deflection and the radius of the obtained helix glide is shown in Fig. 4.4.2. Particularly, increasing the rudder deflection reduces the glider radius of curvature and hence the radius of the helix along which the glider moves. Also, having assumed that the glider is symmetric in the $e_1 - e_2$ plane, the effect of the rudder deflection results symmetrical.



(a) Straight equilibrium glide

(b) Helix equilibrium glide

Figure 4.4.3: Equilibrium glides

The movement of the glider along a straight and an helix glide is shown in Fig. 4.4.3. In Fig. 4.4.3a, the initial conditions corresponding to the trim solution in Table 4.4.3, with $\theta = -26°$, are used. The glider's trajectory is linear and its pitch, yaw and roll angles remain constant. The trajectory of the glider when moving along an helix equilibrium is shown in Fig. 4.4.3b. The initial conditions correspond here to the trim solution obtained with $\delta_R = 10°$, which generates a helix with radius $R \approx 27.5$ m.

4.5 Three-dimensional boundary tracking

A guidance strategy for an underwater glider is defined here, exploiting the kinematic model in Section 4.3.2. The output of the guidance strategy is the desired heading ψ^* to be maintained by the glider. This is then passed to the proportional controlled in (4.3.13), which ensures that the desired heading ψ^* is tracked using the rudder actuator of the glider.

The aim of the guidance strategy is to make the glider track a constant level set of an unknown three-dimensional oceanic feature. The unknown feature is represented as:

$$\gamma(x, y, z) : \mathcal{D} \subset \mathbb{R}^3 \to \mathbb{R} \tag{4.5.1}$$

which associates a value of the spatial field to each location (x, y, z) inside the compact set $\mathcal{D} \subset \mathbb{R}^3$. The spatial field $\gamma(x, y, z)$ is assumed satisfying the set of hypothesis introduced in Section 3.2. Let γ^* represent the value of interest of the feature for which the associated level set needs to be determined. The underwater glider is thus required to follow a γ^*- contour of the field. The level set is defined as:

$$\mathcal{B} = \{(x, y, z) \in \mathbb{R}^3 : \gamma(x, y, z) = \gamma^*\} \tag{4.5.2}$$

Additionally, similarly to Section 3.3.2 in Chapter 3, it is assumed that only the point measurement $\gamma(\cdot)$ at the location of the vehicle $(x(t), y(t), z(t))$ is taken and higher order information about the spatial field, such as the spatial gradient or the Hessian information, are unknown and not estimated.

Similarly to Section 3.2.3 in Chapter 3, the difference between the measurement at the position of the glider and the tracked value γ^* is defined as the sliding variable:

$$\sigma(t) = \gamma(x(t), y(t), z(t)) - \gamma^* \tag{4.5.3}$$

and the manifold $\mathcal{S} := \{(x, y, z) \in \mathcal{D} : \sigma(t) = 0\}$ is defined as the sliding surface.

The glider is assumed constrained to remain within a certain depth range in the vertical plane and to move with a constant pitch angle. The pitch and buoyancy controllers are used to guarantee this requirement. The boundary tracking control objective is achieved by influencing the movement of the vehicle in the lateral plane only.

For the boundary tracking problem definition, the kinematic model of the vehicle in (4.3.8) is used. The vehicle is therefore assumed to move at flight level, with $\phi = 0$. The pitch angle of the glider is assumed to be constant: considering the typical glider motion patterns, this is true for most of the operational time, with the only exception being the inflection movements. In the kinematic model in (4.3.8), the

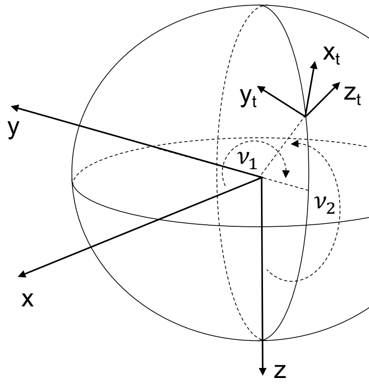


Figure 4.5.1: Tangent plane characterization

controlled variable results to be the heading of the glider, ψ. The proposed sliding mode boundary tracking algorithm is therefore aimed at defining the steering control $u(t) = \dot{\psi}(t)$ to obtain the desired heading $\psi^*(t)$. The desired heading $\psi^*(t)$, obtained through an Euler integration step, is then used by the rudder controller in (4.3.13), which determines the desired rudder deflection in order to obtain perfect tracking.

4.5.1 Proposed guidance strategy

Similarly to Section 3.3, taking the derivative of the sliding variable along the vehicle trajectory in (4.3.8) yields:

$$\begin{aligned}
\dot{\sigma} &= \frac{\partial \gamma}{\partial x}\dot{x} + \frac{\partial \gamma}{\partial y}\dot{y} + \frac{\partial \gamma}{\partial z}\dot{z} \qquad (4.5.4)\\
&= \|\nabla \gamma\| \left(\dot{x}\cos\nu_1 \sin\nu_2 + \dot{y}\sin\nu_1 \sin\nu_2 + \dot{z}\cos\nu_2\right)\\
&= \|\nabla \gamma\| \underbrace{[-v_1\left(\cos\theta \sin\nu_2 \cos(\psi - \nu_1) + \sin\theta\cos\nu_2\right) + v_2 \sin\nu_2 \sin(\psi - \nu_1)}_{a(t)}\\
&\quad \underbrace{-v_3(\sin\theta\sin\nu_2\cos(\psi-\nu_1) - \cos\theta\cos\nu_2)]}_{a(t)}\\
&= \|\nabla \gamma\|\, a(t)
\end{aligned}$$

where ν_1, ν_2 are the angles defining the tangent plane to the tracked manifold, characterized as in Fig. 4.5.1. The tangent plane reference frame is obtained from the inertial reference frame through a rotation of ν_1 around z and a rotation of ν_2 around y. Being $\{x_t, y_t, z_t\}$ the tangent plane reference frame, the gradient is oriented as z_t and its projection in the inertial reference frame is obtained as in (4.5.4).

Differentiating (4.5.4) once more yields:

$$\begin{aligned} \ddot{\sigma} &= a(t)\frac{d\,\|\nabla\gamma\|}{dt} + \|\nabla\gamma\|\,\dot{a}(t) \\ &= a(t)\frac{d\,\|\nabla\gamma\|}{dt} + \|\nabla\gamma\|\left[c(t)(\dot{\psi} - \dot{\nu}_1) + d(t)\right] \end{aligned} \tag{4.5.5}$$

with:

$$c(t) = (v_1\cos\theta + v_3\sin\theta)\sin\nu_2\sin(\psi - \nu_1) + v_2\sin\nu_2\cos(\psi - \nu_1) \tag{4.5.6}$$

$$\begin{aligned} d(t) &= (-v_1 - v_3)\cos\theta\cos\nu_2\cos(\psi - \nu_1) - v_3\sin\theta\cos\nu_2 \\ &\quad +v_1\sin\theta\sin\nu_2 + v_2\cos\nu_2\sin(\psi - \nu_1) \end{aligned} \tag{4.5.7}$$

A term dependent on $\dot{\psi}$, the control input, appears in (4.5.5). Hence, the dynamics relating the sliding variable in (4.5.3) to the yaw rate of the glider $\dot{\psi}$ has a constant relative degree two [85]. The relative degree of the system is not affected by the hypothesis of the glider moving with a constant pitch angle. Removing this hypothesis would also lead to a result of identical structure, with certain additional terms depending on the speed of the vehicle, its pitch and the angles ν_1 and ν_2 in the expression of $d(t)$.

Rearranging (4.5.5) as follows:

$$\ddot{\sigma} = \underbrace{\left(a(t)\frac{d||\nabla\gamma||}{dt} + d(t)||\nabla\gamma||\right)}_{\xi(t)} + \underbrace{c(t)||\nabla\gamma||}_{b(t)}(\dot{\psi} - \dot{\nu}_1)\ , \tag{4.5.8}$$

yield to the generic form in [84]:

$$\ddot{\sigma} = \xi(t) + b(t)(\dot{\psi} - \dot{\nu}_1) \tag{4.5.9}$$

Consistently with the control strategy in Chapter 3, the second order suboptimal sliding mode controller is used to steer the vehicle in the lateral plane. Being the movement of the vehicle in the vertical plane controlled by the buoyancy engine, the steering control only acts on the direction of movement of the glider in the lateral plane. The suboptimal sliding mode steering law for the glider is:

$$u(t) = -r_1 \text{sign}\left(\sigma - \frac{\sigma^*}{2}\right) + r_2 \text{sign}(\sigma^*) \ , \tag{4.5.10}$$

where σ^* is the value of the sliding variable at the last time instant when the condition $\dot{\sigma} = 0$ was verified and r_1 and r_2 are the strictly positive controller gains. Identically to (3.3.16), the following constraints on the controller's design parameters are introduced:

$$\left\{ r_1 > r_2 > 0 \in \mathbb{R}^+ : \begin{array}{rcl} (r_1 - r_2) & > & \dfrac{C}{K_m}, \\ (r_1 + r_2) & > & \max\left\{\dfrac{4C + K_M(r_1 - r_2)}{3K_m}, \dfrac{V}{R_{min}}\right\} \end{array} \right\} \tag{4.5.11}$$

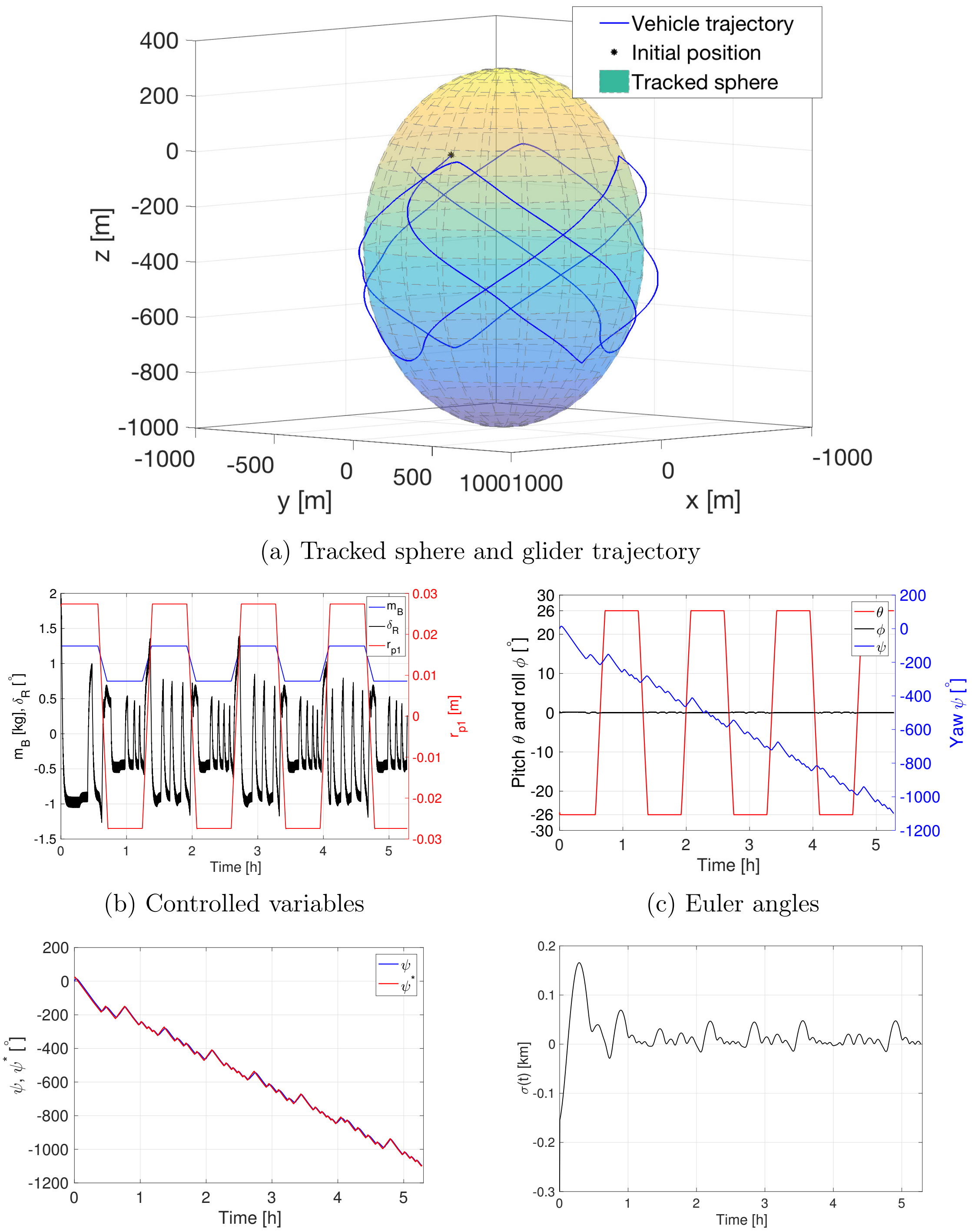
The desired heading of the vehicle $\psi^*(t)$ is obtained through an Euler integration step and it is passed to the rudder controller in (4.3.13), which determines the desired rudder deflection in order to obtain perfect tracking.

4.6 Three-dimensional boundary tracking - a numerical example

The boundary of a three-dimensional feature is assumed to have a spherical shape, the centre of which is located at $(x_c, y_c, z_c) = (0, 0, -350)$ m, with a radius of $R = 650$ m. It is assumed that the vehicle is capable of measuring its distance from the centre of the sphere, (x_c, y_c, z_c), through a range measurement with respect to a fixed reference point. The sliding variable is obtained through:

$$\sigma(t) = r(t) - R \tag{4.6.1}$$

where $r(t) = \sqrt{(x - x_c)^2 + (y - y_c)^2 + (z - z_c)^2}$ is the measurement at the vehicle's position.



(a) Tracked sphere and glider trajectory

(b) Controlled variables

(c) Euler angles

(d) Heading

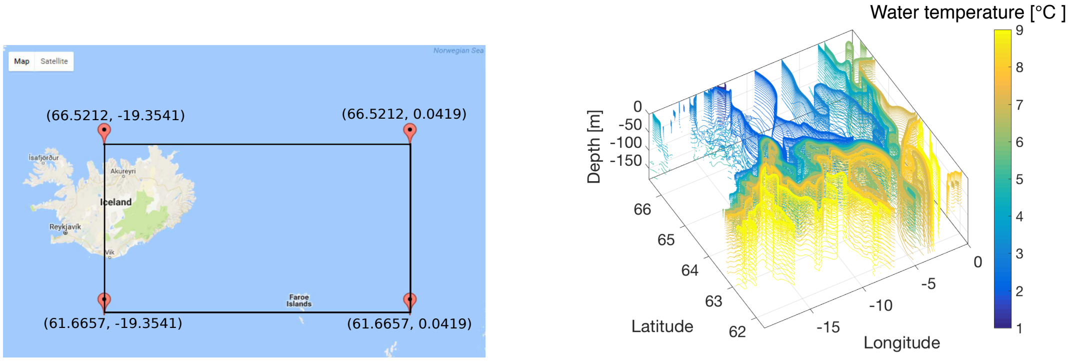
(e) Tracking error

Figure 4.6.1: Sphere tracking simulation results

The suboptimal steering law in (4.5.10), with $r_1 = 10.1$, $r_2 = 1.1$, is used to update the desired heading ψ^* of the vehicle every 10 seconds. With this choice of the controller gains, the bounded uncertainty terms in (3.3.8) can be addressed. The initial conditions for the simulation are taken from the solutions of the trim algorithm presented in Section 4.4.1. The initial Euler angles are: $\theta = -26°, \psi = 0°, \phi = 0°$.

Additionally, the rudder deflection is set initially to $\delta_R = 0°$, the buoyancy mass is exploited fully, setting $m_B = 1.25$ kg, and the glider speed is constrained to $\bar{V} = \sqrt{v_1^2 + v_2^2 + v_3^2} = 0.4$ m/s. With these constraints, the initial position of the movable mass $\bar{m}$, $r_{p1} = 0.0267$ m, and specific values of v and Ω are obtained. The glider is initially deployed on the surface, at position $(x(0), y(0), z(0)) = (350, 0, 0)$. Additionally, the glider is constrained to operate between $z_{min} = 50$ m and $z_{max} = 700$ m, while the desired pitch is set to $\theta^* = \pm 26°$. The actuators proportional gains in (4.3.12) and (4.3.13) are set to $K_P = 0.05$, and $K_{P,\delta_R} = 0.1$ respectively [117], [107].

The obtained results are shown in Fig. 4.6.1. Fig. 4.6.1a shows the tracked sphere and the trajectory of the glider. The initial position of the glider, internal with respect to the tracked sphere, is shown as a black star. The glider follows a conventional sawtooth trajectory in the vertical plane, but its direction of movement in the lateral plane is updated in order to track the spherical boundary. The controlled variables are shown in Fig. 4.6.1b. The ballast mass m_B, shown in blue, can be seen to vary between 1 ± 0.25 kg. In particular, when the ballast mass is at its maximum, the glider sinks with a negative pitch angle and vice versa. The position of the movable mass $\bar{m}$ (r_{p1}) is shown in red. When the ballast mass is maximum, the position of the movable mass is also at its maximum, in order to obtain the desired negative pitch. Finally, the rudder deflection is shown in black in Fig. 4.6.1b. The Euler angles are shown in Fig. 4.6.1c: the vehicle's pitch θ, shown in red, oscillates between the desired values of ±26° depending on the direction of movement in the vertical plane. Additionally, the heading ψ is shown in blue and the roll angle is shown in black. The evolution of the desired heading $\psi^*(t)$, obtained from integration of the steering law in (4.5.10), and the actual heading of the glider $\psi(t)$ are shown in Fig. 4.6.1d for comparison. The rudder control successfully modifies the rudder deflection δ_R, in order to maintain the desired direction of movement over time, that is the desired vehicle's heading. Finally, the sliding variable, obtained as in (4.6.1), is shown in Fig. 4.6.1e. This is initially different from zero, as the vehicle is deployed inside the tracked sphere. Despite the slow control update frequency



(a) Location

(b) Three-dimensional visualization

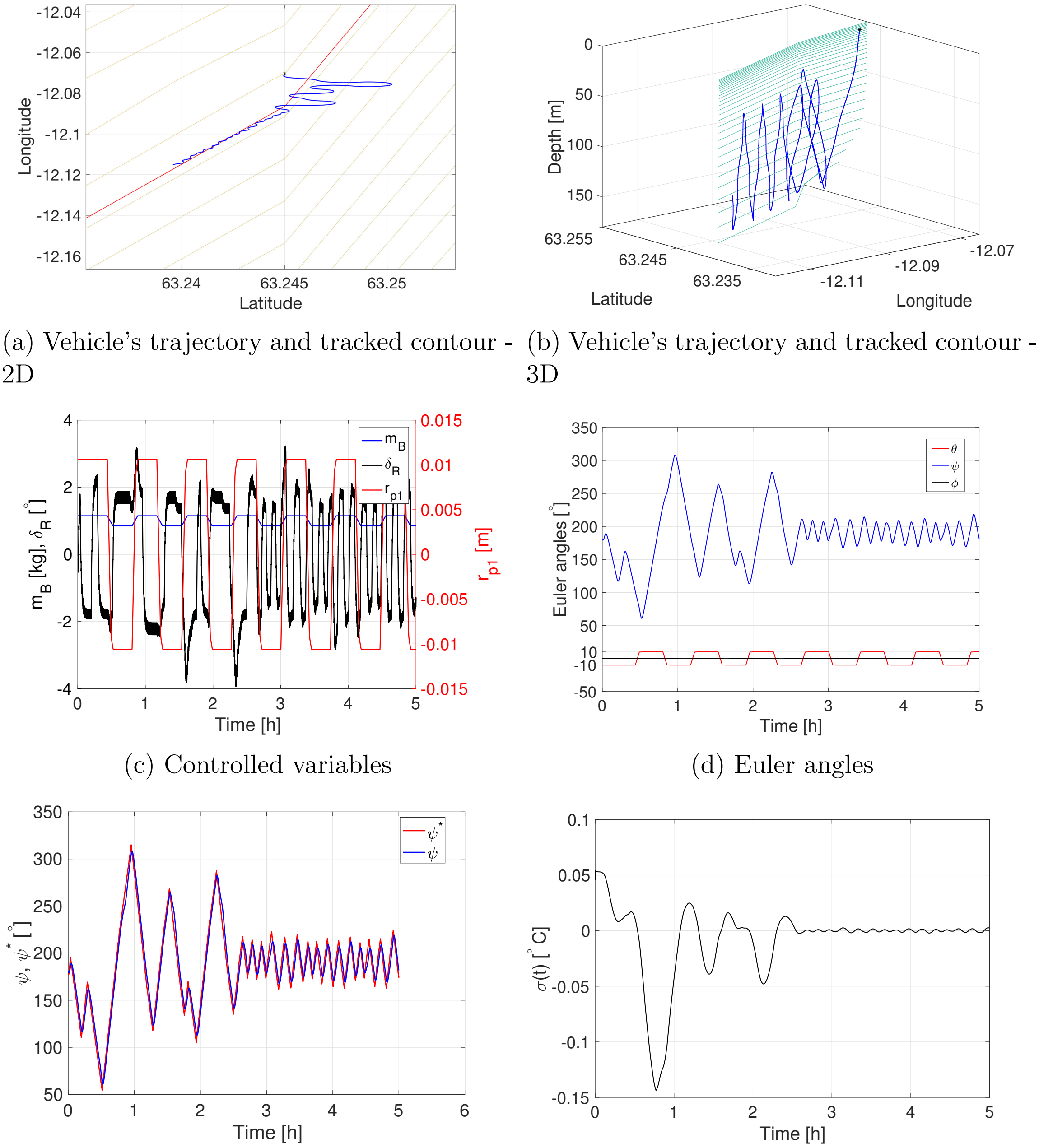
Figure 4.6.2: NEMO-ORCA 12 sea temperature in the Iceland Faroes Front area

and the slow movement and manoeuvre capabilities of the glider, the tracking error is brought near zero in finite time. The sliding variable then oscillates around zero for the entire length of the simulation, successfully solving the boundary tracking control objective.

4.7 Data based validation

The available dataset, obtained from the Met Office NEMO-ORCA 12 model [118], [119], is three-dimensional and it consists of water temperature observations in the area of the Iceland Faroes Front. The area between Iceland and the Faroes Island has for long been the subject of oceanographic studies focussed on the Meridional Overturning Circulation [120]. The available dataset covers the area shown in Fig. 4.6.2a, with a resolution of $1/12°$, for water depths up to 5 km. The water temperature in the area varies in the range $[1°C - 9°C]$ and it is characterized by a well defined temperature front. The clear separation between the warm and cold waters is visible in Fig. 4.6.2b, where the water temperature three-dimensional contours are shown.

The results of a 5 hours long simulation are shown in Fig. 4.7.1. The glider is requested to track the three-dimensional boundary characterized by a value of the water temperature belonging to the central part of the temperature front, specifically $\gamma^* = 7.5°C$. The initial conditions of the simulation are obtained from the trim algorithm in Section 4.4.1, constrained choosing a desired pitch angle of $\theta^* = \pm 10°$.



(a) Vehicle's trajectory and tracked contour - 2D

(b) Vehicle's trajectory and tracked contour - 3D

(c) Controlled variables

(d) Euler angles

(e) Desired and measured heading

(f) Tracking error

Figure 4.7.1: Temperature front tracking simulation results

Additionally, the glider is assumed to move at flight level, with $\phi = 0^\circ$ and with an initial heading $\psi = 180^\circ$. The rudder deflection is assumed to be initially null $\delta_R = 0^\circ$ and the total glider speed is constrained as $\bar{V} = \sqrt{v_1^2 + v_2^2 + v_3^2} = 0.4$ m/s. The solution of the trim algorithm gives an initial ballast mass of $m_B = 1.150$ kg, a movable mass initial position of $r_{p1} = 0.0106$ m and specific values for v and Ω. The glider is initially deployed on the surface, with longitude, latitude coordinates $(x(0), y(0)) = (-12.0737, 63.2450)$. Because of the topography of the region, the movement of the glider is constrained in this case to a layer of water closer to the surface. This has been done in order to limit the risk of collision with the sea-floor.The topography of the region, in fact, shows significant features in depths higher than 150 m. For this reason, the allowed depths of operation are $z \in [z_{min}, z_{max}] = [50, 150]$ m. This is similar to what has been done in [100], where the maximum operational depth of a Slocum glider is fixed to 100 m, and in [97], where the maximum depth is fixed to 200 m. The steering control gains in (4.5.10) are chosen as $r_1 = 10.1$, $r_2 = 1.1$. The movable mass proportional control gain in (4.3.12) is chosen as $K_p = 0.05$, while the rudder proportional gain in (4.3.13) is chosen as $K_{P\delta_R} = 0.2$.

The results obtained are shown in Fig. 4.7.1. Fig. 4.7.1a shows the vehicle trajectory projected onto the two-dimensional space and the contour plot of the sea temperature at the surface level. The tracked contour, characterized by $\gamma^* = 7.5°C$ is highlighted in red, the vehicle's trajectory is shown in blue and the vehicle's initial position is highlighted as a red star. The glider successfully detects the tracked contour, obtaining a sliding motion. The vehicle's trajectory in the three-dimensional space is shown in Fig. 4.7.1b, where the contours characterized by $\gamma^* = 7.5°C$ at different depths are shown in green. The glider's trajectory in the vertical plane follows the conventional sawtooth pattern, while its direction of movement in the lateral plane is varied in order to track the desired level set. The controlled variables are shown in Fig. 4.7.1c: the movable mass position r_{p1} is shown in red, the value of the buoyancy mass m_0 is shown in blue, and the deflection of the rudder is shown in black. The Euler angles are shown in Fig. 4.7.1d. The vehicle's pitch θ, shown in red, alternates between $\pm 10°$, as desired, while the vehicle's roll angle ϕ remains null for the entire simulation. Fig. 4.7.1e shows the evolution over time of the desired glider heading $\psi^*(t)$, defined through the procedure proposed in Section 4.5, and the actual heading of the vehicle. Finally, Fig. 4.7.1f shows the tracking error $\sigma(t)$. Despite being initially of the order of $0.05°C$, the tracking error is reduced to zero in finite time, but it shows a series of oscillations, up to approximately 2.5 hours. These oscillations are due to the slow dynamics of the glider in the lateral plane, resulting from the bounds on the rudder angular velocity. Despite these limitations, the glider successfully moves along the tracked level set, performing several crossings.

4.8 Concluding remarks

A three-dimensional boundary tracking guidance strategy for an autonomous underwater glider is proposed in this chapter. The approach builds on the two-dimensional suboptimal sliding mode boundary tracking algorithm presented in Chapter 3. The proposed boundary tracking algorithm controls the movement of the glider in the lateral plane, while the movement in the vertical plane is assumed limited within a depth range and performed through traditional glider control techniques. The proposed guidance strategy, based on the suboptimal sliding mode control law, is built using a simple kinematic model of the movement of the underwater glider in the three-dimensional space. The designed control action is a steering law, which is then converted in a desired direction of movement for the glider.

The approach is tested in a set of simulations, in which a complete model of the underwater glider, accounting for both its kinematics and its dynamics, is used. The simulations are performed firstly on a numerical example, and secondly on a realistic dataset obtained from the Met Office and relative to the water temperature in the Iceland Faroes Front area. The proposed approach successfully drives the underwater glider along the boundary of an unknown three-dimensional oceanic feature. One peculiarity of the proposed approach is its independence of the path planning techniques, used for the design of a pre-defined trajectory of waypoints, which are typical of the operation of underwater gliders. Additionally, the underwater glider does not require the knowledge nor the estimation of its position while underwater, as the proposed boundary tracking approach relies solely on the instantaneous measurements of the spatial field along the glider's trajectory. Consequently, this approach renders the operation of underwater gliders independent of path planning and dead reckoning techniques, which reduce their autonomy level and may affect the efficacy of their deployment.

Chapter 5

Source seeking with a single autonomous vehicle

5.1 Introduction

A review of the literature on source seeking methods using a single autonomous vehicle is given in Section 2.3.1. In this chapter, a gradient-free source seeking algorithm using a single ASV is presented, which makes use of instantaneous local measurements of the spatial field at the vehicle's position. The approach is based on the suboptimal sliding mode control and exploits ideas from sliding mode based extremum seeking control. The proposed source seeking approach is shown to successfully tackle both static and dynamic spatial fields. Explicit conditions ensuring the finite time convergence to a neighbourhood of the sought source are also introduced. Additionally, the proposed approach is used to relax one of the hypothesis of the suboptimal sliding mode boundary tracking approach presented in Chapter 3. Particularly, the hypothesis on the initial position of the vehicle relative to the tracked contour is relaxed through the introduction of an initial boundary seeking strategy.

The chapter is organized as follows: the characteristics of the considered two-dimensional spatial field are presented in Section 5.2. An initial version of the source seeking approach is presented in Section 5.2 and validated on a numerical example in Section 5.3. A modified source seeking approach, which further develops the approach proposed in Section 5.2, is presented in Section 5.4 and sufficient conditions which guarantee the finite time convergence to a neighbourhood of the sought source are given. The approach is also shown useful in boundary tracking applications in order to allow the vehicle to seek the tracked boundary when initially deployed outside a vicinity of the boundary itself. The local nature of the proposed approach is described in Section 5.5. The approach is extended to the study of dynamic fields in Section 5.6 and its effectiveness is shown through a set of numerical simulations. Finally, the results obtained applying the proposed approach to a realistic dataset relating to a tracer release scenario are presented in Section 5.7, and some concluding remarks are given in Section 5.8.

5.2 Source seeking problem definition

It is assumed that the two-dimensional spatial field $\gamma(x, y)$, considered as in Section 3.2.1 over a compact two-dimensional region $\mathcal{D} \subset \mathbb{R}^2$, has an isolated local maximum defined as:

$$\gamma^s(x^s, y^s) := \max_{(x,y)\in\mathcal{D}} \gamma(x, y) \tag{5.2.1}$$

where neither the location of the maximum $(x^s, y^s) \in \mathcal{D}$ nor the value $\gamma^s(\cdot)$ are known. It is assumed that the spatial field at the points closer to the isolated local maximum has higher intensity, whereas it has lower intensity at the points on the boundary of the spatial phenomenon. Additionally, it is assumed that the spatial field $\gamma(\cdot)$ is bounded $\gamma_{min} \leq \bar{\gamma}(x, y) \leq \gamma^s$, where $\gamma_{min} := \min_{(x,y)\in\mathcal{D}} \gamma(x, y)$.

The objective of the source seeking algorithm is to design a suitable steering law so that the vehicle, modelled as in (3.2.3), climbs the gradient of the spatial phenomenon and reaches a neighbourhood of the isolated local maximum in (5.2.1), defined as:

$$\mathcal{N} = \{(x, y) \in \mathcal{D} : ||(x, y) - (x^s, y^s)|| \leq \epsilon\} \tag{5.2.2}$$

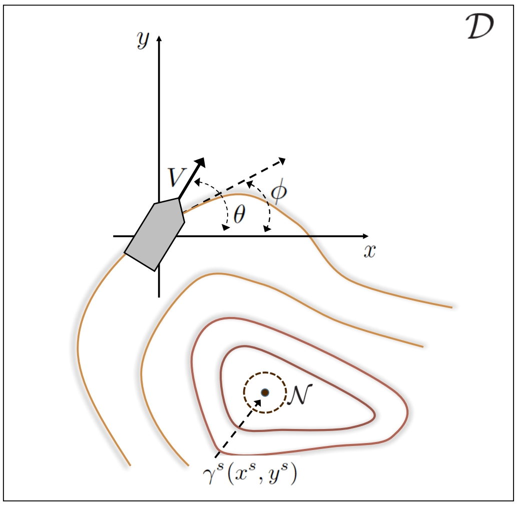


Figure 5.2.1: Spatial field source and neighbourhood characterization

where (x, y) and (x^s, y^s) are the positions of the vehicle and the source respectively. The positive constant ϵ defines the radius of the neighbourhood within which the vehicle should remain since the neighbourhood is firstly entered. The radius of the neighbourhood ϵ is a function of the speed of the vehicle V and the nonholonomic constraint, that is the minimum turning radius of the vehicle. If t_s is the time when the vehicle first enters the neighbourhood of the source, the vehicle is required to remain inside the neighbourhood $\forall t \geq t_s$. A schematic characterization of the spatial field and of the neighbourhood of the source is given in Fig. 5.2.1.

5.2.1 Control algorithm

A review of the extremum seeking approaches based on sliding mode control principles has been given in Section 2.3.1. The main idea of the source seeking algorithm proposed in this chapter is to exploit a sliding motion in order to force the states of the system to follow a reference trajectory, deliberately defined in order to make the vehicle move towards the sought extremum. The reference trajectory describes the desired evolution of the measurements collected by the vehicle, from the measured value of the spatial field at the initial position of the vehicle, to values of the field in the neighbourhood of the source (typically a local maximum). This reference trajectory needs to be designed.

Different designs for the reference trajectory have been proposed in the literature. In [121], periodic search signals, specifically sinusoidal or sawtooth type signals, have been considered. As highlighted in Section 2.3.1, the use of this sort of reference trajectories may cause the vehicle to perform superfluous and expensive manoeuvres. In [65], a strictly monotonically increasing reference signal is considered in order to reach the extremum of a one-dimensional function. A strictly monotonically increasing reference signal is considered also in [67], where environmental extremum seeking is performed with a single autonomous vehicle.

The sliding variable $\sigma(t)$ is defined here as:

$$\sigma(t) := \gamma(x(t), y(t)) - \gamma_{ref}(t) \tag{5.2.3}$$

where $\gamma(x(t), y(t))$ is the instantaneous point measurement taken at position $(x(t), y(t))$ at time t by the vehicle, while traversing a trajectory in the region $\mathcal{D}$. Differently from the previous definitions of $\sigma(t)$ in (3.2.6) and (4.5.3), here $\gamma_{ref}(t)$ is a time-varying reference trajectory. The sliding surface is defined as:

$$\mathcal{S} := \{(x(t), y(t)) \in \mathcal{D} : \sigma(t) = 0\} \tag{5.2.4}$$

It is evident from (5.2.3) and (5.2.4) that, if sliding occurs, then the vehicle traverses a trajectory along which the point measurements precisely match $\gamma_{ref}(t)$.

The reference trajectory $\gamma_{ref}(t)$ is defined here as a non-decreasing function. The initial value of $\gamma_{ref}(t)$ coincides with the value of the spatial field at the initial location of the vehicle, $\gamma_{ref}(0) = \gamma(x(0), y(0))$. The reference trajectory is defined as a function of the sliding variable. Particularly, for reaching the neighbourhood of the source $\mathcal{N}$ defined as in (5.2.2), the dynamics of the reference trajectory is defined as:

$$\dot{\gamma}_{ref}(t) = \begin{cases} K & \text{if } |\sigma(t)| < \delta \\ 0 & \text{otherwise} \end{cases} \tag{5.2.5}$$

where the constant slope $K > 0$ is a design parameter to be selected and δ is a small positive design scalar. The scalar δ is used to determine the occurrence of sliding, and hence the parameter defines an accepted level of tracking error. According to (5.2.5), $\gamma_{ref}(t)$ is a non-decreasing function, strictly increasing when sliding occurs, and constant when sliding is lost. Inside the neighbourhood of the source defined as in (5.2.2), by design, sliding is lost and never regained. In fact, sliding is definitely lost when the value of the reference trajectory verifies $\gamma_{ref}(t) > \gamma^s(x^s, y^s)$. When this occurs, the control action $u(t)$ remains constant and the vehicle starts circling on a circumference of radius $R = V/|u(t)|$. The controller gains determine the magnitude of $u(t)$, which effectively determines the capability of the vehicle to perform sharp turns of radius $R > R_{min}$. For a vehicle modelled as in (3.2.3), moving along a circumference of radius R, the following relation holds:

$$V = u(t)R \tag{5.2.6}$$

Consequently, the minimum radius of curvature the vehicle trajectory can undertake results:

$$R_{min} = \frac{V}{u_{max}} \tag{5.2.7}$$

where u_{max} is the maximum value of the angular velocity, which is achieved for a maximum value of the steering control $u(t)$.

As derived in Section 3.3, the system between the sliding variable in (5.2.3) and the control action $u(t)$ has a relative degree two. This can be verified by taking subsequent derivatives of (5.2.3) along the vehicle's trajectory, following the steps in Section 3.3. The second derivative of the sliding variable along the vehicle's trajectory, considering that $\ddot{\gamma}_{ref}(t) = 0$, results:

$$\ddot{\sigma}(t) = \underbrace{V \sin(\theta - \phi)\frac{d||\nabla\gamma||}{dt}}_{\xi(t)} + \underbrace{||\nabla\gamma||V\cos(\theta - \phi)}_{b(t)}(\dot{\theta} - \dot{\phi}) \tag{5.2.8}$$

The expression of $\ddot{\sigma}(t)$ in (5.2.8), has the same structure of the system considered

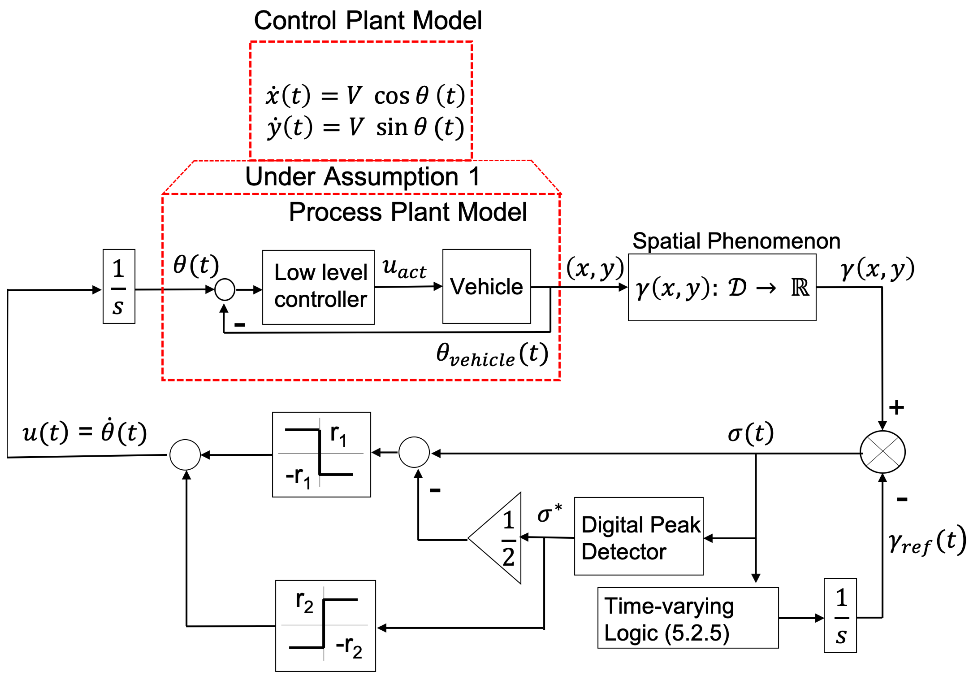
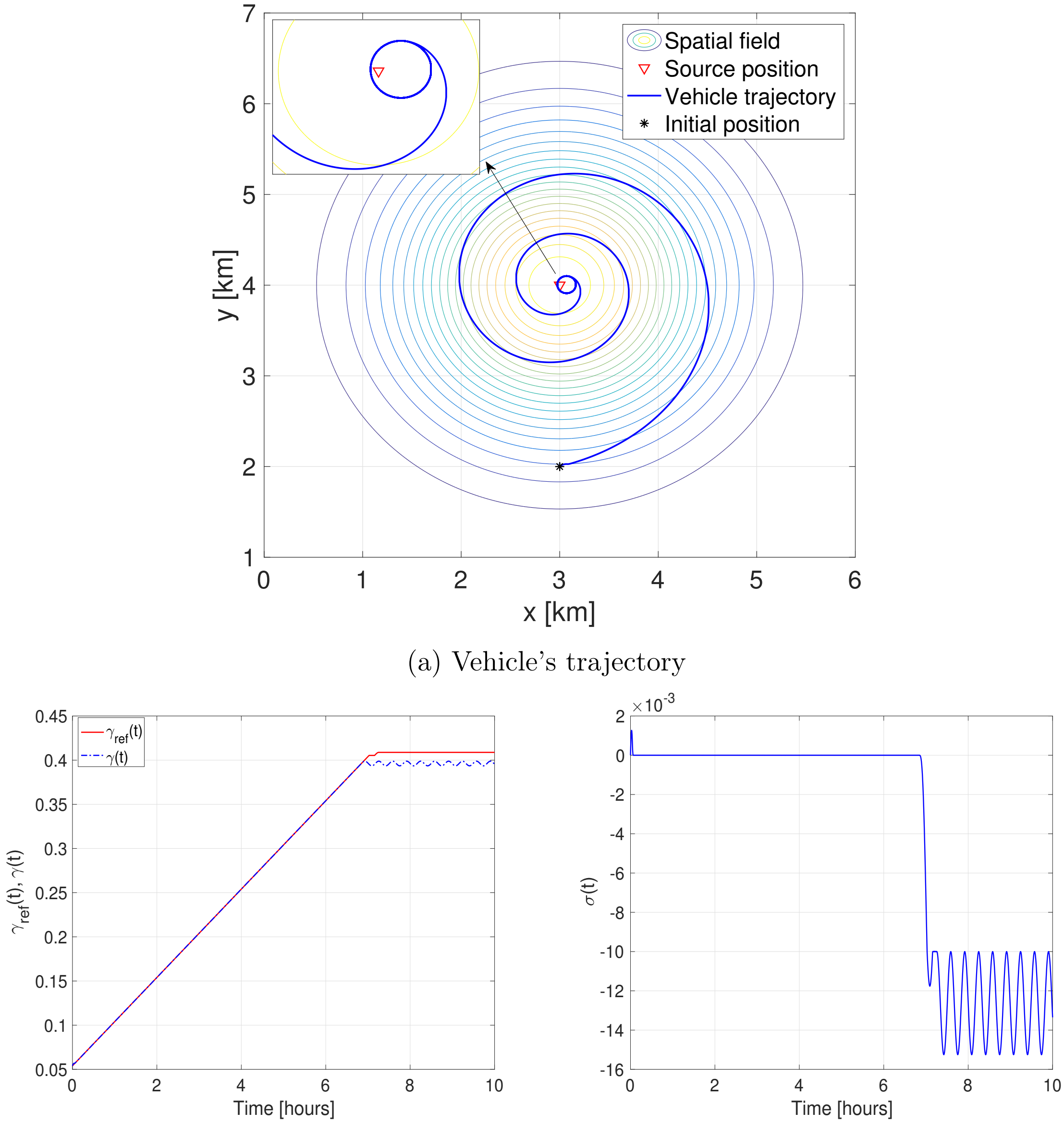


Figure 5.2.2: Schematic of the proposed source seeking approach

in Section 3.3, and hence the relative degree is two. Consequently, the suboptimal sliding mode steering law in (3.3.10) can be used to seek the source of the considered spatial field. A schematic of the proposed source seeking approach is shown in Fig. 5.2.2. With this approach, if the controller gains are chosen according to the constraints in (3.3.16), the minimum turning radius of the vehicle results as in (5.2.7), with $u_{max} = |r_1| + |r_2|$.

5.3 Approach validation on a numerical example

The numerical simulations are carried out using Matlab/Simulink (version R2016b). A fixed step Euler integration method, with a step size of 0.1 s, is used. A Gaussian distribution, represented by (3.4.1) where the parameters σ_G, μ_x and μ_y are set to 1, 3 and 4 respectively, is considered and it is shown in Fig. 5.3.1. The vehicle is modelled as in (3.2.3), with velocity 0.5 m/s $\approx$ 1.8 km/h. The angular velocity is controlled using the suboptimal sliding mode approach in (3.3.10), where the control gains are $r_1 = 20$ and $r_2 = 1$. With these choices of the parameters, the minimum turning radius of the autonomous vehicle is limited to approximately $R_{min} \approx 90$ m. The initial location of the vehicle is $(x(0), y(0)) = (3, 2)$, with $\theta(0) = \pi/4$, and it is



(a) Vehicle's trajectory

(b) Vehicle's measurements and reference trajectory

(c) Tracking error

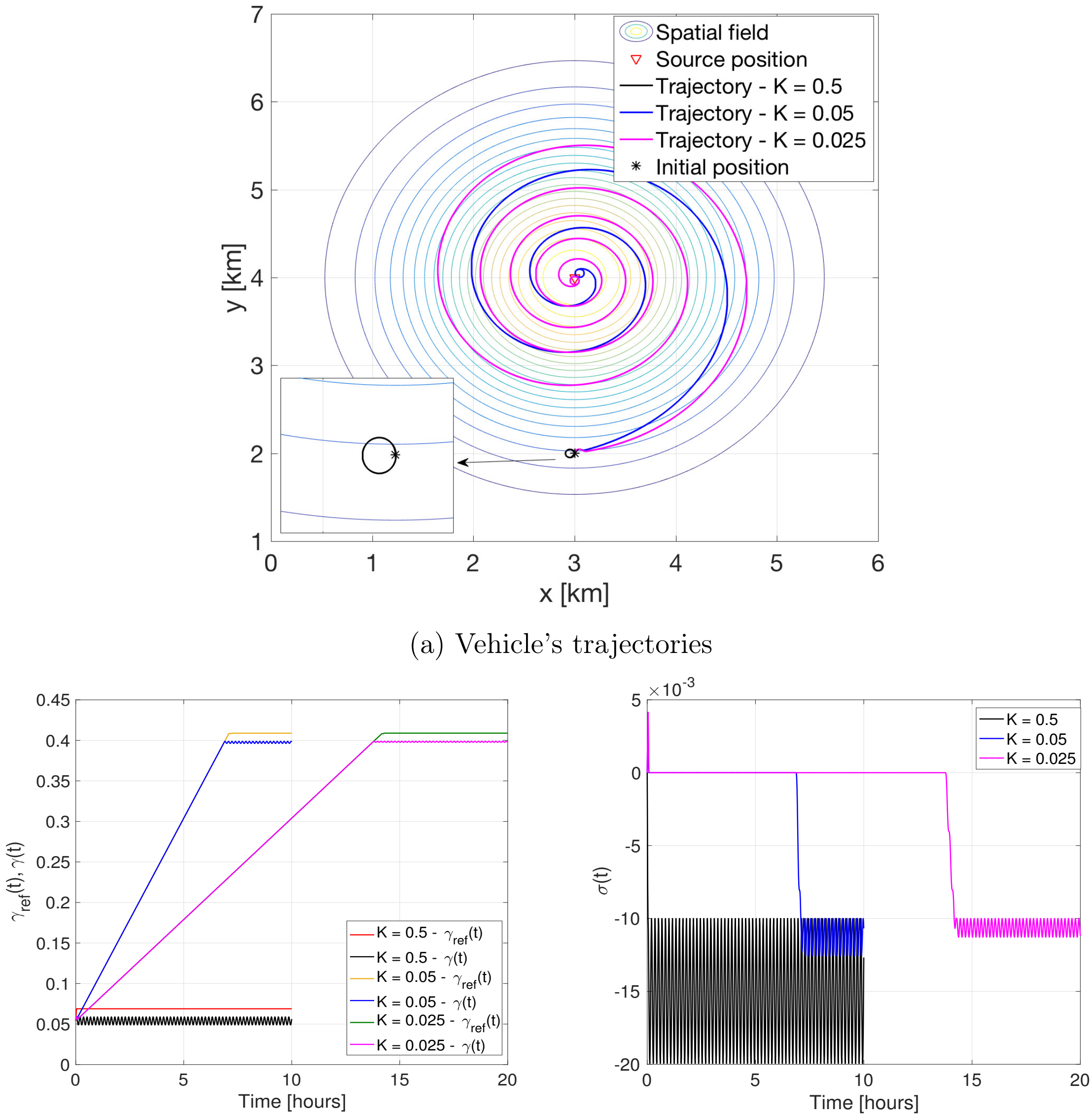
Figure 5.3.1: Numerical source seeking results

indicated with a black star in Fig. 5.3.1a.

The parameters defining the reference trajectory in (5.2.5) are set to $K = 0.05$ and $\delta = 0.01$. The choice of δ fixes the accepted tracking error to ± 0.01, while the chosen slope K determines a relatively slow varying reference trajectory. In Fig. 5.3.1a, the position of the source (x^s, y^s), highlighted with a red triangle, and the value of the spatial field at the source $\gamma^s(\cdot)$ are assumed to be unknown to the vehicle. The trajectory of the vehicle is shown in blue and the initial position of the vehicle is highlighted with a black star. The vehicle successfully moves from its initial position towards a neighbourhood of the source. As highlighted in the enlargement in Fig. 5.3.1a, the vehicle starts circling around the source, on a circumference having a radius of approximately 90 m, as soon as the source's neighbourhood is entered. Fig. 5.3.1b shows the reference trajectory, designed as in (5.2.5), in red and the time evolution of the vehicle's measurements in blue. The measurements along the vehicle's trajectory follow the reference trajectory accurately until the neighbourhood of the source is entered, after approximately 7 hours and since then sliding is lost. When the vehicle moves inside the neighbourhood, the reference trajectory remains constant and the measurements at the vehicle's position oscillate because of the circling behaviour of the vehicle, whose steering control $u(t)$ remains constant. The tracking error, that is the sliding variable in (5.2.3), is shown in Fig. 5.3.1c. The tracking error is maintained approximately zero until the neighbourhood of the source is entered, showing how sliding is successfully maintained during the movement towards the neighbourhood of the source.

The results obtained from an additional set of simulations are shown in Fig. 5.3.2, where the control parameters in (3.3.10) are modified to $r_1 = 40$, $r_2 = 1$, in order to reduce R_{min} to approximately 45 m. Choosing the parameter K in (5.2.5) as $K = 0.05$ or $K = 0.025$, the vehicle eventually reaches a neighbourhood of the source. These results, however, highlight one limitation of the approach introduced in Section 5.2. This is the time taken by the vehicle in order to enter the neighbourhood of the source, which is due to the non optimal design of the parameter K. A small value of K, in fact, can make the vehicle perform a spiral motion and a consequent slow movement towards the source.

The main limitation of the approach introduced in Section 5.2 is the absence of an explicit condition which guarantees, for a certain choice of the parameter K in (5.2.5), the optimal convergence to a neighbourhood of the source. As visible in the enlargement in Fig. 5.3.2a, if $K = 0.5$ is chosen, the vehicle indefinitely circles on a circumference of radius R_{min} from its initial position and fails to reach the



(a) Vehicle's trajectories

(b) Vehicle's measurements and reference trajectories

(c) Tracking errors

Figure 5.3.2: Numerical source seeking results - different designs of $\gamma_{ref}(t)$

neighbourhood of the sought source. When a high value of the reference trajectory's slope K is chosen, the reference value is rapidly increased initially and it can happen that the vehicle is unable to find points where the constraint in (5.2.4) is verified. This is confirmed in Fig. 5.3.2b, where the reference trajectory $\gamma_{ref}(t)$ in the case $K = 0.5$ is shown in red and the measurements collected by the vehicle for the same choice of K are shown in black. When this happens, sliding is lost initially and never regained, the reference trajectory remains constant and the vehicle continues circling, failing to move towards the sought maximum.

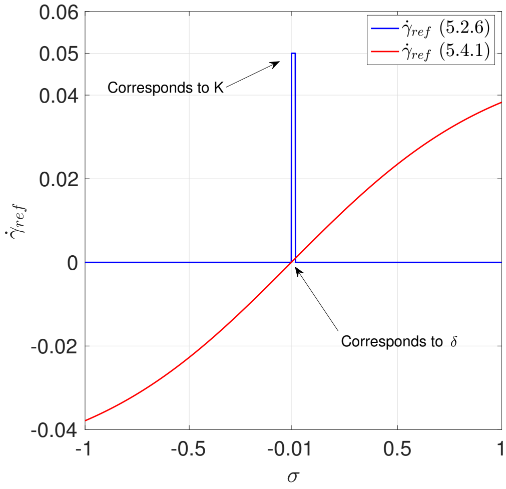


Figure 5.4.1: Reference trajectories (5.2.5) and (5.4.1) comparison

5.4 A modified reference trajectory definition

The results in Section 5.3 highlight the capability of the proposed approach to drive the vehicle inside a neighbourhood of the sought source in finite time. The convergence to the neighbourhood of the source, however, is highly influenced by the choice of the parameter K in (5.2.5), whose tuning is rather intuitive and it requires a few trial and error attempts. The reference value controlled as in (5.2.5) is kept constant whenever $|\sigma(t)| > \delta$. When $\sigma(t) < -\delta$, the measurements taken by the vehicle are smaller than the reference values; in this case, it would be logical to reduce the value of the reference trajectory, in order to ease sliding recovery. Similarly, when $\sigma(t) > \delta$, the measurements taken by the vehicle are higher than the reference trajectory. Instead of keeping $\gamma_{ref}(t)$ constant, it would be desirable to further increase the value of the reference trajectory. In other words, it might be desirable to have $\dot{\gamma}_{ref}(t) > 0$ when $\sigma(t) > \delta$ and vice versa.

In order to improve the generation of the reference trajectory in (5.2.5), the following modified design is introduced:

$$\dot{\gamma}_{ref}(t) = k(t) = K \tanh(\sigma(t) + \delta) \tag{5.4.1}$$

where K is a positive design constant, defining the maximum slope of $\gamma_{ref}(t)$, and

$\gamma_{ref}(0) = \gamma(x(0), y(0))$. The reference trajectory designed through (5.4.1) has the desired characteristics; i.e. it is positive for $(\sigma(t) + \delta) > 0$ and negative otherwise. This means that the reference trajectory is not kept constant whenever $|\sigma(t)| > \delta$, as in (5.2.5), but it is varied according to the sign of $\sigma(t)$. The small drift term δ has been introduced in order to ensure that $\dot{\gamma}_{ref}(t) \neq 0$ when $\sigma(t) = 0$. This parameter, additionally, defines the accepted value of tracking error. A graphical comparison of the reference trajectories obtained through (5.2.5) and (5.4.1) is given in Fig. 5.4.1, where the choices $K = 0.05$ and $\delta = 0.01$ are made.

The control strategy presented in Section 5.2 is still valid with the reference trajectory in (5.4.1). With the choice of $\gamma_{ref}(t)$ in (5.4.1), the first derivative of the sliding variable in (5.2.3) becomes:

$$\dot{\sigma}(t) = ||\nabla\gamma||V \sin(\theta - \phi) - K \tanh(\sigma(t) + \delta) \ , \tag{5.4.2}$$

while the second derivative in (5.2.8) results:

$$\ddot{\sigma}(t) = \underbrace{V \sin(\theta - \phi)\frac{d||\nabla\gamma||}{dt} - K(1 - \tanh^2(\sigma(t) + \delta)\dot{\sigma}(t)}_{\xi(t)} + \underbrace{||\nabla\gamma||V \cos(\theta - \phi)}_{b(t)}(\dot{\theta} - \dot{\phi}) \tag{5.4.3}$$

As a consequence, the only effect of the modified reference trajectory definition on the design approach detailed in Section 5.2 is the requirement of a larger bound for the drift uncertainty $\xi(t)$, to account for the effect of the additional term deriving from the reference trajectory's dynamics.

5.4.1 Sufficient condition for source seeking

With the modified reference trajectory definition introduced in Section 5.4, it is possible to derive a bound on the design parameter K in (5.4.1), to guarantee that the source seeking approach successfully drives the autonomous vehicle to a neighbourhood of the sought source in finite time.

A dynamic spatial field is considered here for the introduction of a sufficient condition. The formulation of the introduced condition in the case of a static two-dimensional spatial field is also given. A dynamic two-dimensional spatial field is defined as:

$$\gamma(x,y,t) : \mathcal{D} \times \mathbb{R}^+ \to \mathbb{R} \tag{5.4.4}$$

The spatial mapping $\gamma(x,y,t)$ associates a numerical value of the spatial field at each location $(x,y) \in \mathcal{D}$ at each time t.

The underlying concept for the introduction of the sufficient condition is as follows. It is assumed that the speed of the vehicle is higher than the maximum speed of the reference contour, which is identified by the value of the reference trajectory $\gamma_{ref}(t)$ at a specific instant t.

The velocity of the reference contour, as previously derived in [122] through the implicit function theorem, is:

$$v_{\Gamma_{ref}}(x,y,t) = -\frac{\gamma'(x,y,t)}{||\nabla\gamma||} + \frac{k(t)}{||\nabla\gamma||} \tag{5.4.5}$$

where $\gamma'(x,y,t) = \partial\gamma(x,y,t)/\partial t$ is the partial derivative of the spatial field with respect to time and $\nabla\gamma$ is the spatial field gradient, defined as in (3.3.5).

If the vehicle's speed is greater than the maximum speed of the reference contour, the vehicle can successfully follow the reference trajectory, while moving towards the neighbourhood of the source. The maximum speed of the reference contour during sliding results, from (5.4.5):

$$v_{\Gamma_{ref}max}(x,y,t) = \frac{|-\partial\gamma(x,y,t)/\partial t|_{max}}{||\nabla\gamma||_{min}} + \frac{K\tanh(\delta)}{||\nabla\gamma||_{min}} \tag{5.4.6}$$

where $K\tanh(\delta)$ is the slope of the reference trajectory when sliding is verified, that is when $\sigma(t) = 0$ in (5.4.1). The minimum spatial field gradient $||\nabla\gamma||_{min}$ is the minimum value in the vehicle region of operation, and it is defined as:

$$||\nabla\gamma||_{min} = \min_{(x,y)\in\mathcal{O}} ||\nabla\gamma|| \tag{5.4.7}$$

The region of operation of the autonomous vehicle $\mathcal{O}$ is the area in which the vehicle moves from its initial position to the desired neighbourhood $\mathcal{N}$ of the sought source while performing source seeking. The requirement on the vehicle's speed therefore translates into:

$$V \geq v_{\Gamma_{ref}max}(x, y, t) \tag{5.4.8}$$

By rearranging (5.4.8) and considering (5.4.6), the following condition on the choice of the control parameter K is obtained:

$$K \leq \frac{||\nabla\gamma||_{min}(V - v_{\Gamma,max}(x, y, t))}{\tanh(\delta)} \tag{5.4.9}$$

where $v_{\Gamma,max}(x, y, t) = \frac{|-\partial\gamma(x,y,t)/\partial t|_{max}}{||\nabla\gamma||_{min}}$. In conclusion, if the reference trajectory design parameter K is chosen according to (5.4.9), the proposed approach successfully drives the vehicle from its initial deployment to a neighbourhood of the sought source.

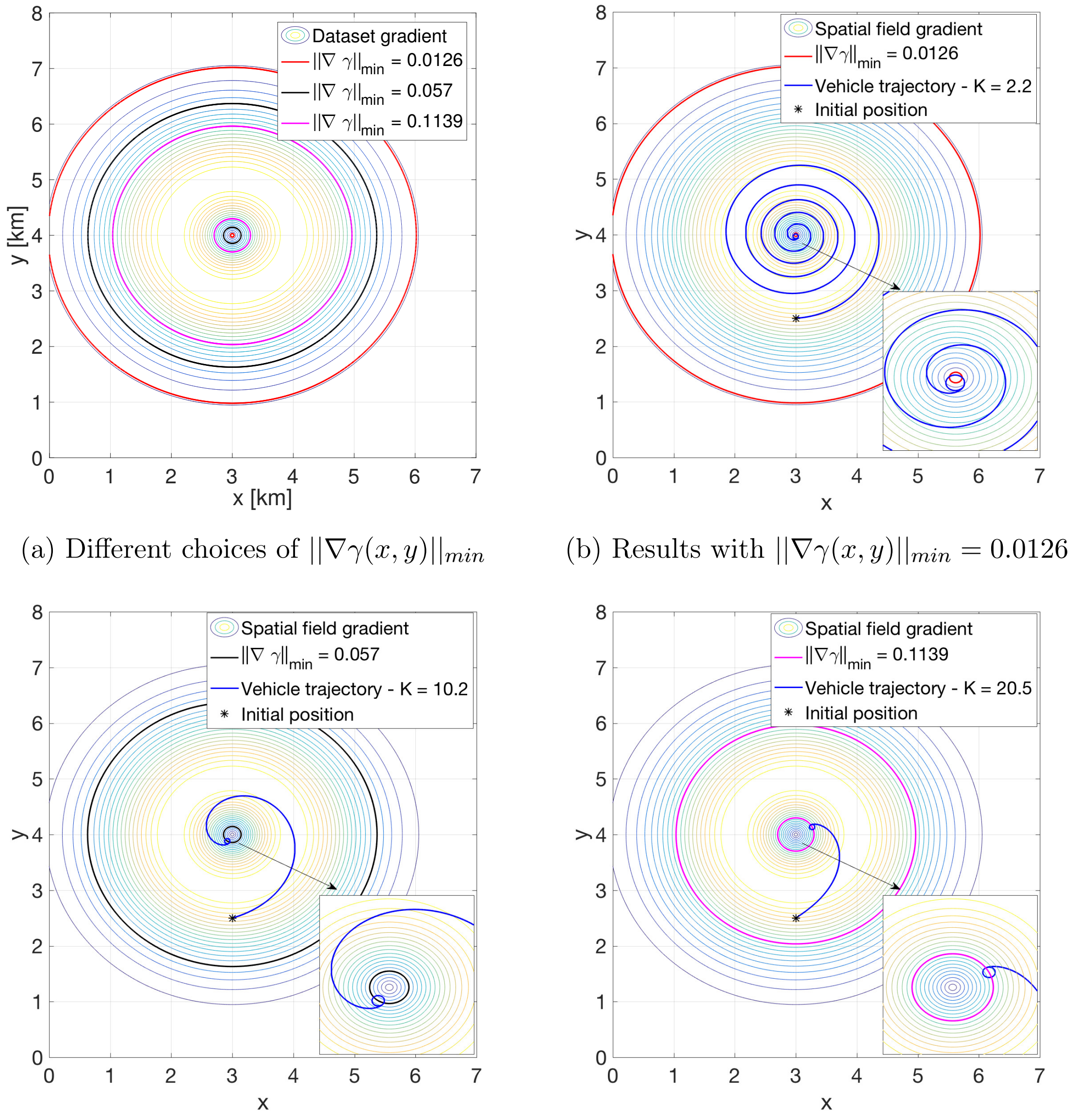
A particular type of dynamic spatial field is the translating field. A translating field is characterized by a spatial distribution which undergoes an isometric transformation. This transformation can be described through the movement of the source, which happens at velocity V_c. When a translating field is considered, the sufficient condition in (5.4.9) results modified in:

$$K \leq \frac{||\nabla\gamma||_{min}(V - V_c)}{\tanh(\delta)} \tag{5.4.10}$$

Finally, if the considered spatial field is static, the sufficient condition in (5.4.9) results simplified as:

$$K \leq \frac{||\nabla\gamma||_{min}V}{\tanh(\delta)} \tag{5.4.11}$$

The validity of this sufficient condition is demonstrated through its application to a numerical example. The considered static spatial phenomenon is the Gaussian distribution previously introduced in Section 5.3. In this numerical example, the spatial field gradient can be evaluated through the availability of the analytical



(a) Different choices of $||\nabla\gamma(x, y)||_{min}$

(b) Results with $||\nabla\gamma(x, y)||_{min} = 0.0126$

(c) Results with $||\nabla\gamma(x, y)||_{min} = 0.057$

(d) Results with $||\nabla\gamma(x, y)||_{min} = 0.1139$

Figure 5.4.2: Numerical source seeking results - validation of the sufficient condition in (5.4.11)

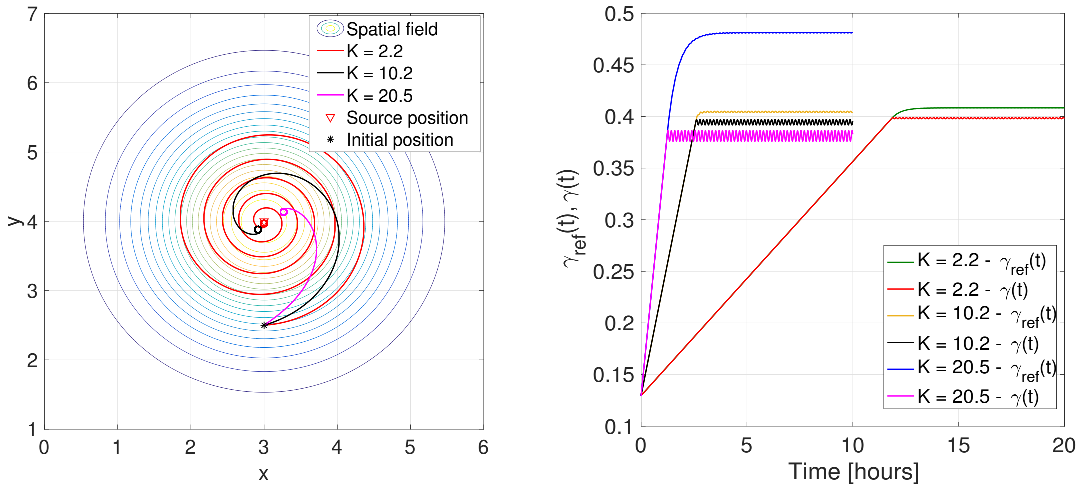
expression of the Gaussian distribution:

$$f(x, y) = \frac{1}{\sqrt{2\pi\sigma_G^2}} e^{-\frac{(x-\mu_x)^2+(y-\mu_y)^2}{2\sigma_G^2}}, \tag{5.4.12}$$

The spatial field gradient is shown in Fig. 5.4.2a, where three different contours, corresponding to different values of $||\nabla\gamma||$ are highlighted. These have been chosen as $||\nabla\gamma||_{min}$ in three different simulations, defining different regions of operation. In each simulation, a vehicle moving with speed $V = 1.8$ km/h is deployed at

$(x(0), y(0)) = (3, 2.5)$, with $\theta(0) = \pi/4$. For the definition of the reference trajectory as in (5.4.1), $\delta = 0.01$ is chosen. The sufficient condition in (5.4.11) is used to determine the parameter K in (5.4.1), which is influenced by the chosen minimum value for the spatial gradient. Specifically, if $||\nabla\gamma||_{min} = 0.0126$, then the choice of K needs to satisfy $K \leq 2.2680$, and $K = 2.2$ is chosen. Applying the same sufficient condition with $||\nabla\gamma||_{min} = 0.0570$, $K = 10.2$ is selected. Finally, selecting $||\nabla\gamma||_{min} = 0.1139$, $K = 20.5$ is chosen. The results obtained in the three simulations are shown in Fig 5.4.2, where the trajectory of the vehicle in each case in shown along with the spatial field gradient contours. In each case, the vehicle successfully moves from its initial position to a neighbourhood of the source. The chosen value of the spatial gradient determines how close the vehicle arrives to the sought source. In each of the three simulations, in fact, it is possible to observe how the vehicle fails to continue moving towards the source as soon as the constant gradient contour, characterized by the chosen value of $||\nabla\gamma||_{min}$, is crossed. When this happens, because of the conformation of the Gaussian distribution, the vehicle enters a region where $||\nabla\gamma|| < ||\nabla\gamma||_{min}$ and therefore the sufficient condition in (5.4.11) is no longer verified. Hence, sliding is lost and never regained and the circling behaviour is obtained. The vehicle's trajectories are also shown in Fig. 5.4.3a, with the spatial field contours. Finally, the obtained reference trajectories and the vehicle's measurements are shown in Fig. 5.4.3b.

The results obtained through this set of simulations demonstrate how the introduced sufficient condition represents an effective strategy for the choice of the control parameter K in (5.4.1). The sufficient condition in (5.4.11) introduces a *trade-off* between the speed of convergence to a neighbourhood of the sought source and the precision in the estimation of the source position. It is possible to observe from Fig. 5.4.3a how increasing the control parameter K shortens the time employed by the vehicle to enter the neighbourhood of the source, but reduces the accuracy in the estimation of the source position itself.



(a) Vehicle's trajectories

(b) Vehicle's measurements and reference trajectories

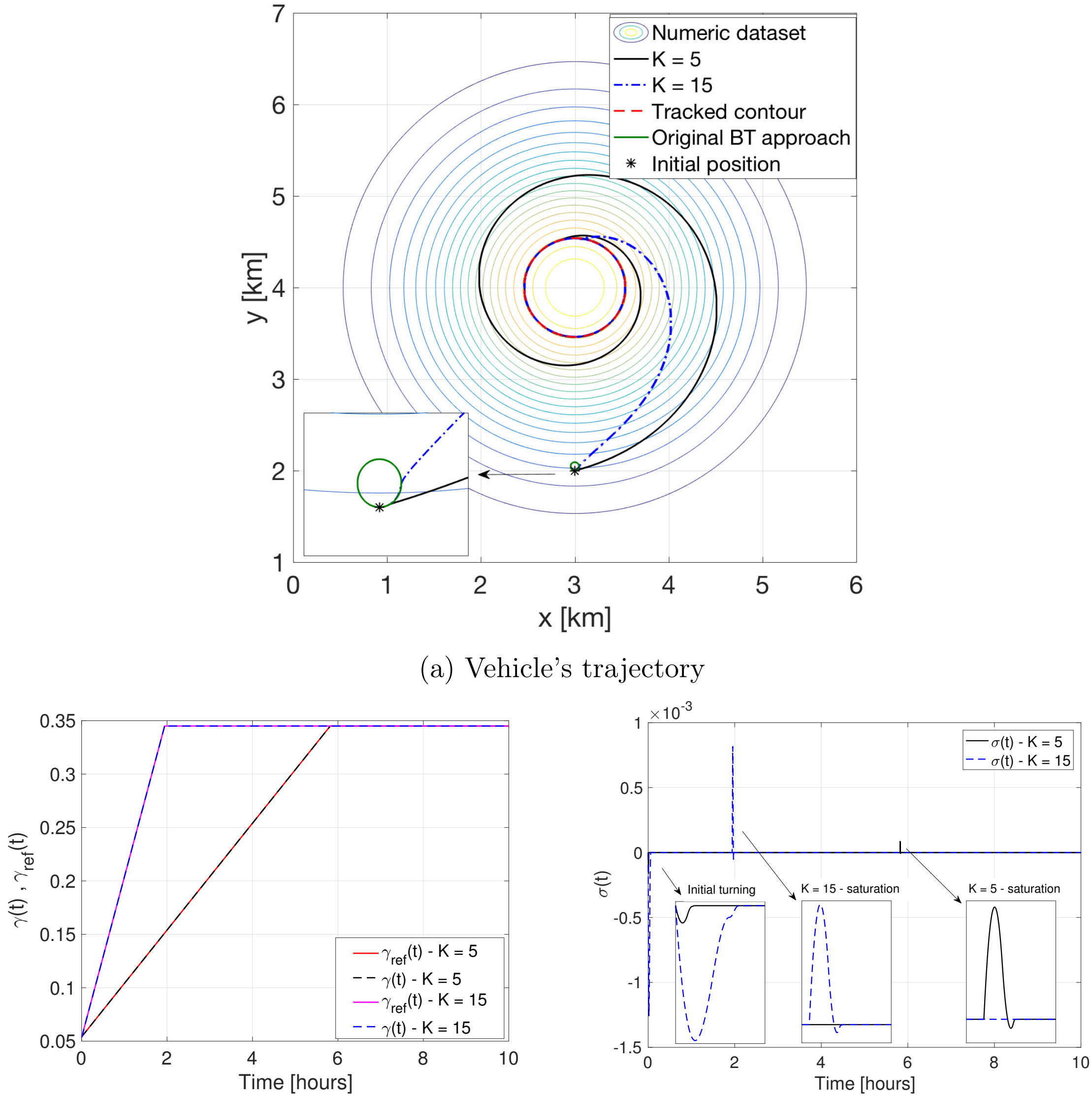
Figure 5.4.3: Numerical source seeking results - different choices of K

5.4.2 Use of source seeking for boundary tracking

The assumption of the boundary tracking strategy originally proposed in [39] and presented in Chapter 3, which required the vehicle to be initially deployed in the vicinity of the tracked contour, is relaxed here. The strategy introduced in Section 5.2 can be used to make the vehicle seek the tracked contour from its initial position, before starting tracking it. A reference trajectory similar to the one defined in (5.4.1), in fact, can be introduced to make the vehicle seek the tracked contour, defined as in (3.2.2) and characterized by $\gamma(x(t), y(t)) = \gamma^*$. Particularly, the reference trajectory is introduced to describe the desired evolution of the measurements collected by the vehicle from its initial measurement $\gamma(x(0), y(0))$ to the value of the spatial field on the tracked contour γ^*. The reference trajectory is defined as:

$$\dot{\gamma}_{ref}(t) = \begin{cases} K \tanh(\sigma(t) + \delta) & \text{if } (\gamma_{ref}(t) - \gamma^*)(\gamma_{ref}(0) - \gamma^*) > 0 \\ 0 & \text{otherwise} \end{cases} \tag{5.4.13}$$

with $\gamma_{ref}(0) = \gamma(x(0), y(0))$. The condition introduced in (5.4.13), which verifies if the tracked value γ^* is reached, has a saturating effect on the reference trajectory. When the first condition in (5.4.13) is no longer verified, the slope of the reference trajectory is set to 0 and a constant value of $\gamma_{ref}(t)$ is obtained, which identifies



(a) Vehicle's trajectory

(b) Vehicle's measurements and reference trajectories

(c) Tracking error

Figure 5.4.4: Numerical boundary tracking results - effectiveness of the initial seeking strategy

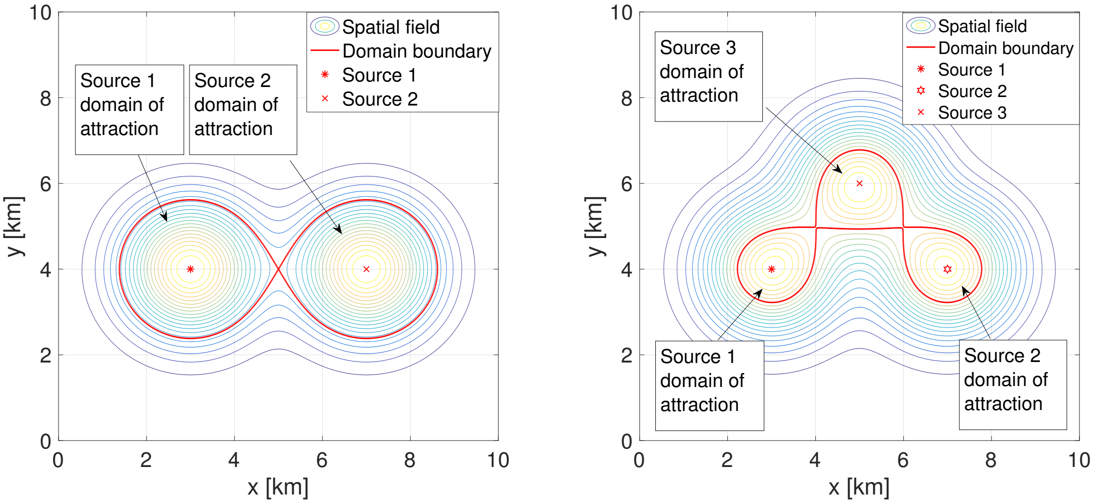
the value of the spatial feature along the tracked contour. In this way, the requirement of the initial position of the vehicle relative to the tracked contour is relaxed. Moreover, the advantages of the source seeking strategy remain valid; particularly, if the design parameter K in (5.4.13) is chosen according to the sufficient condition introduced in Section 5.4.1, then the finite time convergence to the tracked contour can be guaranteed.

The validity of the boundary seeking strategy is demonstrated on the same numeric Gaussian distribution example considered in Section 5.4.1. The vehicle, whose steering control design is maintained as in Section 5.3, is initially deployed at $(x(0), y(0)) = (3, 2)$ with $\theta(0) = 0$, and it is required to seek and track the contour characterized by $\gamma^* = 0.345$. The trajectories in Fig. 5.4.4a, obtained with different choices of the design parameter K in (5.4.13), specifically $K = 5$ and $K = 15$, show how the approach effectively makes the vehicle reach the tracked contour, before starting sliding along it. The different time taken before converging to the tracked contour is determined by the different choices of the design parameter K, whose effect on the slope of the reference trajectory is visible in Fig. 5.4.4b. Increasing K, in fact, the convergence time can be reduced from approximately 6 hours to 2 hours. Finally, the tracking error is shown in Fig. 5.4.4c, where the highlighted spikes are due to the limited turning capability of the vehicle, and correspond to the time instances where the condition in (5.4.13) is no longer verified. The trajectory in the enlargement in Fig. 5.4.4a, where the vehicle fails to seek the tracked boundary and indefinitely circle around its initial position, is obtained applying the original boundary tracking strategy presented in Chapter 3, with $K = 5$, without the seeking strategy.

In conclusion, the reference trajectory generation strategy discussed in this chapter allows not only to effectively perform source seeking, but also to perform boundary tracking independently of the initial deployment of the vehicle relative to the tracked contour.

5.5 Local behaviour

The source seeking algorithm introduced in Section 5.4 is based on the assumption that the spatial field is characterized by a single isolated local maximum. In the presence of multiple local maxima, the proposed algorithm shows a local behaviour. Particularly, the algorithm drives the vehicle from its initial deployment to a neighbourhood of the source whose domain of attraction is first entered. This behaviour is typical of optimization algorithms based on local information, as highlighted in



(a) Two identical sources

(b) Three identical sources

Figure 5.5.1: Domain of attraction of multiple sources

[123]. When considering a field with multiple local maxima, the domain $\mathcal{D}$ can be divided into different regions. Particularly, each local maxima is characterized by a *domain of attraction*, which is an area surrounding the local maximum itself. If the vehicle is initially deployed inside the domain of attraction of a local maximum, then it will be driven by the algorithm introduced in Section 5.4 towards a neighbourhood of that specific local maximum. If the vehicle is initially deployed outside the region of attraction of every source, instead, it will be driven to a neighbourhood of the source whose domain of attraction is first entered.

The boundary of the domain of attraction of each local maxima is characterized by a constant value of the spatial field and it is therefore a constant level set of the considered spatial field. The value of the spatial field on the boundary of the domain of attraction can be determined by studying the evolution of the spatial field gradient along the direction joining two local maxima. The value of the spatial field at the point where the spatial gradient is null ($||\nabla\gamma(x,y)|| = 0$) defines the boundary of the domain of attraction. This is because at that specific point the gradient shifts direction from one peak to the other. Fig. 5.5.1 and Fig. 5.5.2 show some examples of the domains of attraction in the presence of multiple local maxima. In Fig. 5.5.1a, the field is characterized by two identical sources, whose domains of attraction are

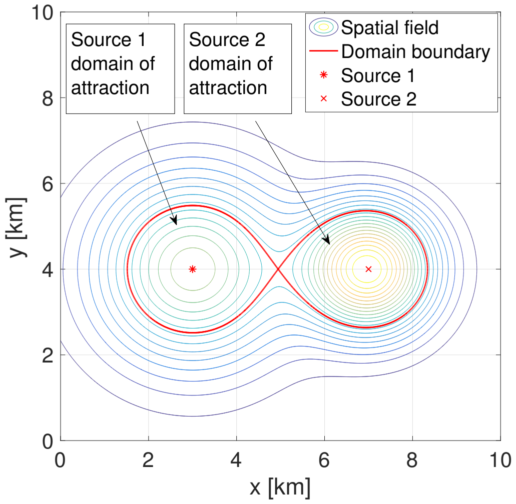


Figure 5.5.2: Domain of attraction - two different sources

delimited by the contour line highlighted in red. In the case of two identical sources, the point where $||\nabla\gamma(x, y)|| = 0$ is verified is exactly the mid point between the two sources. In Fig. 5.5.1b, instead, three identical sources are shown and the domain of attraction of each one is highlighted; in this case, the point characterized by $||\nabla\gamma(x, y)|| = 0$ lies in the mid point between sources 1 and 3 and between sources 2 and 3. Finally, the field shown in Fig. 5.5.2 is characterized by two sources having different intensity. In this case, the point characterized by $||\nabla\gamma(x, y)|| = 0$ is not the midpoint between the two sources, but it is shifted in the direction of the weakest source.

The results of a set of simulations, based on the numeric spatial field shown in Fig. 5.5.1b, are presented in order to highlight additional properties of the algorithm's local behaviour. The spatial field is obtained through three identical sources, having a Gaussian distribution. The cumulative effect of the sources at each position is obtained through:

$$f(x, y) = \sum_{i=1}^{3} \frac{1}{\sqrt{2\pi\sigma_{Gi}^2}} e^{-\frac{(x-\mu_{i_x})^2+(y-\mu_{i_y})^2}{2\sigma_{Gi}^2}} \qquad (5.5.1)$$

with $\sigma_{Gi} = 1$, $i = 1, \ldots, 3$, $(\mu_{1_x}, \mu_{1_y}) = (3, 4)$, $(\mu_{2_x}, \mu_{2_y}) = (7, 4)$ and $(\mu_{3_x}, \mu_{3_y}) = (5, 6)$. The results of a set of three simulations are shown in Fig. 5.5.3. In each

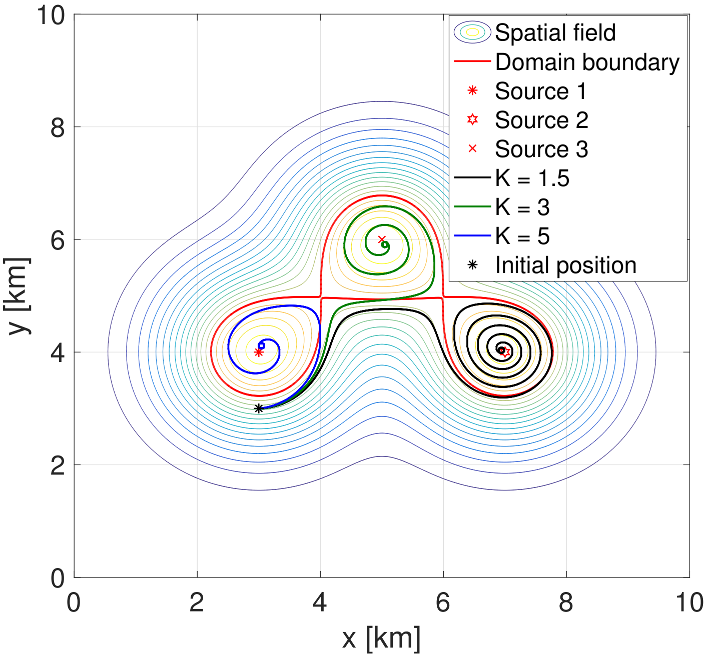
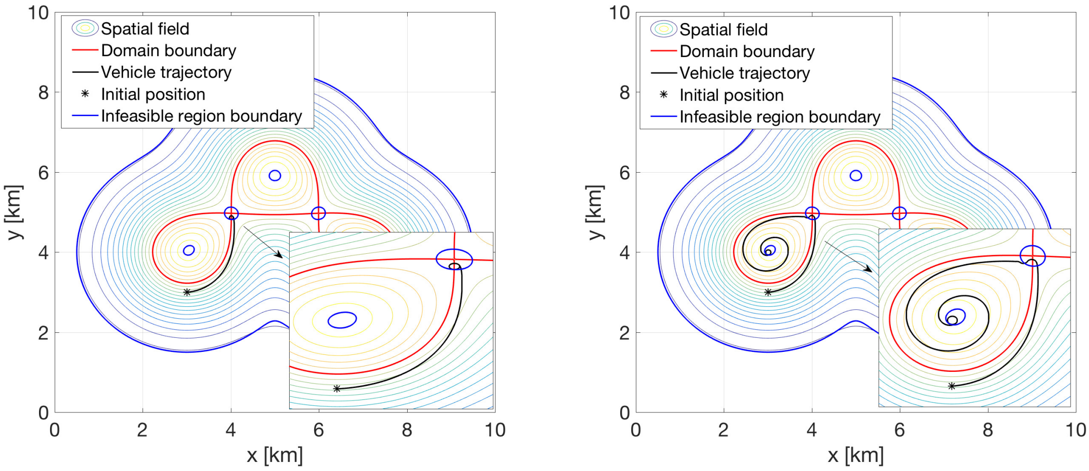


Figure 5.5.3: Numerical source seeking results - local convergence in the presence of three identical sources

simulation, a vehicle with speed $V = 1.8$ km/h is initially deployed at $(x(0), y(0)) = (3, 3)$, with $\theta(0) = 0$. The suboptimal sliding mode controller gains are set to $r_1 = 40$ and $r_2 = 1$, in order to obtain a minimum turning radius of the order of 45 m. The parameters of the reference trajectory definition are chosen as $\delta = 0.01$, and $K = 1.5$, $K = 3$ and $K = 5$ in the three different simulations. The vehicle's trajectories corresponding to the different reference trajectories are shown in Fig. 5.5.3 in black, green and blue respectively. It can be observed how the initial deployment of the vehicle is outside the domain of attraction of each of the local maxima, whose boundary is highlighted in red. Consequently, the source seeking algorithm drives the vehicle to a neighbourhood of the source whose domain of attraction is first entered. Choosing a slow reference trajectory, characterized by $K = 1.5$, the vehicle is driven to a neighbourhood of source 2; gradually increasing the reference trajectory parameter K to $K = 3$ and then to $K = 5$, makes the vehicle enter the domain of attraction of source 3 and source 1 respectively. In each case, the vehicle successfully reaches a small neighbourhood of the source and starts circling.

The reference trajectory's parameter K is chosen according to the condition in-



(a) Infeasible region - case 1

(b) Infeasible region - case 2

Figure 5.5.4: Numerical source seeking results - infeasible region

troduced in Section 5.4. When the spatial field is characterized by multiple local maxima, however, the implications of this choice need to be accounted for. It has been highlighted that, in the presence of more than one local maximum, there exists points in the domain, different from the sources, where $||\nabla\gamma(x,y)|| = 0$. Consequently, if the entire domain was chosen as the area of operation of the vehicle, the condition in (5.4.11) would have required $K = 0$, neglecting the possibility to perform source seeking. When choosing $K \neq 0$, consequently, an infeasible region around the points characterized by $||\nabla\gamma(x,y)|| = 0$ is introduced. This infeasible region can be characterized as:

$$\mathcal{I} = \left\{(x,y) \in \mathcal{D} : ||\nabla\gamma(x,y)|| < \frac{K\tanh(\delta)}{V}\right\} \tag{5.5.2}$$

If the vehicle enters the infeasible region, or if the vehicle is initially deployed inside this infeasible region, it is not possible to guarantee that it will successfully reach the neighbourhood of one of the local maxima, as the condition in (5.4.11) is not verified. As an example of this, the results shown in Fig. 5.5.4a are obtained with the same set up used previously, but with $K = 3.95$. With this choice, the value of the spatial field gradient on the boundary of the infeasible region results $||\nabla\gamma(x,y)|| = 0.022$, according to (5.5.2). The spatial field gradient contour line characterized by this specific value is shown in blue in Fig. 5.5.4a. The infeasible region is composed of the most external area of the domain, where the effect of the three sources is almost null, of a small area around each source, and of a small area around the points where $||\nabla\gamma(x, y)|| = 0$. As visible in the enlargement in Fig. 5.5.4a, as soon as the vehicle enters the infeasible region, it starts circling indefinitely, failing to further move towards the neighbourhood of one of the local maxima. One factor that may allow the vehicle to recover is its minimum turning radius R_{min}: if the minimum turning radius is big enough to allow the vehicle to leave the infeasible region while circling, than source seeking may be successful. An example of this is shown in Fig. 5.5.4b, where the controller's gain r_1 is reduced to $r_1 = 30$, thus increasing R_{min} to approximately 58 m. In this case, as visible from the enlargement in Fig. 5.5.4b, the vehicle recovers and successfully continues moving towards a neighbourhood of the source whose domain of attraction is entered. As a further comment, it can be observed how circling is started again as soon as the vehicle enters the small infeasible region centred at the local maxima position. This explains why only a neighbourhood of each source was reached in each simulation shown in Fig. 5.5.3.

5.6 Numerical example - dynamic source seeking

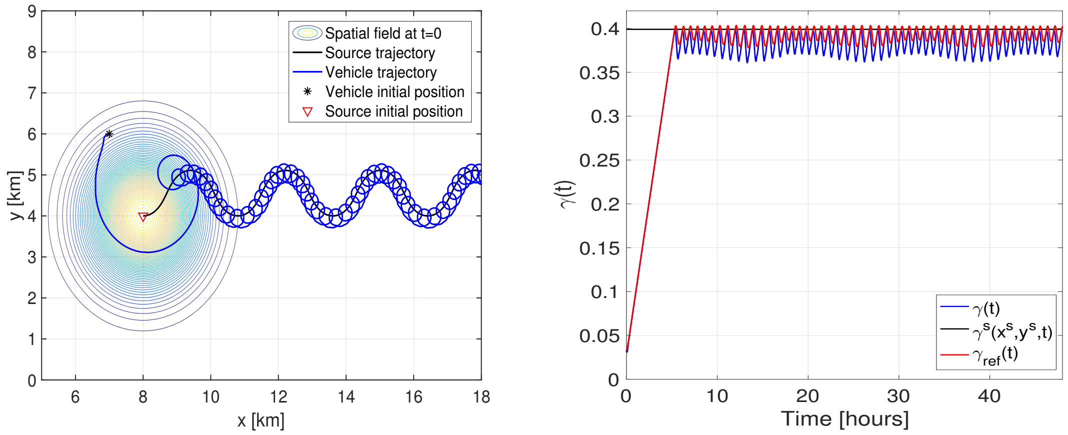
The following dynamic Gaussian distribution is considered:

$$\gamma(x, y, t) = \frac{a(t)}{\sqrt{2\pi}} e^{-\frac{(x-\mu_x(t))^2+(y-\mu_y(t))^2}{2}} \tag{5.6.1}$$

The movement of the centre of the distribution $(\mu_x(t), \mu_y(t)) = (x_c(t), y_c(t))$ is described through the following kinematic model:

$$\begin{aligned} \dot{x}_c(t) &= V_c \cos\theta_c(t) \\ \dot{y}_c(t) &= V_c \sin\theta_c(t) \\ \dot{\theta}_c(t) &= \frac{\pi}{3}\sin(0.5t) \end{aligned} \tag{5.6.2}$$

where $V_c = 0.3$ km/h is the speed of the centre of the distribution. Consequently, the centre of the distribution is modelled as a nonholonomic unicycle as in (3.2.3),



(a) Spatial field and vehicle's trajectory

(b) Vehicle's measurements

Figure 5.6.1: Numerical source seeking results - translating field

and it moves at constant speed along a sinusoidal trajectory. The initial position of the moving source is chosen as $(x_c(0), y_c(0)) = (8, 6)$, with $\theta_c(0) = 0$. The gain $a(t)$ is designed in order to obtain two different typologies of dynamic fields. In Section 5.6.1, $a(t) = 1$ is chosen, in order to obtain a translating spatial field. In section 5.6.2, the time-varying gain:

$$a(t) = 1 + 0.2\sin(2\pi 0.1t) \tag{5.6.3}$$

is considered, which describes the time-evolution of the intensity of the field; the maximum value at the position of the source, consequently, oscillates over time, and the value of the spatial field at each position of the domain varies accordingly.

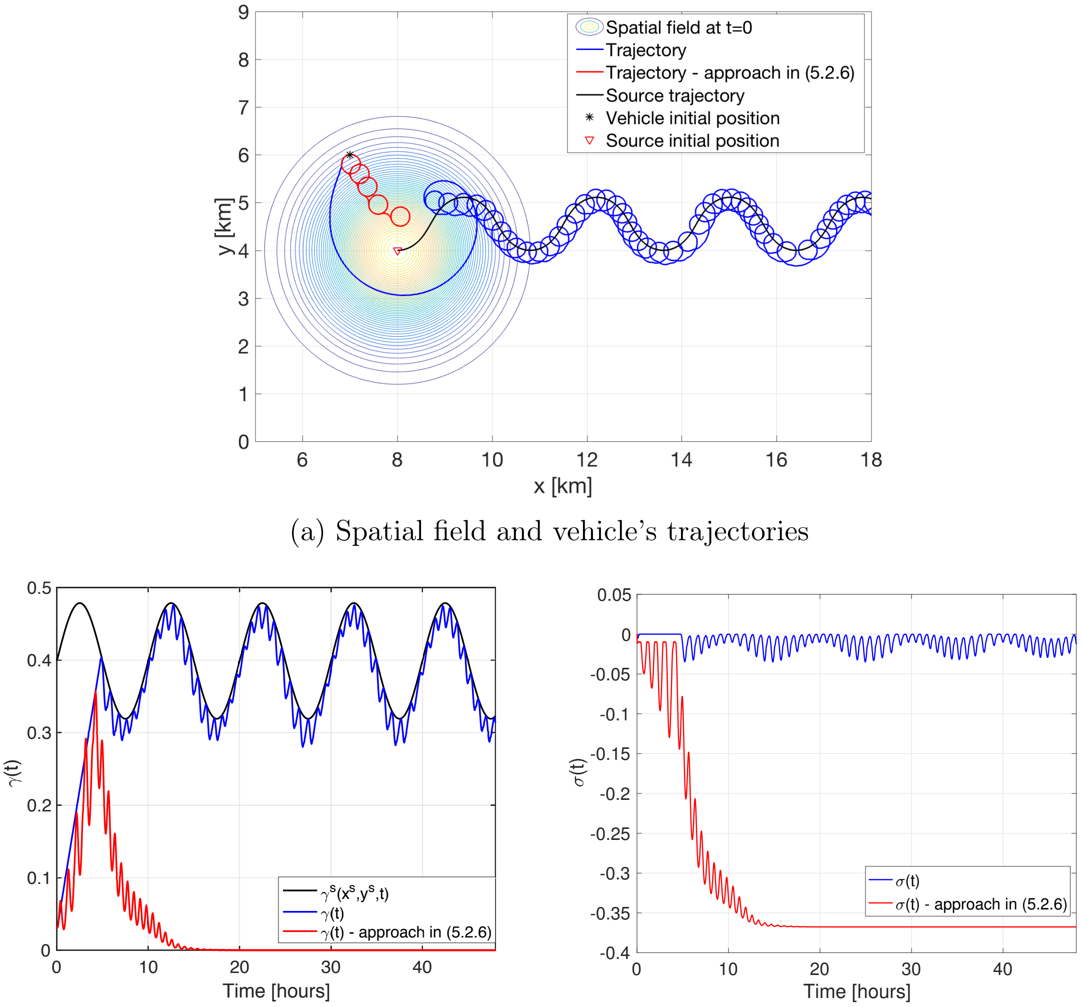
5.6.1 Approach validation - Translating field

The translating field is obtained through the Gaussian-like moving distribution in (5.6.1), with unitary variance and unitary gain $a(t)$. The conformation of the spatial distribution at $t = 0$ is shown in Fig. 5.6.1a, and it is insensitive to the movement of the source. The initial position of the source $(x_c(0), y_c(0))$ is highlighted with a red triangle and the trajectory of the source, which moves along a sinusoidal path, is shown in black.

The autonomous vehicle is modelled as in (3.2.3), with $V = 1.8$ km/h, and it is initially deployed at $(x(0), y(0)) = (7, 6)$ with $\theta(0) = 0$. The controller's gains in (3.3.10) are chosen as $r_1 = 10$, $r_2 = 1$. The parameters defining the reference trajectory in (5.4.1) are defined according to the sufficient condition in (5.4.10), which holds for translating fields. The parameter δ is set to $\delta = 0.01$. For the choice of the gain K, $||\nabla\gamma(x, y)||_{min} = 0.057$ is selected, obtaining, through (5.4.10), $K \leq 8.5503$. Consequently, $K = 8.55$ is chosen. The obtained results are shown in Fig. 5.6.1. The vehicle successfully reaches a neighbourhood of the sought source in finite time and starts circling as soon as sliding is lost. The vehicle, in addition, successfully follows the movement of the source. The time evolution of the reference trajectory $\gamma_{ref}(t)$, the measurements collected along the trajectory of the vehicle and the time-varying value of the spatial field at the source's position are compared in Fig. 5.6.1b, highlighting how the neighbourhood of the source is entered in approximately 5 hours and never left. In conclusion, the proposed approach successfully drives the vehicle inside a neighbourhood of the moving source and subsequently makes the vehicle move with the source, allowing the solution of a source seeking control problem in the case of a translating field.

5.6.2 Approach validation - Dynamic field

The dynamic field in (5.6.1) is considered, with the gain $a(t)$ designed as in (5.6.3). The sufficient condition in (5.4.9) is applied in order to choose the design parameter K in (5.4.1). The value of $||\nabla\gamma(x, y)||_{min}$ is chosen as in Section 5.6.1, obtaining, through (5.4.9), $K \leq 7.834$; consequently, $K = 7.8$ is chosen. The results obtained in simulation are shown in Fig. 5.6.2. Fig. 5.6.2a compares the trajectory of the moving source, shown in black, with the trajectory of the vehicle, in blue. The source and the vehicle's initial positions are highlighted with a red triangle and a black star respectively. It is possible to observe how the vehicle successfully moves from its initial position to a neighbourhood of the moving source and then starts circling around it, following its movement. This result is comparable to that presented in Section 5.6.1, with the difference that in this case the maximum value of the spatial field at the position of the source is itself a function of time. Fig. 5.6.2b shows



(a) Spatial field and vehicle's trajectories

(b) Vehicle's measurements

(c) Tracking error

Figure 5.6.2: Numerical source seeking results - dynamic field

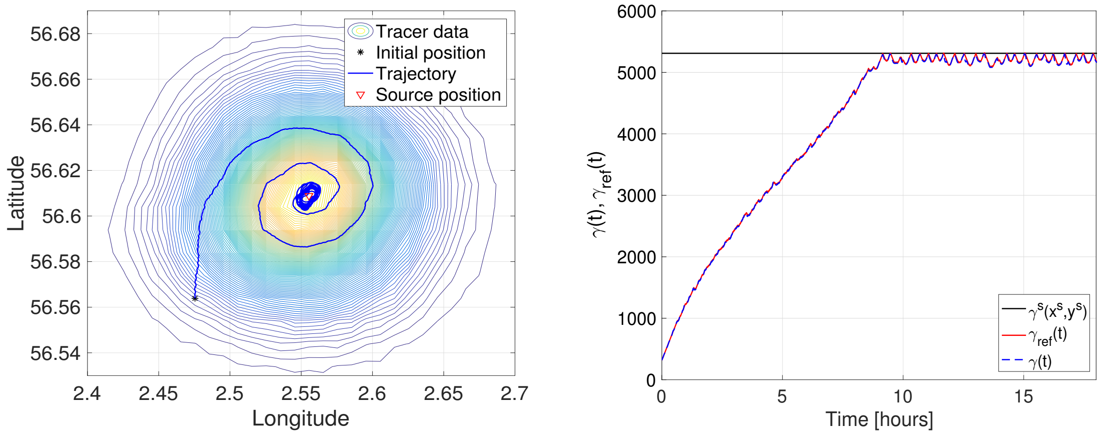
the time evolution of the field value at the position of the source in black and the measurements collected at the position of the vehicle in blue. The measurements collected by the vehicle follow the evolution of the maximum value $\gamma^s(x^s, y^s, t)$ with high accuracy.

The original source seeking approach, based on the reference trajectory defined in (5.2.5), is applied, with $K = 7.8$, to the same dataset. The results obtained are shown in Fig. 5.6.2 for comparison. It is possible to see how the original approach fails to make the vehicle seek a source with a time-varying value. The trajectory of the vehicle is shown in red in Fig. 5.6.2a. The vehicle stops before reaching a neighbourhood of the source position and it starts circling, unable to follow the movement of the source. This is confirmed by the measurements collected by the vehicle, shown in red in Fig. 5.6.2b: despite initially increasing, these decrease to a value very close to zero as the source moves away from the point where the vehicle starts circling.

5.7 Tracer release scenario

The source seeking approach proposed in this chapter is applied in a set of simulations based on the realistic dataset describing the evolution of a tracer release patch in an area of the Southern North Sea. The dataset has been introduced in Section 3.6. The results shown in Fig. 5.7.1 are obtained considering the spatial field as static, as in Section 3.6.1. Particularly, the tracer release configuration at the beginning of day 10 is considered. The vehicle's movement is modelled as in (3.2.3) and the vehicle's direction of movement is controlled through the source seeking approach based on the reference trajectory $\gamma_{ref}(t)$ defined in (5.4.1). The vehicle is initially deployed at $(x(0), y(0)) = (2.4754°, 56.5639°)$, with $\theta(0)$ arbitrarily chosen as $\frac{\pi}{2}$. Additionally, the controller's gains are set to $r_1 = 20$, $r_2 = 2$. The reference trajectory parameters are set to $\delta = 10$ and $K = 1000$. With these choices, the level of the acceptable tracking error is set to ±10 particles and a fast varying reference trajectory is introduced, determining a fast movement towards the sought source. The vehicle, as visible from Fig. 5.7.1a, successfully moves towards a small neighbourhood of the static source, whose position is highlighted with a red triangle. Once the small neighbourhood of the source is entered, it is never left and the vehicle definitely looses sliding. The reference trajectory $\gamma_{ref}(t)$, obtained through (5.4.1), is shown in Fig. 5.7.1b, together with the measurements collected at the vehicle's position. The obtained results highlight the applicability of the proposed source seeking approach to realistic datasets.

Being the gradient of the spatial field unknown when working with real world features, the choice of the parameter K in (5.4.1), which should satisfy the sufficient condition in (5.4.9), can be made following some general guidelines. Firstly, K is directly proportional to the vehicle's speed V; consequently, if working with a fast



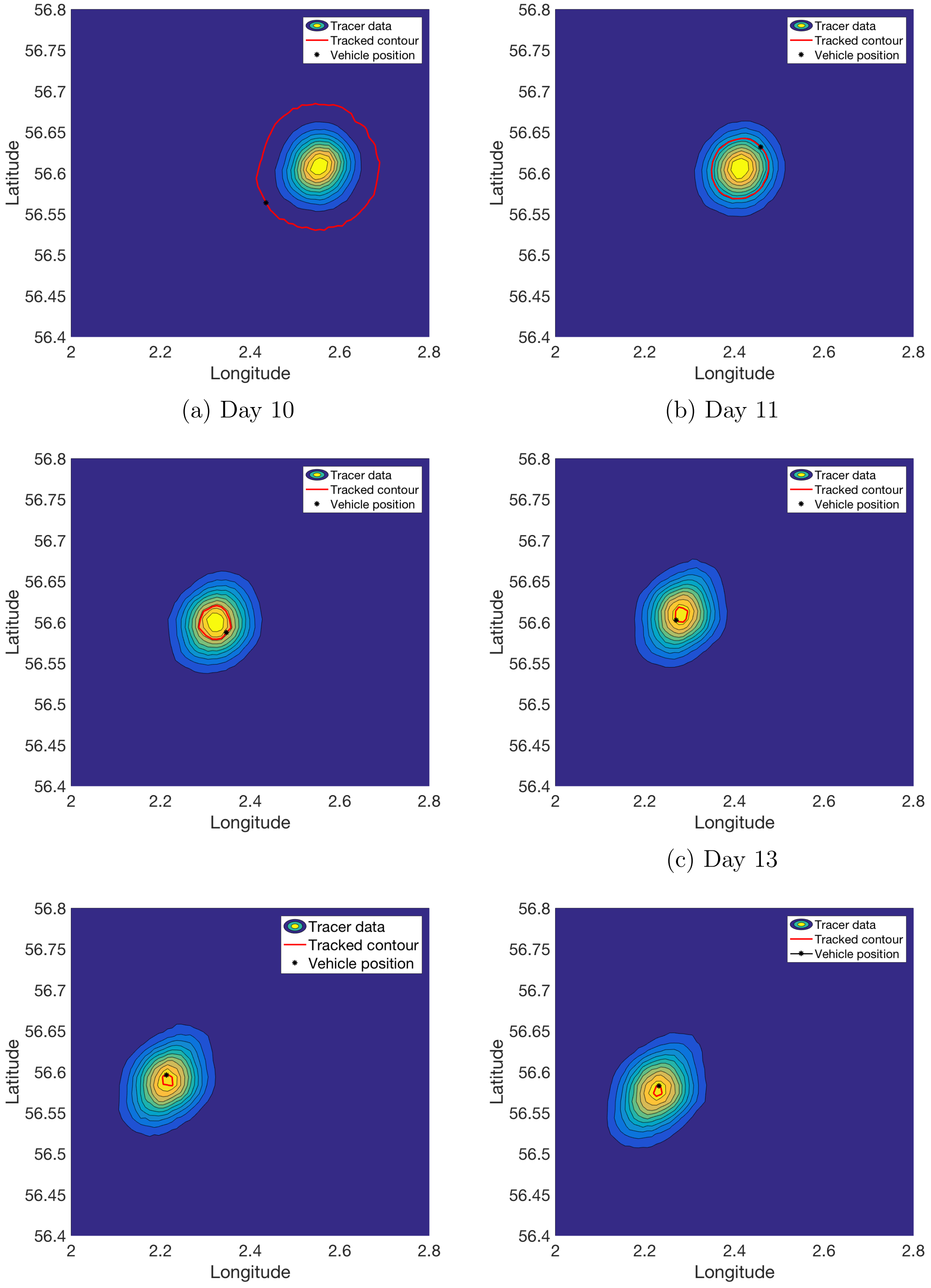
(a) Vehicle's trajectory

(b) Vehicle's measurement and reference trajectory

Figure 5.7.1: Source seeking results - static tracer release

vehicle, higher values of the parameter K should be preferred. Additionally, K is proportional to $\frac{1}{\tanh(\delta)}$, hence for small values of the acceptable tracking error δ, large slope parameters K can be selected. Finally, K is proportional to $||\nabla\gamma||_{min}$; consequently, larger values of K can be considered for highly changing fields, while smaller values should be chosen for shallower fields. As a final comment, it has been highlighted in Section 5.4.1, how the parameter K should be chosen in order to find a trade-off between the fast convergence to a neighbourhood of the source and the precision in the estimation of the position of the source.

The results shown in Fig. 5.7.2 are obtained considering a dynamic spatial field, obtained through the time-varying dataset relative to the tracer release scenario. The tracer release patch evolves through both the advection and diffusion phenomena, determined by the water currents present in the area and by the movement of particles towards areas of lower concentration. Consequently, the value of the spatial field at the source's location is not constant and it varies as shown in red in Fig. 5.7.3a. Despite having an overall decreasing nature, the number of particles at the source's location varies with oscillations, similarly to the periodic variations introduced in Section 5.6.2. As shown in the numerical simulation, the source seeking approach introduced in this chapter is capable of dealing with spatial fields having such characteristics. To confirm this, the results of two six days long simulations based on



(a) Day 10

(b) Day 11

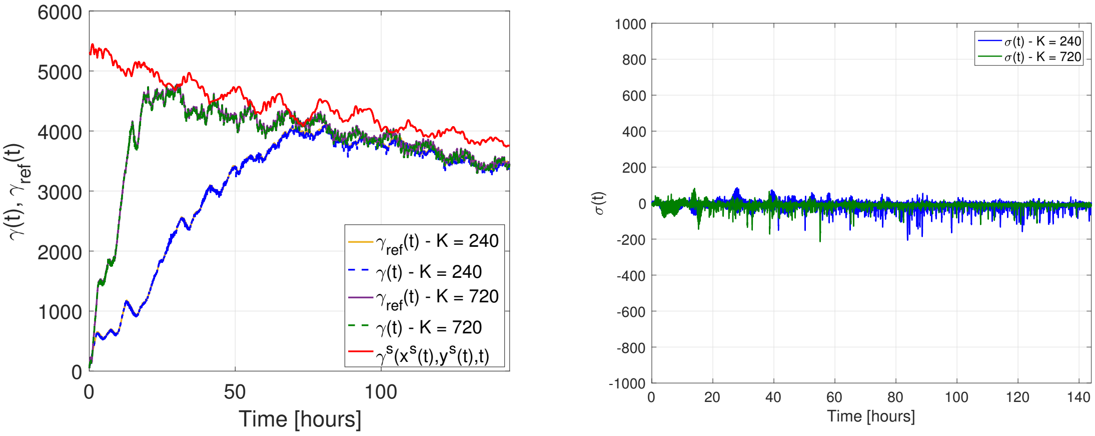
(c) Day 13

(d) Day 14

(e) Day 15

Figure 5.7.2: Dynamic tracer release source seeking - screen shots of the spatial field conformation and the vehicle's position

the realistic dynamic tracer release dataset are shown. The vehicle, moving with $V = 1$ m/s, is initially deployed at $(x(0), y(0)) = (56.5639°, 2.4354°)$, with $\theta(0) = 0$.

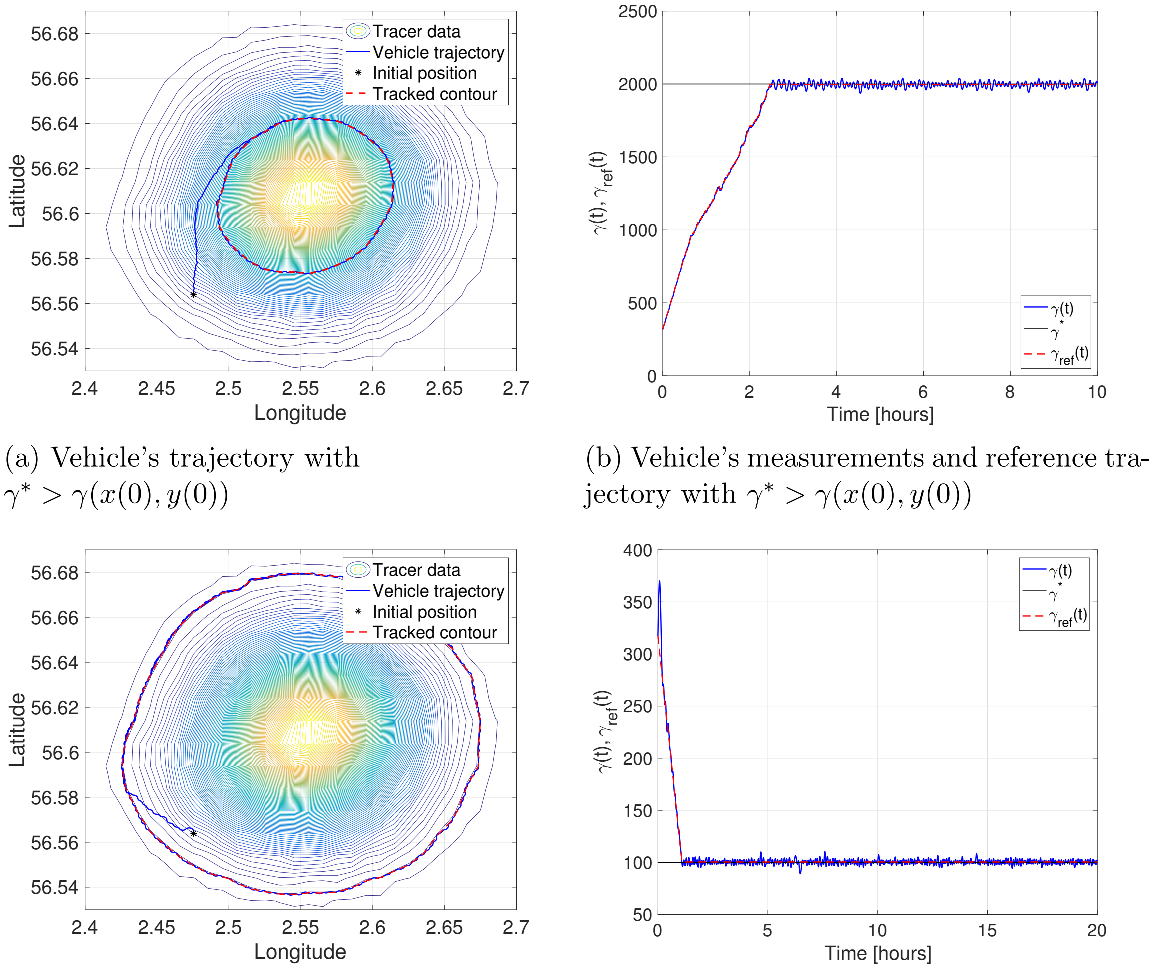


(a) Vehicle's measurements and reference trajectory

(b) Tracking error

Figure 5.7.3: Source seeking results - dynamic tracer release

The controller's gains in (3.3.10) are set to $r_1 = 11$, $r_2 = 0.4$. The reference trajectory's parameters in (5.4.1) are chosen as $\delta = 10$ and $K = 240$ and $K = 720$ respectively in two separate simulations, in order to alter the speed of convergence towards the sought source. The screenshots in Fig. 5.7.2 show the position of the vehicle and the configuration of the tracer patch at the beginning of each of the six days, in the case when $K = 240$ is chosen. Particularly, the position of the vehicle is shown as a black star, and the contour identified by $\gamma_{ref}(t)$ at a particular instant t is highlighted in red. These screenshots allow the visualization of the tracer patch evolution during the six days: the movement of the patch is mainly due to oceanic currents, while the patch spreads through diffusion. The reference trajectories and the measurements collected by the vehicle in the two simulations are shown in Fig. 5.7.3a: despite converging to a neighbourhood of the source in different times, in approximately 20 and 70 hours respectively, in both cases the vehicle successfully reaches a vicinity of the sought source in finite time and succeeds in subsequently following the movement of the source itself. The tracking errors obtained in the two cases are comparable in magnitude, being less than 100 particles in absolute value for the whole length of the simulation, and are shown in Fig. 5.7.3b.



(a) Vehicle's trajectory with $\gamma^* > \gamma(x(0), y(0))$

(b) Vehicle's measurements and reference trajectory with $\gamma^* > \gamma(x(0), y(0))$

(c) Vehicle's trajectory with $\gamma^* < \gamma(x(0), y(0))$

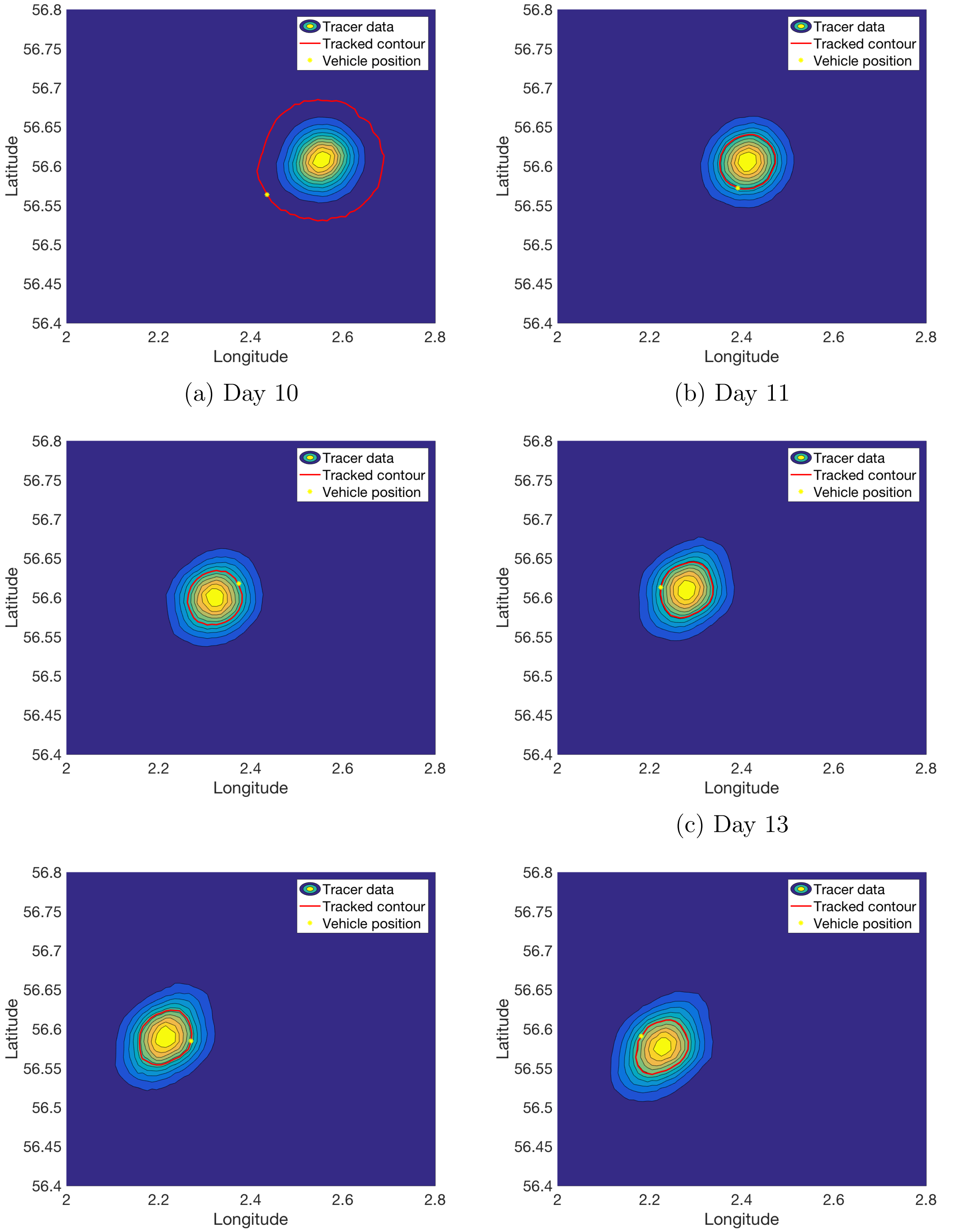
(d) Vehicle's measurements and reference trajectory with $\gamma^* < \gamma(x(0), y(0))$

Figure 5.7.4: Boundary tracking results - Static tracer release

5.7.1 Use of source seeking for boundary tracking - Tracer release results

In this section, the possibility to seek and track the boundary of the spatial field following the approach introduced in Section 5.4.2 is demonstrated in simulations. The results obtained from two separate simulations are shown in Fig. 5.7.4. In both simulations, the vehicle is initially deployed outside the vicinity of the tracked contour, at $(x(0), y(0)) = (2.4754°, 56.5639°)$, with $\theta(0) = \pi/2$. Additionally, the controller's gains are kept to $r_1 = 20$ and $r_2 = 2$ in both simulations. The results in Fig. 5.7.4a and Fig. 5.7.4b are obtained choosing the value of the spatial field on the tracked contour as $\gamma^* = 2000$ particles/unit. In order to guide the vehicle from its initial position to the tracked contour, the reference trajectory $\gamma_{ref}(t)$ is defined as in (5.4.13), with $K = 1000$ and $\delta = 10$. As visible in Fig. 5.7.4a, the vehicle is successfully driven from its initial position towards the tracked contour, which is reached in approximately 2.5 hours. As soon as the tracked contour is reached, the condition in (5.4.13) is triggered, and a good tracking of the boundary is attained by the vehicle. The evolution of the measurements collected by the vehicle, shown in Fig. 5.7.4b, confirms the performance. The reference trajectory $\gamma_{ref}(t)$ is shown in red. After the tracked value is reached, the measurements at the vehicle's position oscillate around it. This is due to the slow control action update rate, fixed to 30 seconds, and to the limited turning capability of the vehicle, imposed by the chosen values of the controller's gains r_1, r_2. Another factor impacting on the tracking accuracy is the high gradient of the spatial field in the area. The tracking error, however, is within a ± 40 particles band centred at the tracked value for the entire length of the sliding phase, from approximately 2.5 hours to the end of the simulation. In Fig. 5.7.4c and Fig. 5.7.4d the value of the spatial field on the tracked contour γ^* is chosen to be smaller than the measurement at the initial position of the vehicle $\gamma(x(0), y(0))$; particularly, $\gamma^* = 100$ particles/unit. In this case, the parameter K in the reference trajectory definition in (5.4.13) is set to $K = -200$, as the reference trajectory is required to be a decreasing signal, in order to drive the vehicle from its initial position to lower concentration points, until γ^* is reached. In this case, the tracked contour, highlighted in red in Fig. 5.7.4c, is reached after approximately 1 hour. Having reached the tracked boundary, the vehicle starts tracking it with high accuracy: the tracking error is contained, for the entire length of the simulation, in a ± 5 particles band around the tracked contour. Despite having maintained the same configuration, the tracking accuracy is increased in this case. This is due to the conformation of the tracked contour, which is significantly smoother, and to the shallower characteristics of the spatial field around the tracked contour.

Boundary tracking in the presence of a dynamic spatial field is also performed. Similarly to the source seeking application, the dynamic tracer release dataset is used in a six days long simulation. The tracked contour is characterized by $\gamma^* = 2000$



(a) Day 10

(b) Day 11

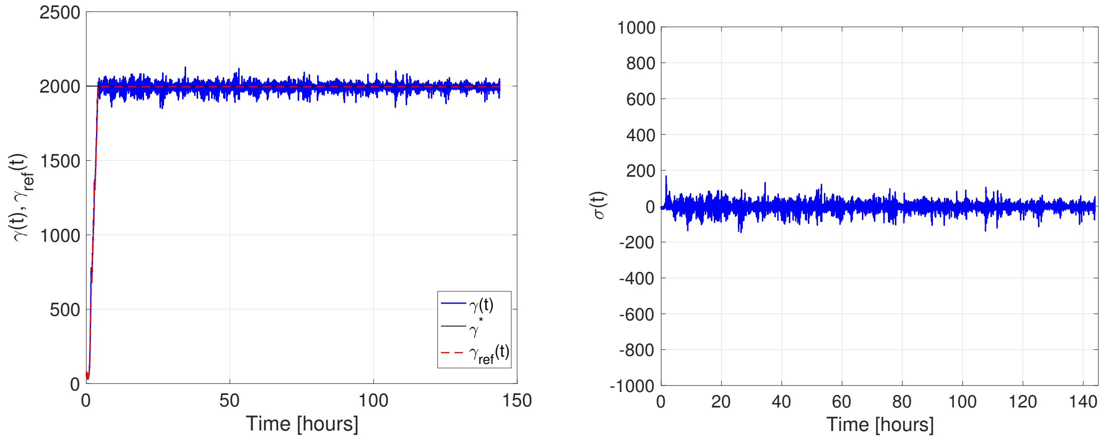
(c) Day 13

(d) Day 14

(e) Day 15

Figure 5.7.5: Dynamic tracer release boundary tracking - screen shots of the spatial field conformation and the vehicle's position

particles/unit, while $K = 240$ and $\delta = 10$ in (5.4.13) are chosen. The screenshots in Fig. 5.7.5 show the tracked boundary, characterized by the value of $\gamma_{ref}(t)$ at a specific instant t, in red and the vehicle's position as a yellow star. Boundary



(a) Vehicle's measurements and reference trajectory

(b) Tracking error

Figure 5.7.6: Boundary tracking results - dynamic tracer release

tracking is successfully achieved, as confirmed by the results in Fig. 5.7.6, where it is visible how the tracked value γ^* is reached in approximately 4 hours, after which the vehicle starts following the tracked contour with an accuracy of ±50 particles.

5.8 Concluding remarks

A source seeking algorithm using a single autonomous vehicle is proposed in this chapter. The methodology is developed starting from the suboptimal sliding mode boundary tracking approach presented in Chapter 3, and exploiting the principles of sliding mode extremum seeking techniques. The development of the algorithm is described in detail, starting from a preliminary version. The final formulation of the source seeking approach allows the introduction of a sufficient condition for the convergence to a neighbourhood of the sought source. The local nature of the approach is highlighted and the possibility to extend its applicability when considering a dynamic spatial field is shown. A modification of the proposed source seeking approach, moreover, is used to relax the assumption introduced in Chapter 3 on the initial position of the vehicle relative to the tracked contour. Through this approach, boundary tracking can be achieved even if the vehicle is initially deployed far from the tracked contour. Throughout the chapter, the approach is tested in a series of simulations based on numerical examples. Additionally, the dataset relative to the tracer release in an area of the Southern North Sea is used for the validation of the approach in a realistic example. The efficacy of the approach when dealing with both a static or a dynamic spatial field is highlighted. Additionally, the robustness of the approach to external disturbances, specifically the water currents affecting the movement of the vehicle, is practically demonstrated in the simulations.

Chapter 6

Probabilistic learning boundary tracking

6.1 Introduction

The principal advantages of deploying autonomous vehicles for the characterization of environmental features have been highlighted in Chapter 2. One less frequently highlighted advantage, which is mentioned in [1], is the possibility of integrating path-planning and control strategies with the available forecast models. The use of the available preliminary information for the guidance of autonomous vehicles can maximize the science outcomes, reduce the operational costs and the risks for the autonomous vehicles [21].

High fidelity mathematical models of the environment, such as the Met Office models, can provide forecast information about specific environmental features of interest prior to deploying the autonomous vehicles. This information can be useful at a high level planning stage for monitoring the specific event or feature. Model-aided path planning is aimed at maximizing the information collected through the deployment of autonomous vehicles [1]. This technique consists of pre-planning the trajectory to be followed by the vehicle, which is determined using a model of the environmental feature. The trajectory can also be determined online, using representative mathematical models of the feature. In [22], the output from an oceanic forecast model is used to design subsequent waypoints of a pre-planned trajectory for the autonomous vehicles, in order to optimize the sampling of an algae bloom phenomenon. The predictive tool used in [22] is the open source Regional Ocean Model System (ROMS) [124].

When using prior information from a model, it is advantageous to make use of the real world measurements collected by the autonomous vehicles for the calibration of the model itself, in order to improve the model prediction accuracy. In [22], for example, the water current measurements collected by a set of sensors deployed in the region of interest are used to improve the quality of the regional model. Similar techniques have been used to track a dynamically evolving oceanic algal bloom using multiple autonomous underwater gliders in [125]. The ocean predictions are used to plan the trajectory to be followed by the gliders, which is pre-programmed in order for the vehicle to be "in the right place at the right time", to collect data about the dynamic algal bloom.

In this chapter, the available forecast information relating to an environmental features of interest is made use of. A probabilistic model of the feature is built from the available forecast information and, similarly to [22], the probabilistic model is iteratively updated as and when additional real world observations are made available, in an effort to increase the model accuracy. The feature exploration is performed deploying a formation of autonomous vehicles. The vehicles, deployed over the region of interest, need to track the unknown boundary of the considered spatial field, while distributing themselves in a predefined formation. Similarly to Chapter 3, the spatial feature exploration objective is posed as a boundary tracking problem.

The proposed approach is significantly different from the model-aided path planning approaches available in the literature, as it does not involve a pre-defined path planning. The probabilistic model is used to estimate the spatial gradient of the field, to be used on-line for the boundary tracking control definition, rather than to pre-plan a trajectory of WPs. The proposed approach is tested in simulations, on the basis of a realistic dataset relating to the sea surface temperature in the

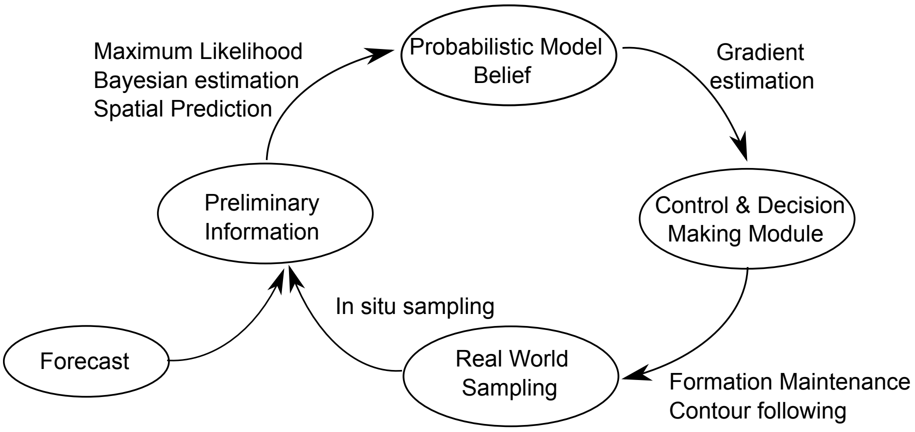


Figure 6.1.1: Schematic representation of the proposed approach. The main components are: (i) the real world; (ii) the forecast information; (iii) the preliminary information set; (iv) the probabilistic model; (v) the decision making module.

European North-West continental shelf.

The chapter is organized as follows: the proposed approach is briefly described in Section 6.2. The probabilistic model definition, validation and update procedure is presented in Section 6.3, while the gradient estimation technique is introduced in Section 6.4. The proposed probabilistic boundary tracking approach is presented in Section 6.5. A set of simulation results is presented in Section 6.6, while some concluding remarks are given in Section 6.7. Additional details about Gaussian Process models are given in Appendix C.

6.2 Approach description

A schematic of the probabilistic learning boundary tracking approach proposed in this chapter is shown in Fig. 6.1.1. The forecast block represents the available initial knowledge of the feature. The available preliminary forecast information is used to build a probabilistic model of the feature. The probabilistic model is an analytical statistical expression which can be used to estimate the spatial gradient of the spatial field at a specific location. The real world block represents the environment where the autonomous vehicles are deployed. For this application, a formation of three autonomous vehicles is considered, but similar ideas can be exploited if working with a single autonomous vehicle or a larger formation. The preliminary information, initially constituted of the sole forecast dataset, is iteratively enriched with the real world measurements collected by the vehicles and it is used to build and subsequently update the probabilistic model of the feature. Therefore, the probabilistic model of the feature is updated every time the preliminary information block is enriched with real world measurements. The updated model is used for the estimation of the gradient of the measured quantity at the desired location. The estimated gradient is finally used in the decision making module. A control law which makes use of the estimated gradient is applied to each vehicle. The control law aims at accomplishing two objectives: (i) the achievement and maintenance of the desired formation of vehicles; and (ii) the boundary tracking control objective. Differently from the boundary tracking problem considered so far, the idea here is that the virtual centroid of the formation of vehicles is required to precisely move on the tracked contour. The vehicles in the formation distribute themselves on the circumference of a circle around the virtual centroid.

6.3 Gaussian process model

The probabilistic model, built using the available preliminary information, analytically describes the evolution of the quantity of interest. The available preliminary information is a set $\mathcal{P} = \{\mathbf{x_i}, z_i\}$, where $i \in \{1, \dots, N\}$ is the information index. The input vectors $\mathbf{x_i} \in \mathbb{R}^2$ are the spatial locations of the available observations, constituted for example by longitude and latitude stamps, and $z_i \in \mathbb{R}$ are the available measurements, possibly affected by noise.

GP models are probabilistic models widely used to describe the probability distribution of unknown functions [126], [127]. A GP is essentially an extension of a Normal, or Gaussian, distribution. Whereas the Gaussian probability distribution describes random variables which are scalars, a GP governs the properties of functions. Consequently, a GP can be thought of as an infinite dimension Normal distribution [128]. Similarly to the mean and variance parameters of a Gaussian distribution, a GP is completely defined by a mean function $\mu(\boldsymbol{x})$ and a covariance function $c(\boldsymbol{x}, \boldsymbol{x}')$:

$$\chi(\boldsymbol{x}) \sim GP(\mu(\boldsymbol{x}), c(\boldsymbol{x}, \boldsymbol{x}')). \tag{6.3.1}$$

The mean function $\mu(\boldsymbol{x})$ describes the global trend of the data set, and it is often a polynomial function of the input. The coefficients of the polynomial mean function constitute the set of hyperparameters $\boldsymbol{\beta}$, associated with the mean part of the model. The covariance function describes the local trend of the dataset, that is how the output is expected to be correlated as a function of the input configuration. The covariance function $c(\boldsymbol{x}, \boldsymbol{x}')$ generally depends on two hyperparameters: the variance σ^2_{GP}, and the characteristic length scale ϕ. Particularly, the variance σ^2_{GP} determines to what extent the GP can deviate from the mean function, and the characteristic length-scale ϕ affects the smoothness of the dataset [128]. The main covariance function used in this thesis is the Squared Exponential (SE) covariance function:

$$c(\boldsymbol{x}, \boldsymbol{x}') = \sigma^2_{GP} e^{-\frac{h^2}{\phi^2}} \tag{6.3.2}$$

where h is the distance between the two input configurations $\boldsymbol{x}$ and $\boldsymbol{x}'$. The SE covariance function is amongst the most widely used in the literature, but it is sometimes criticized for excessively smoothing the difference between observations [129]. Consequently, a less smooth covariance function, the Matérn covariance function, is also considered:

$$c(\boldsymbol{x}, \boldsymbol{x}') = \sigma^2_{GP} \frac{\left(2\sqrt{\nu}\frac{h}{\phi}\right)^{\nu}}{2^{\nu-1}\Gamma(\nu)} K_{\nu}\left(2\sqrt{\nu}\frac{h}{\phi}\right) \tag{6.3.3}$$

where $\nu > 0$ is a design parameter, $K_{\nu}(\cdot)$ is a modified Bessel function of the second order and $\Gamma(\nu) = (\nu - 1)!$ is the Gamma function [128]. The covariance function in (6.3.3), in the particular case when the parameter ν is chosen as $\nu = \frac{3}{2}$, is used in this chapter, and it results:

$$c(\boldsymbol{x}, \boldsymbol{x}') = \sigma^2_{GP}\left(1 + \sqrt{3}\frac{h}{\phi}\right) e^{-\sqrt{3}\frac{h}{\phi}} \tag{6.3.4}$$

For a specific mean and covariance structure, the GP in (6.3.1) can be reformulated as:

$$\chi(\boldsymbol{x}) = \mu(\boldsymbol{x}) + Z(\boldsymbol{x}) + \epsilon(\boldsymbol{x}), \tag{6.3.5}$$

where $\mu(\boldsymbol{x})$ is the mean function and $Z(\boldsymbol{x})$ is a zero-mean Gaussian process characterized as:

$$Z(\boldsymbol{x}) \sim GP\left(0, c(\boldsymbol{x}, \boldsymbol{x}')\right) \tag{6.3.6}$$

The nugget $\epsilon(\boldsymbol{x})$ is introduced to model instrumental or measurement noise, possibly affecting the observations $z_i \in \mathcal{P}$ [130]. The noise is assumed to follow an independent, identically distributed Gaussian distribution, with zero mean and variance σ_n^2:

$$\epsilon(\boldsymbol{x}) \sim \mathcal{N}(0, \sigma_n^2) \tag{6.3.7}$$

The hyperparameters vector associated with the GP model in (6.3.5) is given by $\boldsymbol{\vartheta} := (\boldsymbol{\beta}, \sigma_{GP}^2, \phi, \sigma_n^2)$. When fitting a GP to the available forecast information, a specific value is associated with each hyperparameter and this is selected through a model fitting and validation procedure. Additional details about the structure of a GP and the most commonly used mean and covariance functions are given in Appendix C.

6.3.1 Maximum likelihood

There exists different approaches for fitting a GP to the available preliminary information $\mathcal{P}$. One approach is the maximum likelihood procedure, whose aim is to obtain the best estimate of the values of the hyperparameters $\boldsymbol{\vartheta}$, given a specific structure for the mean and the covariance functions.

The likelihood $L(\vartheta) = p(\mathcal{P}|\boldsymbol{\vartheta})$ is defined as the probability of obtaining the exact observations $z_i \in \mathcal{P}$, constituting the preliminary information, at locations $\mathbf{x_i} \in \mathcal{P}$, from the fitted probabilistic model, by assigning specific values to the hyperparameters. Considering each data point in the preliminary information set $\mathcal{P}$ separately, the likelihood can be factorized as:

$$L(\vartheta) = \prod_{i=1}^{N} p(z_i|\boldsymbol{\vartheta}) \tag{6.3.8}$$

The objective of the maximum likelihood procedure is to select the set of hyperparameters $\hat{\boldsymbol{\vartheta}}_{MLE}$ that solves the following optimization problem:

$$\hat{\boldsymbol{\vartheta}}_{MLE} = \arg\max_{\boldsymbol{\vartheta}} \; p(\mathcal{P}|\boldsymbol{\vartheta}) = \arg\max_{\boldsymbol{\vartheta}} \; L(\boldsymbol{\vartheta}) \tag{6.3.9}$$

Consequently, when fitting a model through the maximum likelihood procedure, the structure of the mean and the covariance functions is chosen a-priori. The solution of the optimization problem is solely aimed at identifying the best choice for the hyperparameters $\hat{\boldsymbol{\vartheta}}_{\boldsymbol{MLE}}$.

Different optimization methods can be used to determine $\hat{\boldsymbol{\vartheta}}_{\boldsymbol{MLE}}$ from the optimization problem in (6.3.9). The statistic friendly software 'R' [131], which has been used to obtain the results presented in this chapter, exploits the Nelder-Mead simplex optimization method [132], which is a downhill simplex method. The starting point of the method is an initial simplex ($\boldsymbol{\vartheta}_0$), which constitutes the initial guess for the values of the distribution hyperparameters. The sequence of simplexes is generated making use of mainly reflection, expansion, contraction, and shrink operations, while exploiting the tie-breaking rules in [133]. Additional details on the algorithm can be found in [132].

The fitted model can be used in spatial prediction techniques [134] in order to build the belief, which is a probabilistic data set defining the value of the spatial field on a grid, possibly with higher resolution with respect to the preliminary information. Particularly, the belief is built inferring likely values $\hat{\mathbf{z}}$ of the spatial feature at new input configurations $\hat{\mathbf{x}}$. Spatial prediction is performed through a global neighbourhood approach, in which all observed data in the preliminary dataset contribute to the prediction at a specific new location [135]. The prediction $\hat{z}_i$ at the new location

$\hat{x}_i$ is obtained through:

$$\hat{z}_i = \sum_{i=1}^{N} \lambda_i z_i \tag{6.3.10}$$

where the weights λ_i in (6.3.10) determine the influence of each observation in the training data set $\mathcal{P}$. The weights λ_i associated with each observation $z_i \in \mathcal{P}$ are chosen to be inversely proportional to the distance between the observation and the prediction location [136]. In this way, observations close to the prediction location have an higher influence on the prediction with respect to far observations.

In the prediction procedure, the hyperparameters estimated through the maximum likelihood procedure are treated as true values; consequently, the uncertainty associated with the hyperparameters is neglected and the fitted model is treated as a perfect representation of the considered spatial feature.

6.3.2 Bayesian estimation

An alternative model fitting procedure is the Bayesian approach. In this case, every hyperparameter is assumed to be uncertain and is characterized by a prior and a posterior probability distribution, respectively at the beginning and at the end of the estimation procedure. The prior distribution represents the available preliminary knowledge associated with the hyperparameters. As only limited knowledge is typically available, the prior is often represented through a non-informative distribution, such as a flat or a reciprocal distribution [128]. The posterior distribution, which is initially unknown, represents the updated knowledge. This is obtained by combining the existing knowledge (represented by the prior) and the available data (through the likelihood).

The Bayesian estimation procedure is based on Bayes theorem [128], through which the posterior distribution is built. Bayes theorem states that:

$$p(\boldsymbol{\vartheta}|\mathbf{z}_i) \propto p(\mathbf{z}_i|\boldsymbol{\vartheta})p(\boldsymbol{\vartheta}) \tag{6.3.11}$$

where $p(\boldsymbol{\vartheta})$ is the prior distribution, $p(\mathbf{z}_i|\boldsymbol{\vartheta})$ is the likelihood and $p(\boldsymbol{\vartheta}|\mathbf{z}_i)$ is the posterior distribution.

The key difference between the Bayesian estimation and the maximum likelihood procedure described in Section 6.3.1 is the characterization of the hyperparameters. These are not simply a best estimate, but they are characterized by a probability distribution, with a specified mean and variance. The belief is therefore built through spatial prediction, which takes into account the uncertainty associated with the hyperparameters. The result of the spatial prediction is different in this case, as each observation is characterized by a probability distribution[130].

The estimates of the hyperparameters are chosen in order to match the posterior distribution. This is done through the Bayesian estimation process, based on a loss function, which associates a loss or cost of having a hyperparameter estimate with a certain error. The Bayesian estimation procedure aims at minimizing this loss. In the Bayesian approach, the estimate of an hyperparameter is obtained seeking a solution to the following problem:

$$\hat{\vartheta} = \arg\min_{\vartheta} E(C(\hat{\vartheta}, \vartheta)) \tag{6.3.12}$$

where $C(\hat{\vartheta}, \vartheta)$ is a non-negative cost function, whose expected value is to be minimized. Particularly, choosing the mean square error cost function $C(\hat{\vartheta}, \vartheta) = |\hat{\vartheta} - \vartheta|^2$, the mean of the posterior distribution is obtained as the hyperparameter estimate [137].

6.3.3 Model validation metrics

When fitting a probabilistic model, different structures for the model and different methods for developing the models are possible. Consequently, it is important to choose the best fitting model for a specific dataset through a set of validation metrics.

Certain validation metrics can be used when comparing probabilistic models obtained using the maximum likelihood procedure described in Section 6.3.1. The values of the maximized likelihood obtained from the model fitting procedure itself can be considered as a metric. Specifically, a model characterized by a higher maximized likelihood should be preferred. The Bayesian Information Criterion (BIC) and the Akaike Information Criterion (AIC) [128] can also be considered as metrics. These are defined, respectively, as:

$$\begin{aligned} BIC &= \ln(N)H - 2\ln(\hat{L}(\boldsymbol{\vartheta})) \qquad (6.3.13) \\ AIC &= 2H - 2\ln(\hat{L}(\boldsymbol{\vartheta})) \qquad (6.3.14) \end{aligned}$$

where $\ln(\cdot)$ is the natural logarithm, N is the number of training data points, that is the number of data points in $\mathcal{P}$, H is the number of hyperparameters characterizing the fitted model and $\hat{L}(\boldsymbol{\vartheta})$ is the maximized likelihood. These indicators are a useful tool when comparing different probabilistic models fitted to the same initial dataset, as they can be used to reduce the problem of overfitting. The BIC and AIC metrics, in fact, introduce a penalty term for the number of hyperparameters in the model. The model characterized by the smaller AIC or BIC should be preferred.

Additional validation methods can be used in order to compare models fitted through the maximum likelihood procedure. Firstly, the leave-one-out technique [128] can be used. In this technique, one training data point at a time $(\mathbf{x_i}, z_i) \in \mathcal{P}$ is excluded from the preliminary dataset and thus from the model fitting procedure. The prediction at that specific location obtained from the newly fitted model is compared with the exact observation, in order to evaluate the prediction accuracy. To visualize the results obtained from this procedure, the so-called Quantile-Quantile (QQ) plots are often employed. A QQ plot is a graphical method used to compare two statistical distributions [138]. In the leave-one-out validation technique, it is used to compare the theoretical probability distribution of the data and the observed probability distribution. If the obtained points lie on the line $y = x$, it means that the two distributions are the same, while if the points move significantly away from the first quadrant diagonal, then the two distributions can be judged highly distinguished. The leave-one-out technique is computationally expensive, as a new model needs to be fitted for every training data point, requiring a total of N model fittings.

External validation techniques are based on the availability of additional observations, not included in the preliminary information set $\mathcal{P}$. For the applicability of these methods, a set of external validation points is required, where the exact value of the field is known. The known value at the external validation point location is compared with the predictions obtained from the probabilistic model at the same locations. Different evaluation metrics can be considered. The first one is the Root Mean Square Error (RMSE): denoting with z_i the value of the feature at location x_i in the validation set and with $\hat{z}_i$ the estimated value from the probabilistic model at the same location, the RMSE is defined as:

$$RMSE = \sqrt{\frac{1}{m}\sum_{i=1}^{m}\left(z_i - \hat{z}_i\right)^2} \tag{6.3.15}$$

where m is the number of validation points in the validation set. Models characterized by a small RMSE represent the spatial feature with high fidelity and can be considered accurate.

As a second metric, the capability of the fitted probabilistic model to accurately estimate the true value of the mean and the variance of the original data set is considered. The estimated values of the mean and variance, obtained from the probabilistic model on a specific validation set, are obtained as:

$$\hat{\mu}_z = \frac{1}{m}\sum_{i=1}^{m}\hat{z}_i \tag{6.3.16}$$

$$\hat{\sigma}_z = \frac{1}{m}\sum_{i=1}^{m}(\hat{z}_i - \hat{\mu}_z)^2 \tag{6.3.17}$$

These are compared with the actual mean and variance of the observations in the external validation set.

A third metric evaluates the capability of the fitted probabilistic model to estimate the chance of rare events: this is the exceedence probability, that is the probability that the observation exceeds a predefined threshold. Particularly, the following exceedence probability can be considered:

$$P(z_i \geq \mu_z + \sigma_z) \tag{6.3.18}$$

which can be estimated through:

$$\hat{P}(z_i \geq \mu_z + \sigma_z) = \frac{1}{m} \sum_{i=1}^{m} \mathbf{1}(\hat{z}_i \geq \mu_z + \sigma_z) \tag{6.3.19}$$

where $\mathbf{1}$ is the indicator function, which is equal to 1 if the condition $(\hat{z}_i \geq \mu_z + \sigma_z)$ is verified, and 0 otherwise.

6.3.4 Probabilistic learning strategy

In this chapter, the probabilistic model of the feature is initially fitted making use of the preliminary information $\mathcal{P}$ from the forecast model. As previously mentioned, the observations $z_i \in \mathcal{P}$ may be affected by noise. In order to enhance the accuracy of the surrogate model, an iterative probabilistic learning procedure is applied. The measurements collected in the real environment by the autonomous vehicles are used to enrich the preliminary data set $\mathcal{P}$. When new measurements are made available, the model fitting steps presented in Sections 6.3.1 and 6.3.2 are repeated, using the augmented preliminary information, and calibrated belief probabilistic data sets are built. Since the number of real-world measurements increases, the accuracy of the overall model is enhanced. Particularly, the enhancement of the model accuracy can be assessed comparing the RMSE in (6.3.15) obtained at the beginning and at the end of the probabilistic learning strategy.

6.4 Gradient estimation

The fitted GP in (6.3.1), which describes the spatial field through an analytical expression of the spatial coordinates, can be used to obtain an estimate of the field gradient at the desired location. The availability of a gradient estimate is particularly useful when solving a boundary tracking control problem. Traditionally, in the literature, gradient is estimated making use of the measurements collected by a formation of vehicles and of the knowledge of their relative positions [72], [139]. Here, the gradient of the spatial field at the desired location is estimated making use of the fitted probabilistic model. The derivative of a GP, in fact, is another GP which can be used to estimate the gradient information [128].

A formation of n autonomous vehicles is considered. The vehicles are assumed to be identical and to uniformly distribute on the circumference of a circle, whose centre is the virtual centroid. The gradient at the position of the $k-$th vehicle in the formation can be estimated by computing the partial derivatives of the GP in (6.3.1) through:

$$\left(\frac{\partial \chi(\boldsymbol{x}))}{\partial x}, \frac{\partial \chi(\boldsymbol{x})}{\partial y}\right)_k \tag{6.4.1}$$

where the subscript $k = 1, \ldots, n$ indicates that the derivatives are evaluated at the $k-$th vehicle's position. Defining the centroid position as:

$$(x_c, y_c) := \left(\frac{1}{n}\sum_{k=1}^{n} x_k, \frac{1}{n}\sum_{k=1}^{n} y_k\right) \tag{6.4.2}$$

where $(x_k, y_k), k = 1, \ldots, n$ are the vehicles' locations, the gradient at the centroid position, defined as:

$$\nabla \chi(\boldsymbol{x})\big|_{(x_c,y_c)} = \left[\frac{\partial \chi(\boldsymbol{x})}{\partial x}, \frac{\partial \chi(\boldsymbol{x})}{\partial y}\right]^T \;, \tag{6.4.3}$$

can be estimated through an averaging procedure:

$$\left.\frac{\partial \chi(\boldsymbol{x})}{\partial x}\right|_{(x_c,y_c)} = \frac{1}{n}\sum_{k=1}^{n}\left.\left(\frac{\partial \chi(\boldsymbol{x})}{\partial x}\right)_k\right|_{(x_c,y_c)} \tag{6.4.4}$$

$$\left.\frac{\partial \chi(\boldsymbol{x})}{\partial y}\right|_{(x_c,y_c)} = \frac{1}{n}\sum_{k=1}^{n}\left.\left(\frac{\partial \chi(\boldsymbol{x})}{\partial y}\right)_k\right|_{(x_c,y_c)} \tag{6.4.5}$$

In conclusion, the spatial gradient is initially estimated at the location of each vehicle in the formation, making use of the fitted probabilistic model. Subsequently, the gradient at the centroid position is estimated through an averaging step. The main advantage of the averaging step is the attenuation of the effect of measurement noise on the gradient estimation and the consequent increased accuracy of the estimation.

Considering the GP in (6.3.5), with the SE covariance function in (6.3.2), the partial derivatives in (6.4.1) result:

$$\left(\frac{\partial \chi(\boldsymbol{x})}{\partial x}\right)_k = \frac{\partial \mu(\boldsymbol{x})}{\partial x} + Z^{'}_{x,k}(\boldsymbol{x}) \tag{6.4.6}$$

$$\left(\frac{\partial \chi(\boldsymbol{x})}{\partial y}\right)_k = \frac{\partial \mu(\boldsymbol{x})}{\partial y} + Z^{'}_{y,k}(\boldsymbol{x}) \tag{6.4.7}$$

where the GPs $Z^{'}_{x,k}(\boldsymbol{x})$ and $Z^{'}_{y,k}(\boldsymbol{x})$ are defined as:

$$Z^{'}_{x,k}(\boldsymbol{x}) \sim GP\left(0, -\sigma^2_{GP} e^{-\frac{h^2}{\phi^2}}\left(\frac{2}{\phi^2}\right)(x - x_k)\right) \tag{6.4.8}$$

$$Z^{'}_{y,k}(\boldsymbol{x}) \sim GP\left(0, -\sigma^2_{GP} e^{-\frac{h^2}{\phi^2}}\left(\frac{2}{\phi^2}\right)(y - y_k)\right) \tag{6.4.9}$$

If, instead, the probabilistic model with the Matérn covariance function in (6.3.4) is used, then the definition of the GPs $Z^{'}_{x,k}(\boldsymbol{x})$ and $Z^{'}_{y,k}(\boldsymbol{x})$ results:

$$Z^{'}_{x,k}(\boldsymbol{x}) \sim GP\left(0, -\frac{6\sigma^2_{GP}(x - x_k)}{\phi^2} e^{-\sqrt{3}\frac{h}{\phi}}\right) \tag{6.4.10}$$

$$Z^{'}_{y,k}(\boldsymbol{x}) \sim GP\left(0, -\frac{6\sigma^2_{GP}(y - y_k)}{\phi^2} e^{-\sqrt{3}\frac{h}{\phi}}\right) \tag{6.4.11}$$

A numerical example, applying the gradient estimation procedure is given in Ap-

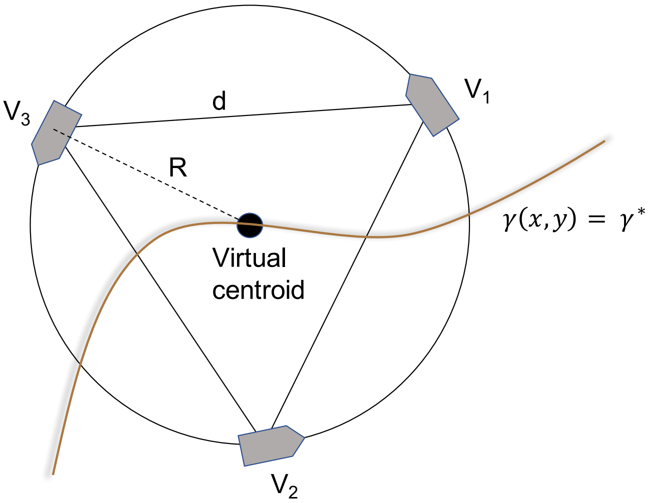


Figure 6.5.1: Desired equilateral triangular formation

pendix C.

6.5 Probabilistic model based boundary tracking

The probabilistic boundary tracking approach is based on the availability of a probabilistic model of the environmental feature, built according to the procedure presented in Section 6.3. Additionally, the approach relies on the deployment of a formation of vehicles, whose real world measurements are used to enhance the accuracy of the probabilistic model. The probabilistic boundary tracking approach influences the movement of the virtual centroid of the formation, which is controlled through a steering law in order to achieve the boundary tracking control objective. The vehicles are maintained in the desired formation around the moving centroid through a formation control strategy.

6.5.1 Formation control scheme

For demonstrating the probabilistic model based boundary tracking control problem, a formation of three identical vehicles is considered here. The vehicles are required to distribute themselves into an equilateral triangular formation around the moving virtual centroid, such that the desired distance between the vehicles d is constant. Equivalently, the vehicles are requested to uniformly distribute on the circumference of a circle of radius $R = d/\sqrt{3}$, centred at the virtual centroid position. A schematic visualization of the desired formation is shown in Fig. 6.5.1. In the literature, there exists several strategies useful to obtain the formation in Fig. 6.5.1. Amongst these, are [69], [140], [78] [141], [142] and [143]. The desired formation should also move with the virtual centroid, whose movement is controlled through the probabilistic model based boundary tracking approach, and happens along an a-priori unknown trajectory.

The formation control approach originally reported in [78] is chosen. This approach models each vehicle in the formation as a modified kinematic unicycle, obtained from the model in (3.2.3) by introducing a second control input. Specifically, the speed of the vehicle is controlled in addition to its angular velocity. As highlighted in [72], the formation control strategy in [78] builds on the work in [144], by allowing the vehicles to maintain a stable formation around a moving virtual centroid, whose trajectory can be unknown a-priori.

The virtual centroid, which moves at constant speed V_c, is assumed to satisfy the kinematic unicycle nonholonomic constraint introduced in (3.2.3). The control action influencing the movement of the virtual centroid is the steering law $\dot{\theta}_c(t) = u_c(t)$ It is assumed in [78] that the first and second derivatives of the centroid position states (x_c, y_c) are known and bounded. This hypothesis is guaranteed by the kinematic unicycle model in (3.2.3); specifically, the following bounds hold:

$$|\dot{x}_c|, |\dot{y}_c| \leq V_c \tag{6.5.1}$$

$$|\ddot{x}_c|, |\ddot{y}_c| \leq V_c\dot{\theta}_{c,max} = V_c u_{c,max} \tag{6.5.2}$$

where $u_{c,max}$ is the maximum value of the steering control applied to the virtual centroid. Additionally, it is assumed in [78] that an all-to-all communication network is established amongst the vehicles in the formation. These hypothesis are maintained here. The modified kinematic unicycle model used for each vehicle in the formation results:

$$\begin{cases} \dot{x}_k(t) & = v_k(t)\cos(\theta_k(t)) \\ \dot{y}_k(t) & = v_k(t)\sin(\theta_k(t)) \\ \dot{\theta}_k(t) & = u_{k1}(t) \\ v_k(t) & = u_{k2}(t) \end{cases} \tag{6.5.3}$$

where $k = 1, \dots, 3$ and the control inputs are the steering control $u_{k1}(t)$ and the speed $u_{k2}(t)$. The second control action $v_k(t) = u_{k2}(t)$ has been introduced to ensure the capability of the vehicles of following the moving centroid [78].

A change of coordinates is introduced in [78] in order to express each vehicle's position relative to the centroid position:

$$\begin{aligned} \dot{\tilde{x}} &= \dot{x}_k - \dot{x}_c = v_0 \cos\psi_k \\ \dot{\tilde{y}} &= \dot{y}_k - \dot{y}_c = v_0 \sin\psi_k \end{aligned} \tag{6.5.4}$$

where the constant v_0 satisfies the bound:

$$v_0 > \max_{t \geq 0}\{|\dot{c}_d(t)|\}, \tag{6.5.5}$$

with $\dot{c}_d = [\dot{x}_c \quad \dot{y}_c]^\top$ and ψ_k is given by:

$$\psi_k = \arctan\left(\frac{v_k \sin\theta_k - V_c \sin\theta_c}{v_k \cos\theta_k - V_c \cos\theta_c}\right) + \varsigma_k \pi \tag{6.5.6}$$

The scalar ς_k is used to guarantee that $\psi_k \in [0, \pi]$ and it is chosen as $\varsigma_k = 0$ if $(v_k(0)\cos\theta_k(0) - V_c \cos\theta_c(0)) > 0$ and $\varsigma_k = 1$ otherwise. The steering law to be applied to each vehicle in the formation is [78]:

$$u_{k1} = \dot{\psi}_k + \frac{V_c}{v_k}\cos(\theta_k - \theta_c)(\dot{\theta}_c - \dot{\psi}_k) \tag{6.5.7}$$

where $\dot{\theta}_c = u_c$ and the evolution of ψ_k is determined through [78]:

$$\dot{\psi}_k = \omega_0 \left(1 + \zeta_1 v_0 \left(\cos\psi_k(x_k - x_c) + \sin\psi_k(y_k - y_c)\right)\right) - \frac{\partial U}{\partial \psi_k} \tag{6.5.8}$$

In (6.5.8), $\omega_0 = v_0/R$ and ζ_1 is a constant control gain, to be designed. The potential function $U(\psi)$ in (6.5.8) is used to force the vehicles to uniformly distribute on the circumference of radius R around the virtual centroid. The analytical expression of the potential function $U(\psi)$ is determined by the structure of the communication network Laplacian matrix; in the current application, it results:

$$U(\psi) = -\zeta_2 + \frac{\zeta_2}{3}\left(\cos(\psi_1 - \psi_2) + \cos(\psi_1 - \psi_3) + \cos(\psi_2 - \psi_3)\right) \tag{6.5.9}$$

where the gain $\zeta_2 > 0$ is a design constant. The speed of each vehicle, finally, is controlled through:

$$u_{k2} = \sqrt{v_0^2 + V_c^2 + 2v_0 V_c \cos(\psi_k - \theta_c)} \tag{6.5.10}$$

For a detailed description of the formation control strategy and the analysis of the convergence properties of the algorithm the reader is referred to [78].

6.5.2 Quasi-continuous boundary tracking steering control

The formation control presented in Section 6.5.1, provided the introduced hypothesis hold, makes the vehicles uniformly distribute on the circumference of a circle centred at the virtual centroid position. The movement of the virtual centroid occurs at constant speed V_c and is influenced by the steering law $\dot{\theta}_c(t) = u_c(t)$. This steering law is designed to solve the boundary tracking problem. If the centroid tightly tracks the desired contour, the vehicles will follow it, moving on a circle along the same contour.

The steering law $u_c(t)$ is designed through sliding mode techniques. Differently from Chapter 3, a probabilistic model of the feature is available here, and this allows to estimate the spatial gradient at the centroid position, through the strategy introduced in Section 6.4. Since the spatial gradient information is available, gradient based boundary tracking approaches can be applied. Here, the quasi-continuous sliding mode control approach, originally proposed in [145], has been applied for boundary tracking purposes.

The sliding variable is defined as:

$$\sigma(t) := \hat{\gamma}(x_c(t), y_c(t)) - \gamma_{ref}(t) \tag{6.5.11}$$

where $\hat{\gamma}(x_c(t), y_c(t))$ is the estimated measurement at the virtual centroid's position. The measurement at the virtual centroid position is estimated through an averaging step from the measurements at the positions of the three vehicles:

$$\hat{\gamma}(x_c(t), y_c(t)) = \frac{\sum_{k=1}^{3} \gamma(x_k(t), y_k(t))}{3} \tag{6.5.12}$$

where $\gamma(x_k(t), y_k(t))$ is the measurement at the location of the k-the vehicle. The sliding surface is defined as the locus where $\sigma(t) = 0$. The time varying reference trajectory $\gamma_{ref}(t)$ in (6.5.11) is defined as in (5.4.1). Specifically, the reference trajectory is defined as a function of the sliding variable $\sigma(t)$.

Similarly to earlier cases, it can be shown that the first derivative of the sliding surface has the following expression:

$$\dot{\sigma}(t) = V_c \|\nabla\gamma\| \sin(\theta_c - \phi) - \dot{\gamma}_{ref}(t), \tag{6.5.13}$$

while the second derivative can be expressed in the following generic form:

$$\begin{aligned} \ddot{\sigma}(t) &= \underbrace{V_c \sin(\theta_c - \phi)\frac{d\|\nabla\gamma\|}{dt} - \ddot{\gamma}_{ref}(t)}_{\xi(t)} - \underbrace{V_c \|\nabla\gamma\| \cos(\theta_c - \phi)}_{b(t)}(\dot{\theta}_c - \dot{\phi}) \\ &= \xi(t) + b(t)(u_c - \dot{\phi}) \end{aligned} \tag{6.5.14}$$

where ϕ is the angle between the tangent to the tracked contour and the horizontal direction (see Fig. 3.2.1).

The term dependent on the control action $u_c(t)$ appears in (6.5.14) for the first time, hence the relative degree of the system between the sliding variable $\sigma(t)$ and the steering control is two. For a relative degree two system, the *quasi-continuous sliding mode control* results [145]:

$$u_c = -\alpha \frac{\dot{\sigma} + |\sigma|^{1/2}\text{sign}(\sigma)}{|\dot{\sigma}| + |\sigma|^{1/2}} \tag{6.5.15}$$

where $\alpha > 0$ is a design constant. Applying this steering control to the centroid movement, a second order sliding mode, characterized by $\sigma(t) = \dot{\sigma}(t) = 0$ can be obtained in finite time, achieving boundary tracking.

In order to apply the control action in (6.5.15), knowledge of $\dot{\sigma}$ is required, whose expression is given in (6.5.13). The only unknown quantity in (6.5.13) is the gradient of the spatial feature $\nabla\gamma$. Following the procedure presented in Section 6.4, however, an estimate of the gradient at the centroid's position can be obtained, and hence $\dot{\sigma}$ can be estimated from (6.5.13). Consequently the control law in (6.5.15) is implementable.

The designed quasi-continuous sliding mode steering law $u_c(t)$ only affects the movement of the formation's virtual centroid, which achieves boundary tracking. Through the formation control in Section 6.5.1, the vehicles uniformly distribute themselves in a triangular formation around the position of the moving centroid and follow its movement, collecting measurements around and on the tracked contour itself.

It has been mentioned how collecting measurements in a certain region around the tracked contour may be beneficial from an oceanographic perspective. Therefore, the structure of the desired formation, and specifically the distance between the vehicles, can be modified to influence the degree at which the vehicles' measurements are distributed around the tracked contour. Specifically, if sampling of a wide area around the tracked contour is required, a greater value of the distance between the

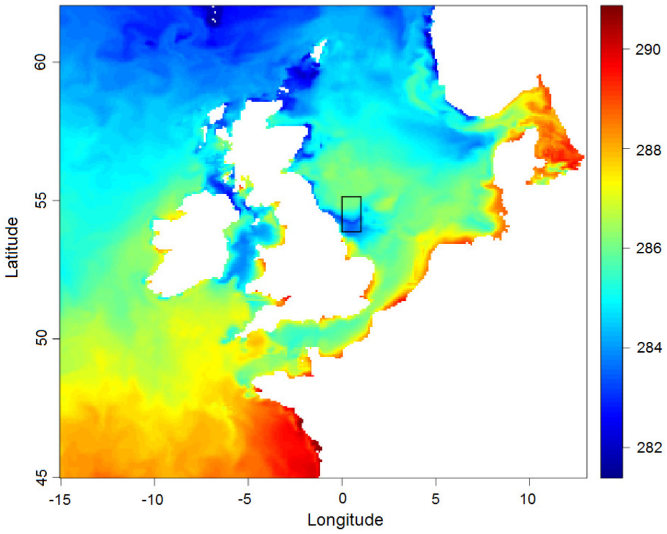


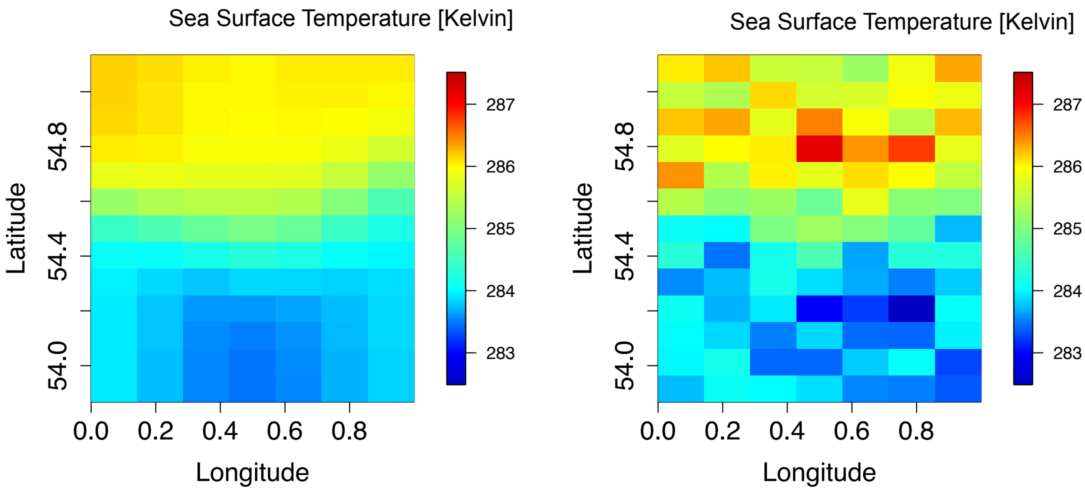
Figure 6.5.2: FOAM predicted sea-surface temperature

vehicles, and thus of the radius of the circumference along which they distribute, should be chosen.

6.6 Approach validation on a realistic dataset

6.6.1 Realistic dataset introduction

The dataset relating to the sea-surface temperature in the European North-West continental shelf, obtained from the Met Office FOAM model [90], is shown in Fig. 6.5.2. The Met Office FOAM model, and the available dataset, have been introduced in Section 3.5. The area highlighted with a black box, which extends $0° - 1°$ E, $53.9° - 55.1°$ N, is considered here. The output of the Met Office oceanic model has a resolution of approximately 2.5 km in both the North-South and the East-West directions, which means that the considered area is characterized through 1800 grid points. For the definition of the preliminary information shown in Fig. 6.6.1a, only a subset of the available data set is used. In particular, the resolution of the preliminary dataset is coarsened to approximately 11 km, and hence a training data set $\mathcal{P}$ composed of 91 observations is used. Additionally, the observations in the training data set $\mathcal{P}$ are artificially affected by adding noise to the model output, in order to reproduce the uncertain knowledge of the feature. Particularly, Gaussian



(a) Noise free preliminary information

(b) Preliminary information $\mathcal{P}$

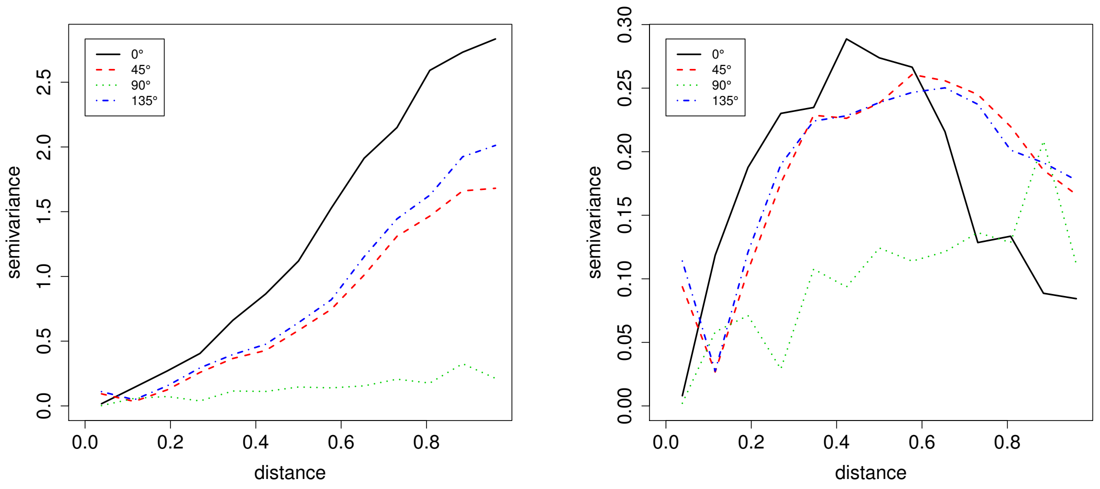
Figure 6.6.1: European North-West continental shelf sea-surface temperature - preliminary information

white noise with zero mean and constant variance, as in (6.3.7), is introduced. Being the overall RMSE in the prediction of the sea surface temperature through the FOAM model of the order of $0.4°C$ [146], the variance of the deliberately introduced Gaussian white noise is set to $\sigma_n^2 = 0.16$. The original subset of the FOAM model output used to build the preliminary information is shown in Fig. 6.6.1a, while the noise added preliminary information is shown in Fig. 6.6.1b. As only a subset of the FOAM model is used to define the preliminary information set, the remaining available observations can be partly or fully used in a model validation procedure, following the approach introduced in Section 6.3.3.

6.6.2 Isotropy analysis

A dataset is said to be isotropic if the variations are a function of solely the distance between points, and not of the direction of movement [128]. The GP model fitted to an isotropic dataset is simplified, as the covariance function results univariate; consequently, it is desirable to work with isotropic datasets.

The dataset considered here is clearly non isotropic, as visible from Fig. 6.6.1a, because the sea surface temperature is highly varying with latitude, while it is almost constant with longitude. In order to simplify the model fitting procedure,



(a) Directional variograms

(b) Directional variograms with a 1st order trend

Figure 6.6.2: Directional variograms. The anisotropy of the considered dataset is corrected through the introduction of a 1st order global trend

the anisotropy can be corrected through the introduction of a trend, which defines the structure of the mean function of the fitted model [147]. The effectiveness of an anisotropy correction procedure is often assessed though the directional variograms. An empirical variogram is a graph that describes the spatial correlation between the data as a function of the separation distance [148]; the empirical variogram, often identified as the sample variogram, is defined as:

$$\hat{\lambda}(u) = \frac{1}{2|N_u|} \sum_{(i,j)\in N_u} (z_i - z_j)^2 \tag{6.6.1}$$

where the differences between observations z_i and z_j, belonging to a distance class N_u, are averaged. The distance class is the set of couple of observations located within a certain distance from each other. The directional variograms are obtained similarly by taking into account not only the distance between data points, but also the direction of movement. The directional variograms of the preliminary dataset are shown in Fig. 6.6.2a, where the movement along the four cardinal directions is considered. The value of the variogram at null distance is called the nugget and it describes the uncertainty associated with the data set, while the value at which the variogram stabilizes for big distances is called the sill. If the directional variograms

Parameter	β_0	β_1	β_2	σ^2_{GP}	ϕ	σ^2_n
$\hat{\vartheta}_{MLE,SE}$	283.5317	−0.0060	0.0450	0.1977	12.2381	0.1191
$\hat{\vartheta}_{MLE,M}$	283.5882	−0.0061	0.0430	0.2202	6.0653	0.1089

Table 6.6.1: Sea surface temperature probabilistic model - distribution hyperparameters obtained from the maximum likelihood estimation for different covariance function choices

Metric	Maximized likelihood $\hat{L}$	BIC	AIC
SE distribution	−53.65	134.4	119.3
Matérn distribution	−54.44	136	120.9

Table 6.6.2: Sea surface temperature probabilistic model - validation metrics for the comparison of the two models fitted through maximum likelihood

have similar nugget, sill, and similar behaviour for intermediate distances, then the dataset can be considered isotropic. The directional variograms in Fig. 6.6.2a, however, are significantly different, especially in terms of the sill. Consequently, an anisotropy correction step is considered necessary. Fig. 6.6.2b shows the directional variograms obtained by fitting a 1st order trend to the dataset; in this way, the global trend of the dataset is modelled as a first order polynomial of the coordinates. The obtained variograms are still not perfectly equal, but their similarity in terms of behaviour and sill is enhanced. As no significant improvement is obtained with the introduction of a 2nd order trend, a 1st order trend is introduced in order to correct the dataset anisotropy; this trend constitutes the mean of the fitted GP model.

6.6.3 Probabilistic model fitting

In this section, different choices for the structure of the probabilistic model to be fitted to the preliminary dataset introduced in Section 6.6.1 are made and the different fitting methodologies detailed in Section 6.3.1 and Section 6.3.2 are applied. The best fitting model is chosen through a detailed validation procedure. The fitted models are characterized by the same global trend structure, as a first order polynomial of the coordinates is used as the mean function $\mu(\boldsymbol{x})$. This choice has been dictated by the isotropy correction step described in Section 6.6.2.

As a first choice, a SE covariance function as in (6.3.2) is considered. With these choices, the GP has the following expression:

$$\chi(x,y) = \underbrace{\beta_0 + \beta_1 x + \beta_2 y}_{\mu(x,y)} + Z(x,y) + \epsilon(x,y) \tag{6.6.2}$$

where $Z(x,y)$ is a zero-mean GP characterized as [128]:

$$Z(x,y) \sim GP\left(0, \sigma_{GP}^2 e^{-\frac{h^2}{\phi^2}}\right) \tag{6.6.3}$$

The nugget $\epsilon(x,y)$ is modelled as in (6.3.7).

As mentioned in Section 6.3, the SE covariance function is sometimes criticized for being excessively smooth [129]. Consequently, the Matérn covariance function with $\nu = \frac{3}{2}$, defined as in (6.3.4), is considered obtaining the following zero-mean GP:

$$Z(x,y) \sim GP\left(0, \sigma_{GP}^2 \left(1 + \sqrt{3}\frac{h}{\phi}\right) e^{-\sqrt{3}\frac{h}{\phi}}\right) \tag{6.6.4}$$

In both cases, the hyperparameter vector associated with the GP model is defined as $\boldsymbol{\vartheta} := (\beta_0, \beta_1, \beta_2, \sigma_{GP}^2, \phi^2, \sigma_n^2)$.

The two different probabilistic models are fitted through the maximum likelihood approach presented in Section 6.3.1. The obtained values of the hyperparameters vectors $\hat{\boldsymbol{\vartheta}}_{MLE,SE}$ and $\hat{\boldsymbol{\vartheta}}_{MLE,M}$, corresponding respectively to the SE and the Matérn distributions, are given in Table 6.6.1. These have been obtained through the `likfit` routine in the statistic friendly software 'R' [149]. The two statistical models are characterized by a similar set of hyperparameters.

In order to choose the preferred model, the first set of validation metrics introduced in Section 6.3.3 is considered. In particular, the maximized likelihood, the BIC and the AIC of the two models are compared. The obtained values are given in Table 6.6.2. It can be observed how the SE probabilistic model is characterized by a higher value of the maximized likelihood and a lower value of both BIC and AIC. Even if these differences are minor, from this preliminary evaluation the probabilistic model characterized by a first order mean function and a SE covariance function can be

Parameter	β_0	β_1	β_2	σ_{GP}^2	ϕ	σ_n^2
Mean	283.6613	−0.00574	0.04117	0.5455	19.0078	0.2823
Median	283.6624	−0.00550	0.04076	0.4742	18.6842	0.2684
Mode	283.5863	−0.00621	0.04376	0.3578	15.0000	0.2263

Table 6.6.3: Sea surface temperature probabilistic model - Bayesian model hyperparameters posterior distribution

judged to be the one better fitting the available preliminary dataset.

A third probabilistic model is fitted through the Bayesian estimation technique described in Section 6.3.2. For the definition of a Bayesian model, the available prior information needs to be specified. In particular, the structure of the model, in terms of mean and covariance functions, should be chosen: a first order mean function and a SE covariance function are chosen. Additionally, the prior distribution of the model hyperparameters should be defined. Specifically, a non informative flat prior distribution is used for the mean parameters β_i, while the variance σ_{GP}, the characteristic length scale ϕ and the nugget prior distributions are modelled as a reciprocal prior distribution.

The use of flat or reciprocal distributions is common in the definition of the prior information, as it reflects the typically low information availability [128], [130]. The posterior distribution of the hyperparameters is built using the Bayes theorem in (6.3.11), through the `krige.bayes` method in 'R'. Some details about the posterior probability distribution of the hyperparameters are given in Table 6.6.3, while the shape of the posterior distributions of the hyperparameters is shown in Fig. 6.6.3. It is visible how, despite the very low information contained in the chosen prior distribution, each hyperparameter is described by a normal-like posterior distribution.

6.6.4 Model validation

As a first validation test, the leave-one-out technique introduced in Section 6.3.3 is used to compare the two probabilistic models fitted through the maximum likelihood procedure. The tool used to compare the two models is the QQ plot. The obtained QQ plots are shown in Fig. 6.6.4, where the results relating to the model described

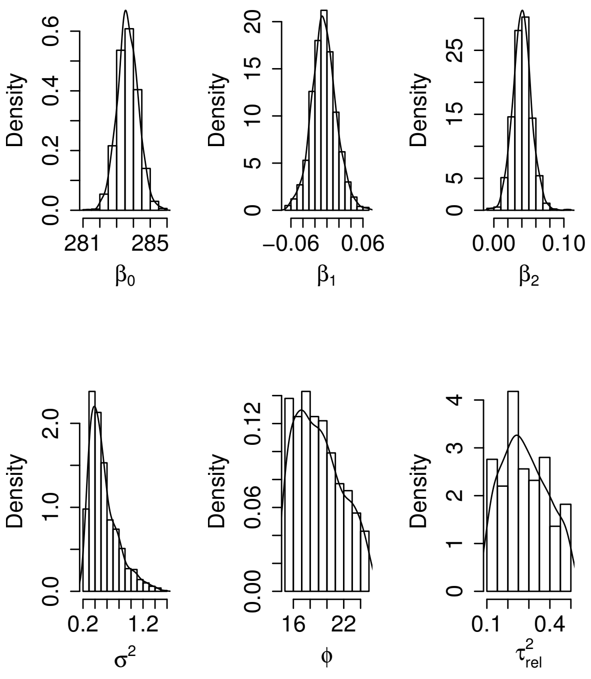


Figure 6.6.3: Hyperparameters posterior distribution for the model fitted through Bayesian estimation

by a SE covariance function are plotted as red circles, while those relating to the Matérn probability distribution are plotted as blue squares. Both QQ plots lie quite precisely on the $x = y$ line. This means that both models construct a probability distribution very similar to the actual distribution of the data in the preliminary dataset. As a consequence, in this case, the QQ plot does not distinguish between the two models and therefore it is not a useful tool for the choice of the best fitting model.

To continue the model validation, external validation metrics are computed, thanks to the availability of additional training data points obtained from the FOAM model output, which have not been used for the preliminary dataset definition. Particularly, four different validation sets are created, composed respectively of 128, 288, 544, and 1081 validation points.

For each of the three fitted models and for each of the four validation sets, each of the external validation metrics introduced in Section 6.3.3 is computed. For the

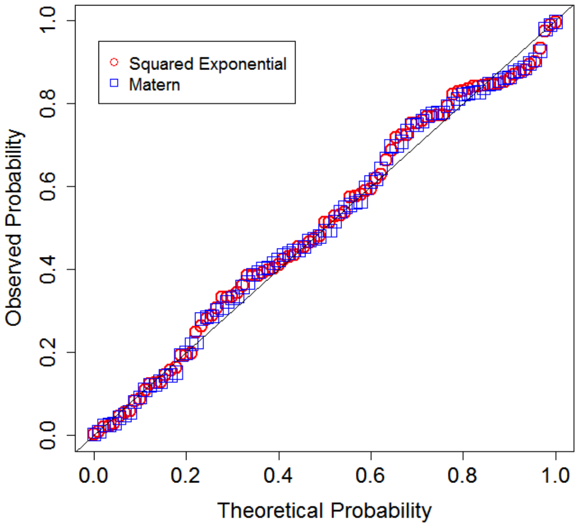


Figure 6.6.4: Comparison of the QQ plots obtained from the two probabilistic models fitted through maximum likelihood.

comparison of the metrics, a graphical approach is chosen. The results are shown in Fig. 6.6.5. In each of the subfigures, the x axis corresponds to the size of the validation set, while the validation metric is shown on the y axis. Additionally, the results obtained from the maximum likelihood SE model are represented as red circles, the results from the maximum likelihood Matérn distribution model are represented as blue squares and the results from the Bayesian SE distribution model are represented as green triangles. Fig. 6.6.5a shows the RMSE evaluated for every probabilistic model, for each of the validation sets. In each case, the two models characterized by a SE covariance function, fitted through the maximum likelihood and the Bayesian approaches respectively, perform better, while the RMSE obtained from the Matérn distribution is always higher, revealing, once again, lower performances. Fig. 6.6.5b shows the evaluation of the exceedence probability as in (6.3.19), where the black line indicates the true value obtained from the whole dataset through (6.3.18). In this case, the most precise estimation is obtained from the SE distribution obtained through the maximum likelihood estimation procedure. Finally, Fig. 6.6.5c and Fig. 6.6.5d show the estimated mean and variance respectively. The difference between the various models in terms of the estimation of the mean is very small, but the SE distribution obtained through the maximum likelihood procedure is the one that

	RMSE				**Mean**			
m	128	288	544	1081	128	288	544	1081
SE	0.1968	0.2009	0.2116	0.2059	284.72	284.77	284.85	284.81
Matérn	0.2103	0.2089	0.2217	0.2139	284.72	284.77	284.85	284.81
Bayesian	0.1807	0.1890	0.2010	0.1950	284.73	284.77	284.86	284.81
Real	-				284.81			
	Variance				**Exceedence Probability**			
m	128	288	544	1081	128	288	544	1081
SE	1.0466	1.0776	1.0860	1.0839	0.5234	0.5382	0.5478	0.5430
Matérn	1.0524	1.0803	1.0920	1.0870	0.4922	0.5208	0.5257	0.5291
Bayesian	1.0641	1.0918	1.0973	1.0977	0.5391	0.5521	0.5735	0.5689
Real	1.0118				0.54			

Table 6.6.4: Validation metrics (m is the number of validation points)

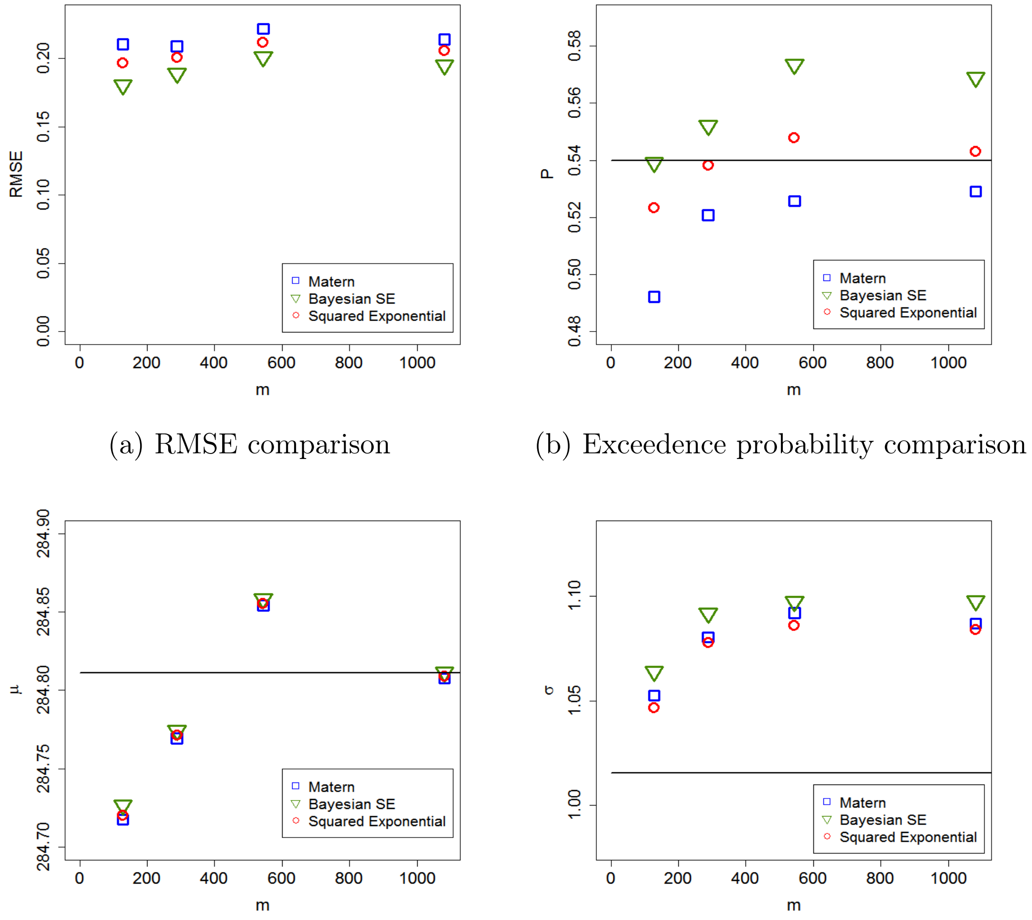
better estimates the distribution variance. The values of the validation parameters, graphically shown in Fig. 6.6.5, are collected in Table 6.6.4. In conclusion, the preferred model is the one obtained from the maximum likelihood procedure and characterized by a SE covariance function.

6.6.5 Formation boundary tracking results

The formation control strategy presented in Section 6.5.1 has been applied to a set of $n = 3$ vehicles. Each vehicle is modelled as in (6.5.3) and the control actions modifying the angular and forward velocities are defined as in (6.5.7) and (6.5.10) respectively. The approach is tested in a Matlab/Simulink environment (R2016b), with an Euler integration scheme with a step size of 1 minute.

The desired distance between the vehicles is $d = 200$ m and the controller parameters are chosen as $\zeta_1 = 0.8$ and $\zeta_2 = 2.8$, which satisfy the constraint $\zeta_1, \zeta_2 > 0$ introduced in [78]. Particularly, having $\zeta_2 > 0$ guarantees that the headings of the vehicles in the formation are synchronized modulo $\frac{2\pi}{3}$ [78]. The control actions are updated every 1 minute and a total simulation time of 35 hours is undertaken.

The virtual centroid is controlled through the quasi-continuous sliding mode boundary tracking algorithm described in Section 6.5.2, with $\alpha = 1.5$. This parameter has been chosen, as suggested in [145], through a simulation based tuning procedure, rather than by estimating redundantly large bounds C, K_m and K_M in (3.3.8).



(a) RMSE comparison

(b) Exceedence probability comparison

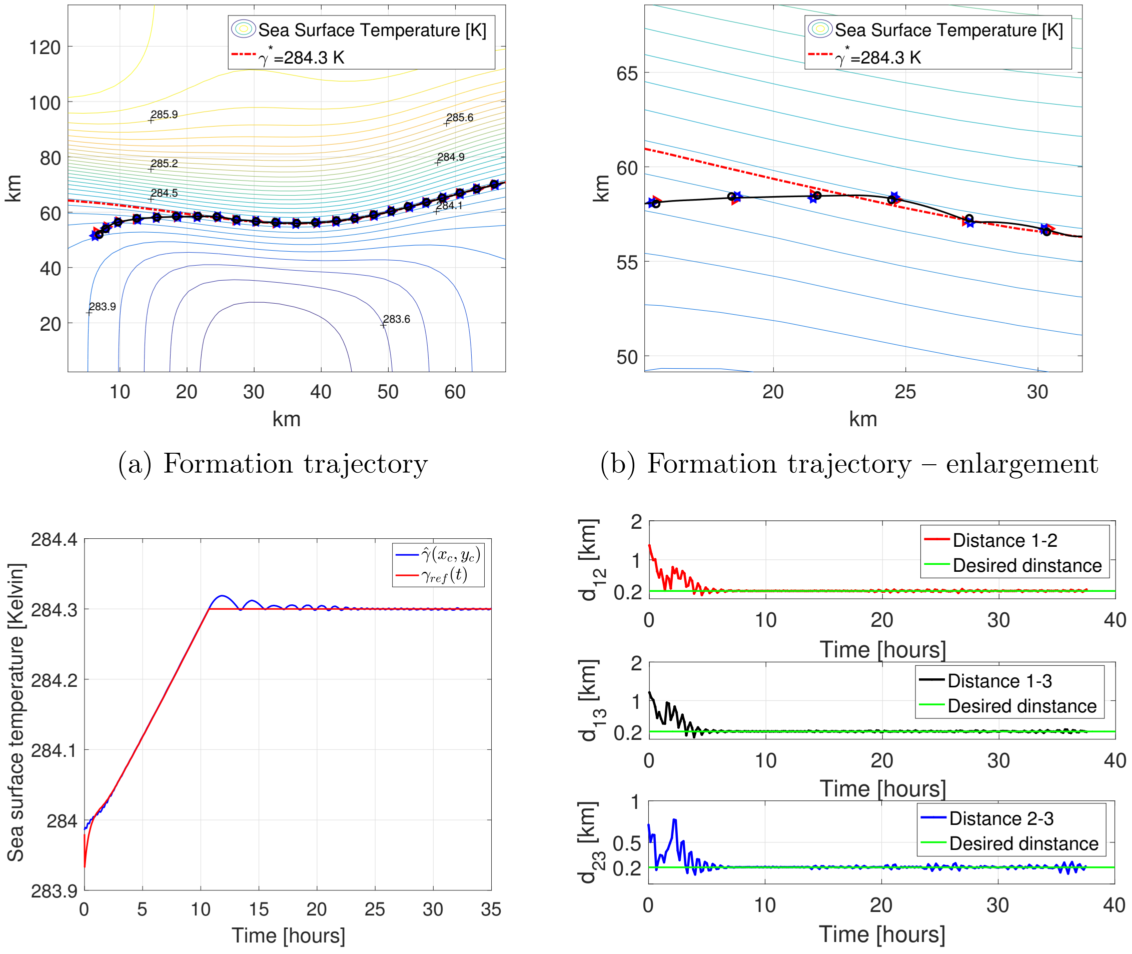
(c) Distribution mean comparison

(d) Distribution variance comparison

Figure 6.6.5: External validation metrics.

The tracked value of the sea surface temperature is chosen as $\gamma^* = 284.3$ Kelvin, which corresponds to a temperature contour in the tidal mixing front area, while $\gamma_{ref}(t)$ is determined through (5.4.13), with $K = 3$ and $\delta = 0.01$. Finally, the measurements at the virtual centroid position are estimated through the averaging step in (6.5.12), where the measurements collected by the vehicles in the real world are obtained from the full noise-free data set obtained from the Met Office FOAM model in Fig. 6.5.2. The gradient information required for computing $\dot{\sigma}$ in (6.5.13) is estimated following the procedure discussed in Section 6.4.

The results obtained are shown in Fig. 6.6.6: Fig. 6.6.6a shows the contour plot of the sea surface temperature in the considered area, and the tracked contour is highlighted in red. The vehicles move from left to right following the virtual centroid,



(a) Formation trajectory

(b) Formation trajectory – enlargement

(c) Centroid's measurements and reference trajectory

(d) Distances between vehicles

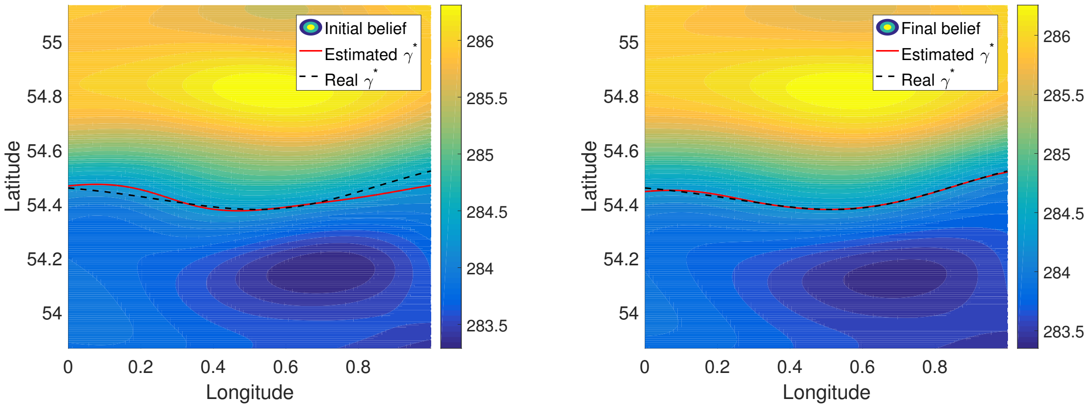
Figure 6.6.6: Sea-surface temperature front tracking results

whose trajectory is shown in black. The position of the vehicles in the formation at discrete time instants is shown and it is possible to observe in the enlargement in Fig. 6.6.6b how these successfully distribute on the circumference around the centroid. The estimated temperature value at the centroid position, obtained through (6.5.12), and the reference trajectory $\gamma_{ref}(t)$ are shown in Fig. 6.6.6c. The measurements at the centroid position stabilize at the tracked value γ^* in finite time, achieving boundary tracking. Finally, the distances between the vehicles are shown in Fig. 6.6.6d, in order to confirm the finite time achievement of the equilateral triangular formation, with $d = 200$ m.

6.6.6 Model update results

Exploiting the fitted probabilistic model, the belief is created, with a resolution of approximately 2.5 km - five time higher than that of the preliminary dataset and equal to the resolution of the original FOAM model output. The predictions are obtained in R through the global neighbourhood spatial interpolation method described in (6.3.10). During the simulation, the belief is updated when new real measurements taken by the vehicles' sensors become available: 'real' measurements are taken, in simulation, from the complete, noise free, FOAM model data set. At each update, a new model is fitted through maximum likelihood and spatial prediction is performed. Specifically, each of the hyperparameters in Table 6.6.1 is updated, except from the nugget parameter σ_n^2. The enhancement of the model accuracy through this periodic update procedure is assessed through the RMSE, evaluated on the 2.5 km resolution grid. The values obtained at the beginning and at the end of the simulation are given in Table 6.6.5, from which an 11% reduction in the RMSE can be appreciated, highlighting the efficacy of the iterative probabilistic learning technique. In addition, the initial and the final belief can be compared from Fig. 6.6.7. The estimation of the tracked contour, characterized by $\gamma = \gamma^*$, results highly improved by the iterative model update procedure. The initial difference between the estimated contour, highlighted in red, and the real contour, shown in black, which can be observed in Fig. 6.6.7a, is almost completely cancelled by the final fitted model. In Fig. 6.6.7b, in fact, the actual and the estimated contours are almost perfectly superimposed.

The model update approach used in this thesis requires fitting a new model through the maximum likelihood method every time the preliminary information set is enriched with real world measurement. This approach has the limitation of losing any knowledge about the spatial field gained through the model fitting when a new model is fitted. The new model, in fact, is fitted using the enriched preliminary information only, not exploiting any output of the previous model fitting step. A possibly more efficient approach is Bayesian learning, which is based on the Bayesian



(a) Initial belief

(b) Final belief

Figure 6.6.7: Comparison of the initial and final belief.

	Initial model	**Final model**
RMSE	0.2194	0.1955

Table 6.6.5: Model update efficacy - the Root Mean Square Errors obtained from the initial and the final beliefs are compared

model fitting method. Through Bayesian learning, the first prior distribution is defined as non-informative, and an initial posterior distribution is obtained. When a model refitting is necessary, the posterior distribution obtained from the previous iteration is used as the prior for the model fitting step. In this way, the knowledge about the spatial field obtained from each model fitting is used to obtain models characterized by increasing accuracy.

6.7 Concluding remarks

In this chapter, a probabilistic learning boundary tracking approach is proposed. The approach relies on the availability of some forecast information about the spatial feature of interest, which is used to fit a probabilistic model of the feature itself. The model is used in order to estimate the spatial gradient of the feature, which is useful when solving a boundary tracking control problem. The proposed probabilistic boundary tracking method is applied to a formation of autonomous vehicles, whose virtual centroid is controlled through a quasi-continuous sliding mode steering control in order to tightly follow the tracked boundary. In order to design the steering control, the model-based gradient estimate is exploited. The measurements collected by the vehicles in the formation are used to enhance the accuracy of the fitted surrogate model, and hence of the estimated gradient. The approach is tested in simulations using realistic data relating to the sea-surface temperature in the European North-West continental shelf, obtained from the Met Office, UK. In this application, a constant sea surface temperature contour, belonging to a tidal mixing front area, is considered. The proposed strategy is proven successful, even in the presence of deliberately introduced noise.

The work presented in this chapter highlights how the available preliminary information has the potential of being exploited when mapping spatial features with autonomous vehicles. Additionally, the introduced probabilistic learning strategy demonstrates how measurements collected through autonomous vehicles can be used in order to enhance the accuracy of already available computer models of environmental features. In this contest, an initial synthetic assessment of the impact of observations collected by autonomous vehicles on the prediction capability of computer based models has been performed in collaboration with the Met Office. Specifically, the impact of assimilating virtual observations of a tidal mixing front into the 7km Atlantic Margin Model (AMM7) has been evaluated, obtaining promising results.

Chapter 7

Experimental validation of the suboptimal sliding mode boundary tracking algorithm

7.1 Introduction

This chapter describes the experimental validation of the boundary tracking algorithm presented in Chapter 3 with a marine surface vehicle, C-Enduro, owned by National Oceanographic Centre (NOC), Southampton, UK. The work presented in this chapter has been supported by a grant on the AAOSN project, a Small Business Research Initiative (SBRI) competition run by the Natural Environment Research Council (NERC) and the Defence Science and Technology Laboratory (DSTL).

The aim of the experimental works is to track a contour of constant depth in the seabed with the marine surface vehicle C-Enduro. The vehicle is autonomously navigated using the boundary tracking algorithm introduced in Chapter 3. The autonomous vehicle is equipped with a suitable sensor to measure the water depth at its current position. The bathymetry tracking problem is a representative example, amongst many other applications which may be tackled applying the boundary tracking algorithm. This specific example has been chosen because of its harmless nature and its persistent availability. The proposed approach can reduce the time and effort involved in determining the boundary of oceanic features when compared with conventional lawnmower-type approaches.

Only a few of the boundary tracking algorithms which consider autonomous vehicles have been experimentally validated in the published literature. The algorithm originally proposed in [150] has been experimentally validated in [42] using wheeled vehicles to track virtual geometric boundaries. The bang-bang contour tracking algorithm proposed in [41] is experimentally validated in [34]. In this work the boundary is created through coloured tape on the floor and the wheeled autonomous vehicle is equipped with a downward looking infra-red sensor, in order to distinguish the floor from the tape. In addition, the adaptive boundary mapping algorithm proposed in [20], where an autonomous surface vehicle is used to map the outflow plume of a nuclear power plant, is experimentally validated with a small surface autonomous system.

The philosophy underlying the work in this chapter is to equip the autonomous vehicle with the computation capability necessary to autonomously determine its trajectory, thus operating the vehicle in a fully autonomous way and avoiding the necessity of pre-planning.

The chapter is organized as follows: the autonomous vehicle C-Enduro is described in Section 7.2, while the developed Robot Operating System (ROS) network is characterized in Section 7.3. The results from a set of virtual trials are presented in Section 7.4, while the results from the pre-trials are presented in Section 7.5. Finally, the sea-trials results are described in Section 7.6, while Section 7.7 reports details about the development of an embedded system, implementing the proposed boundary tracking approach. Some concluding remarks are given in Section 7.8.

7.2 Autonomous vehicle C-Enduro

The autonomous surface vehicle C-Enduro, developed by Autonomous Surface Vehicles (ASV) Ltd, Portchester, UK has been used. This long endurance autonomous vehicle is shown in Fig. 7.2.1 and its main dimensions are indicated. The C-Enduro vehicle is capable of continuous operation for approximately three months, depending on its power configuration.

The typical sensor suit of the C-Enduro allows it to measure its position, attitude, Speed Over Ground (SOG) and heading. The Airmar 200WX sensor[1] provides position information in the global navigation satellite system (GNSS) format, Course Over Ground (COG), SOG, roll, pitch and yaw. For the experiments, the C-Enduro has been equipped with a single beam acoustic depth sensor, the GARMIN Intelliducer[2], which provides the depth below the keel of the vehicle up to a maximum of 275 m and with an accuracy of 0.05 m. The entire on-board instrumentation follows the National Marine Electronics Association (NMEA) 2000 standard. The data exchange has been realized over a ROS environment, whose development is discussed in Section 7.3.

The maximum speed of the C-Enduro is 7 knots (approximately 3.5 m/s). For C-Enduro, two modes of operation are possible, which are fixed speed and fixed thrust. In the *fixed speed* mode the vehicle modifies its level of thrust through a fast low level controller, with the aim of maintaining a constant value of SOG. In the *fixed thrust* mode, the thrust is kept at a constant level independently of the measured SOG. In order to apply the kinematic unicycle model introduced in (3.2.3), the fixed thrust mode of operation has been chosen. With reference to the kinematics in (3.2.3), $(x(t), y(t))$ is the vehicle's position (from the GNSS position information updates), $\theta(t)$ is the vehicle's heading from the Airmar 200WX sensor and $u(t)$ is the guidance command input influencing the vehicle's direction of movement.

[1] `http://www.airmar.com/weather-description.html?id=154`

[2] `https://buy.garmin.com/en-US/US/p/26510`

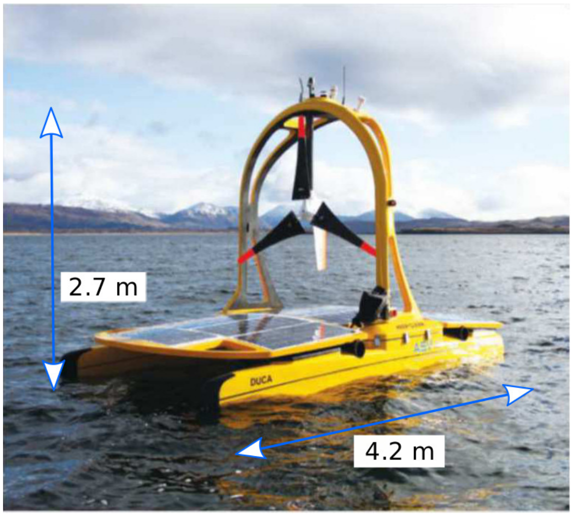


Figure 7.2.1: C-Enduro principal dimensions

The movement of the C-Enduro can be controlled either in the *Seek Position* mode or in the *Heading Hold* mode. In the *Seek Position* mode the vehicle is given a waypoint (WP), which is a coordinate (i.e. latitude and longitude) stamp, to be reached. In this mode of operation, the low level controllers of the vehicle, whose functionalities are developed by the producer, provide to adjust the direction of movement. Specifically, the direction of movement is corrected in order to continuously move along the shortest linear path between the vehicle's position and the sought WP. In other words, the heading of the vehicle is corrected in order to continuously point directly at the specified WP location. The movement is stopped as soon as the vehicle is within a predefined distance (the acquisition distance) from the WP. In the *Heading Hold* mode, instead, when the vehicle is commanded a desired heading, and hence a commanded direction of movement, the low level controllers keep the heading fixed until a new heading command is received.

7.3 ROS network

The communication infrastructure between the sensors, the boundary tracking guidance algorithm and the actuators is realized using the ROS framework [151]. A ROS network is composed of *nodes*, which are typically distinct processes, written in different programming languages, such as C++, Python or Matlab. Each node has a different designated functionality, but it may require to communicate with other nodes in order to obtain the required information. The ROS Master is the main node of the network, to which the remaining nodes need to register in order to be identified in the network with a unique name and to communicate with other registered nodes.

Nodes exchange information through structured data types, identified as ROS *messages*. Different types of ROS messages are available. Standard types, such as integers, floating points and Boolean variables, can be used independently or combined together in a structured way for the definition of customized messages, composed of several fields. The communication of a ROS message happens over a *topic*, which is a named bus used for exchanging a specific type of message. In order to communicate a message, nodes need to publish/subscribe to the corresponding topic. When the node is a publisher to a specific topic, it can communicate information, while it can access the required information by becoming a subscriber to the corresponding topic. It is possible to have multiple subscribers for a single topic, and a single node can publish and/or subscribe to multiple topics. The loose coupling inherent in the publish/subscribe design pattern ensures that the various nodes of the system can be individually developed. It allows quick reconfiguration of the system, as well as easy implementation of several distributed algorithms. Another type of communication, which implements request/reply interactions, is possible through ROS *services*. The server is the node offering a specific service, while the client is the node requesting the service. Once the client sends a request message to the server, it waits until a response message is received.

7.3.1 Implementation details

The ROS network developed for carrying out the experimental validation of the boundary tracking algorithm is shown in Fig. 7.3.1. The nodes in the network represent the main components of the experimental framework. These are the 'UoE Node', the 'ASVPilot' node, the 'Garmin' depth sensor node and the 'Initialization Server' node. The boundary tracking guidance strategy is implemented as the 'UoE

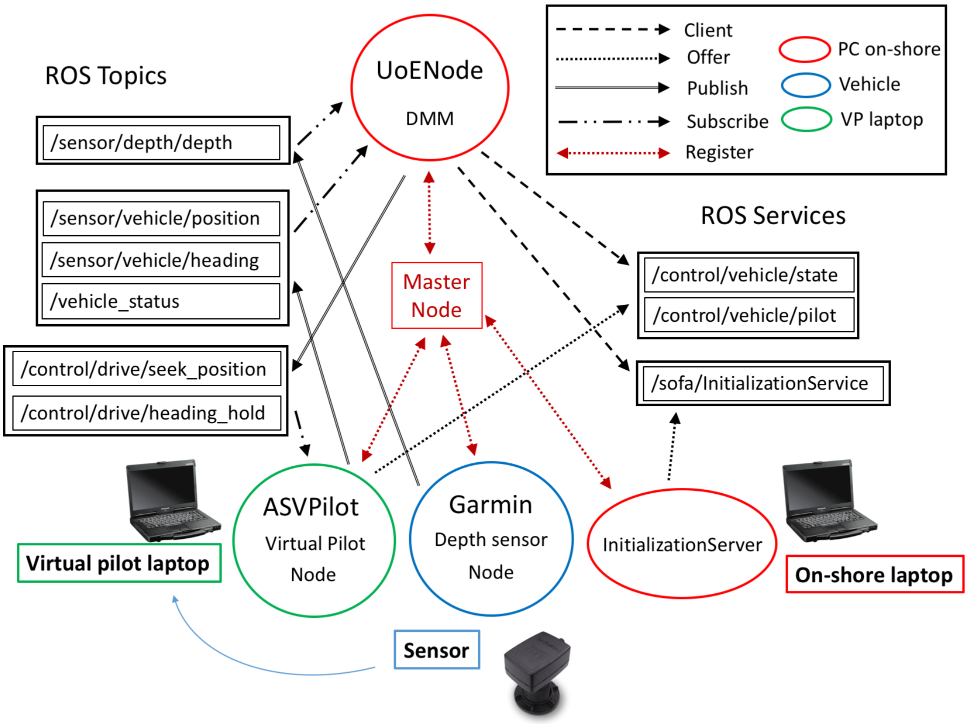


Figure 7.3.1: Robot Operating System (ROS) network

Node' on an on-shore laptop. The 'UoE Node' therefore designs the guidance command for the autonomous vehicle, which is a WP to be reached or a heading to be kept, depending on the mode of operation. The 'ASVPilot' node is the Virtual Pilot (VP) laptop, and serves as an interface between the 'UoE Node' and the vehicle's sensors and actuators. The 'Garmin' node represents the depth sensor installed on the vehicle, while the 'Initialization Server' node allows the user to set up the trial's parameters. Each node in the network needs to register with the ROS Master using its IP address. The location of each node in the ROS network is shown in Fig. 7.3.1: green nodes are located on the VP laptop, blue nodes are located on the vehicle, while red nodes are located on the industrial computer on-shore.

For the experimental validation of the boundary tracking algorithm, because of a safety requirement, a networked set up has been considered, with the 'UoE Node' and the VP located on two industrial computers on-shore. This set up, in fact, allows monitoring of all the parameters of the trial in real-time. Particularly, the 'UoE Node' has been implemented on an industrial laptop running Matlab R2015b and ROS.

Table 7.3.1: ROS topics

Topic	Details	Publisher	Subscriber
`/sensor/vehicle/position`	Position measured through the GNSS sensor	ASVPilot	UoENode
`/sensor/vehicle/heading`	Heading measured through a compass sensor	ASVPilot	UoENode
`/vehicle_status`	Vehicle state (Active or Pause)	ASVPilot	UoENode
`/sensor/depth/depth`	Measured depth	Garmin	UoENode
`/control/drive/seek_position`	Waypoint guidance control (Seek Position)	UoENode	ASVPilot
`/control/drive/heading_hold`	Heading guidance control (Heading Hold)	UoENode	ASVPilot

Table 7.3.2: ROS services

Service	Details	Server	Client
`/InitializationService`	Initialization of trial parameters	InitializationServer	UoENode
`/control/vehicle/state`	Vehicle state change request - response	ASVPilot	UoENode
`/control/vehicle/pilot`	Vehicle control request - response	ASVPilot	UoENode

The information exchange between the 'UoE Node' and the VP has been performed through an Ethernet cable, while the communication between the VP and the vehicle's sensors and actuators has been performed over a Wi-Fi link. The scheme in Fig. 7.3.1 also shows the ROS topics, the ROS services and the role of each node. Additional details about the publishers/subscribers to the ROS topics and the server/clients of the ROS services are given in Table 7.3.1 and Table 7.3.2 respectively.

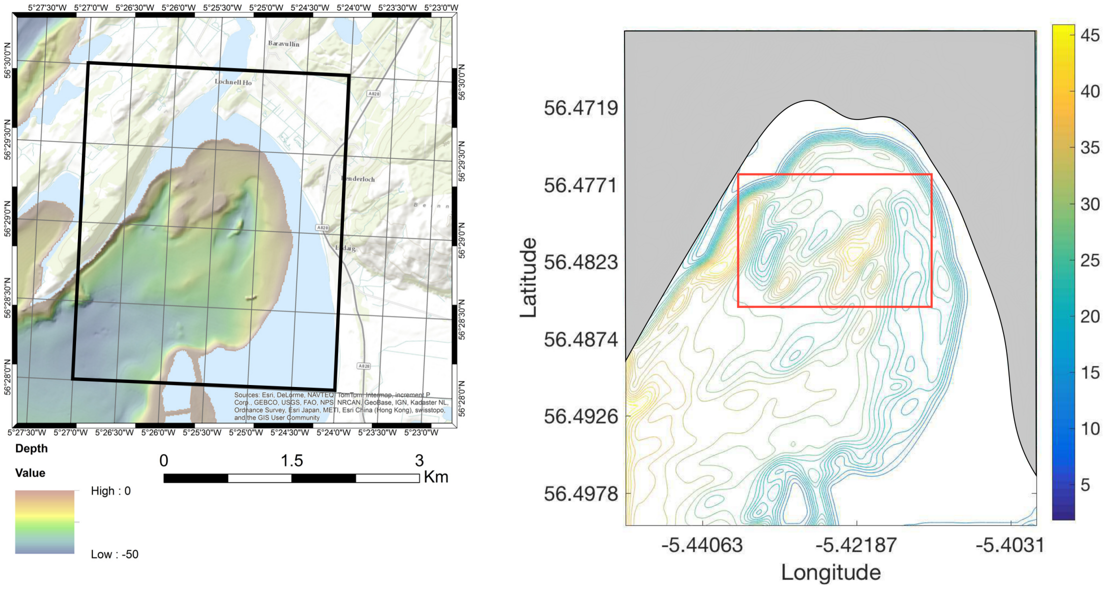
Depending on the mode of operation, the 'UoENode' publishes and subscribes to a set of the available ROS topics. In the *Heading Hold* mode, the 'UoENode' publishes the commanded heading on the `/control/drive/heading_hold` topic. In the *Heading Hold* mode, additionally, the 'UoENode' subscribes to the `sensor/vehicle/heading` topic, in order to receive information about the current heading of the vehicle, measured through the Airmar 200 WX sensor. In the *Seek Position* mode, the 'UoENode' publishes the commanded WP on the `/control/drive/seek_position` topic. In the *Seek Position* mode, additionally, the 'UoENode' subscribes to the `sensor/vehicle/position` topic, in order to receive information about the current position of the vehicle, measured through the GNSS sensor and used to determine the next WP to be published.

The 'ASVPilot' node publishes on the topics relating to the on board sensors, and in particular it publishes the position stamp measured through the GNSS sensor, the heading, measured through the Airmar 200 WX sensor, and the vehicle state.

Additionally, the ‘ASVPilot’ node subscribes to the topic corresponding to the manipulated variable, that is the commanded WP or heading.

The ‘Garmin’ node publishes solely on the `/sensor/depth/depth` topic, to which the ‘UoE Node’ subscribes in order to access the depth measurement at the vehicle’s position, necessary to design the guidance control. As and when a new reading is available from the sensor, the corresponding message is instantaneously updated. Despite this fast information update procedure, a ROS Timer regulates the scheduling of the ‘UoE Node’ subscriptions to the topics. Consequently, messages are only read at the timer interrupts, specifically every 5 seconds. Similarly, the ‘UoE Node’ publishes the designed control command every 5 seconds, even if the output of the decision making module is updated less frequently.

The ‘InitializationServer’ node offers the `/sofa/InitializationService`. The ‘UoE Node’, being a client to this service, can initialize the trial parameters, specifically the controller’s gains, the trial length, the level of thrust of the vehicle and the depth to be tracked. Two additional services are offered by the ‘ASVPilot’ node in order to modify the state of the vehicle and to release control of the vehicle. The ‘UoE Node’ is a client of these services. At the beginning of the trial, the ‘UoE Node’ requires the vehicle state to be set to ‘Active’ through the service `/control/vehicle/state`, in order to make the vehicle engage in movement, rather than remaining in the non controllable ‘Pause’ state. The ‘UoE Node’, which is a client to the service, tries to modify the state of the vehicle through a request message and then waits for the response message from the VP server to determine if the state of the vehicle has been successfully modified. In addition, the ‘UoE Node’ requires the `/control/vehicle/pilot` service in order to gain control of the vehicle, which could be controlled by another system (e.g. an operator). When the ‘UoE Node’ is in control, it can influence the movement of the vehicle by publishing on the topics relative to the guidance control. The ‘UoE Node’ tries to gain control of the vehicle through a request message. It then waits for the response message from the VP in order to determine if the control of the vehicle has been turned over to the ‘UoE Node’ itself. Finally, the ‘UoE Node’ is a subscriber to the `/vehicle_status`


Depth
Value
High : 0
Low : -50
0
1.5
3
Km
Sources: Esri, DeLorme, NAVTEQ, TomTom, Intermap, increment P Corp., GEBCO, USGS, FAO, NPS, NRCAN, GeoBase, IGN, Kadaster NL, Ordnance Survey, Esri Japan, METI, Esri China (Hong Kong), swisstopo, and the GIS User Community
56.4719
56.4771
56.4823
56.4874
56.4926
56.4978
Latitude
-5.44063
-5.42187
-5.4031
Longitude
45
40
35
30
25
20
15
10
5

(a) Ardmucknish Bay bathymetry image

(b) Image processed contour plot

Figure 7.3.2: Ardmucknish Bay - synthetic dataset generation

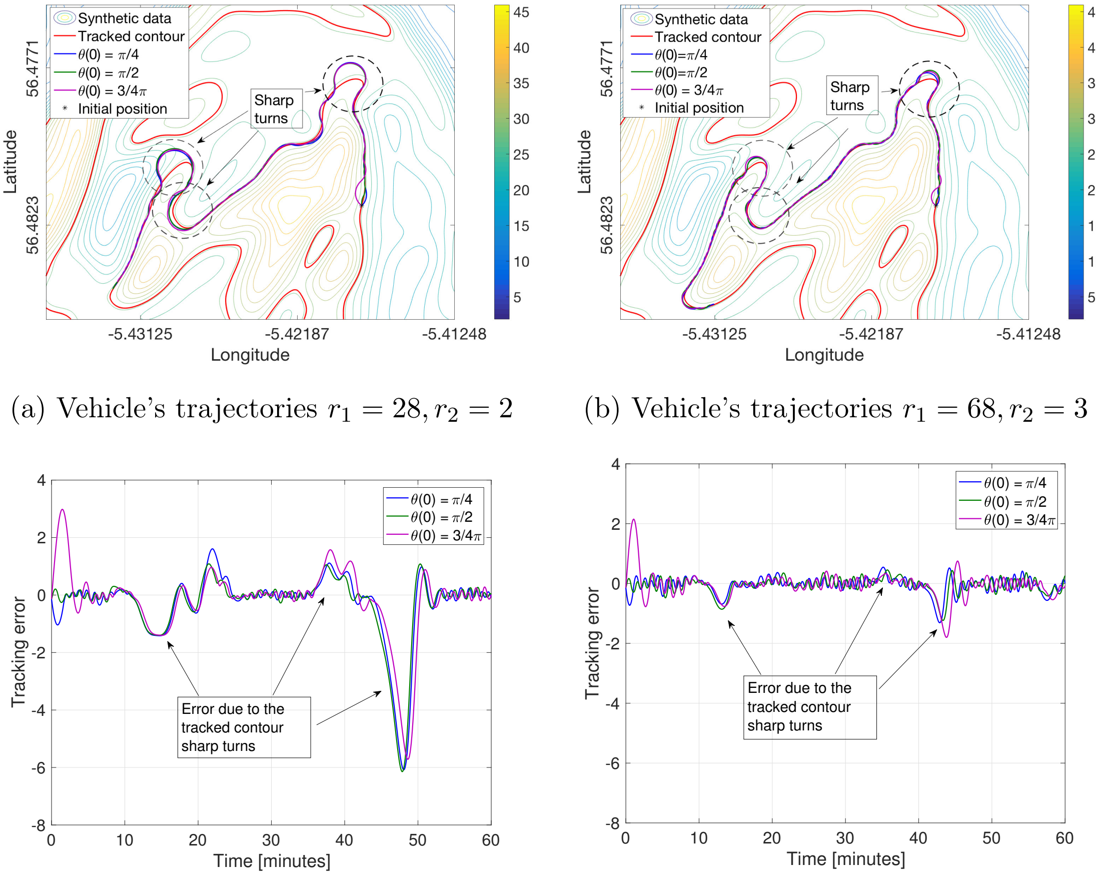
topic. By monitoring the values of the fields of this topic, the ‘UoE Node’ can determine if the vehicle is maintained in the ‘Active’ state and if the correct control strategy (*Heading Hold* or *Seek Position*) is being executed.

7.4 Virtual trials

Data based computer simulations have been used to test the efficacy of the suboptimal sliding mode boundary tracking algorithm to track a constant depth contour. The dataset has been obtained from an available bathymetric image of Ardmucknish Bay, Argyll and Bute, Scotland ($56^{\circ}28'58.1''N - 5^{\circ}25'54.5''W$), shown in Fig. 7.3.2a. The bathymetric contours in Fig. 7.3.2b are obtained by image processing of the region in the black box in Fig. 7.3.2a. These correspond to the gray scale colour levels, and represent scaled bathymetric depths. The area of operation for the vehicle in the simulations is highlighted with a red square in Fig. 7.3.2b.

The simulations have been performed in a Matlab/Simulink environment (version 2016b), with an Euler integration step with a step size of 15 seconds. In the simulations, the vehicle is modelled as a kinematic unicycle as in (3.2.3). The vehicle is required to track a bathymetric contour characterized by $\gamma^* = 28$. The initial location of the vehicle, $(x(0), y(0)) = (-5.4180, 56.4816)$, is chosen to be on the contour, with an initial measurement of $\gamma(x(0), y(0)) = 28$. The initial heading of the vehicle is fixed to $\theta(0) = \pi/4$, $\theta(0) = \pi/2$, and $\theta(0) = 3/4\pi$ respectively in three different simulations. These different heading choices are made to demonstrate the effectiveness of the methodology independently of the vehicle's initial orientation relative to the tracked contour. The vehicle is assumed to move at constant speed $V = 0.5$ m/s. A minimum turning radius of value $R_{min} = 60$ m is assumed for the vehicle. In addition, the movement of the vehicle is considered to be affected by a constant drift term, $||\Delta V|| = 0.22$ m/s, with orientation $\theta_{\Delta V} = 63°$. This constant drift term affects the vehicle's movement through ΔV_{xb} and ΔV_{yb} as defined in Section 3.3.3, and may be considered as a current. The exact values of the drift terms are assumed to be unknown for the purpose of the design of the suboptimal sliding mode based guidance law. These are assumed to be unknown functions with known upper bound values. For determining the control gains in (3.3.10), the bounds of the uncertainties are assumed to be $C = 25, K_m = 10$, and $K_M = 30$. In (3.3.10), the gains are set as $r_1 = 28$ and $r_2 = 2$ respectively. The choice of r_1, r_2 respects the assumed minimum turning radius of the vehicle (60 m), as discussed in (3.3.15).

The results obtained are shown in Fig. 7.4.1a and Fig. 7.4.1c. Fig. 7.4.1a shows the trajectories of the vehicle corresponding to the three different initial headings $\pi/2, \pi/4$ and $3\pi/4$ in blue, green, and purple respectively. The initial position of the vehicle is represented as a black star, and the tracked contour is highlighted in red. In each case, the vehicle robustly tracks the constant-depth contour, despite the effect of the unknown drift terms ΔV_{xb} and ΔV_{yb}. The dotted black circles in Fig. 7.4.1a highlight the sharpest turns to be performed by the vehicle in order to track the constant depth contour. Temporary loss of tracking, associated with these sharp contour features, is also visible in Fig. 7.4.1c, where the time evolution of the tracking error is shown. This temporary tracking loss is due to the effect of the control gains r_1, r_2, which limit the turning capability of the vehicle. Particularly,



(a) Vehicle's trajectories $r_1 = 28, r_2 = 2$

(b) Vehicle's trajectories $r_1 = 68, r_2 = 3$

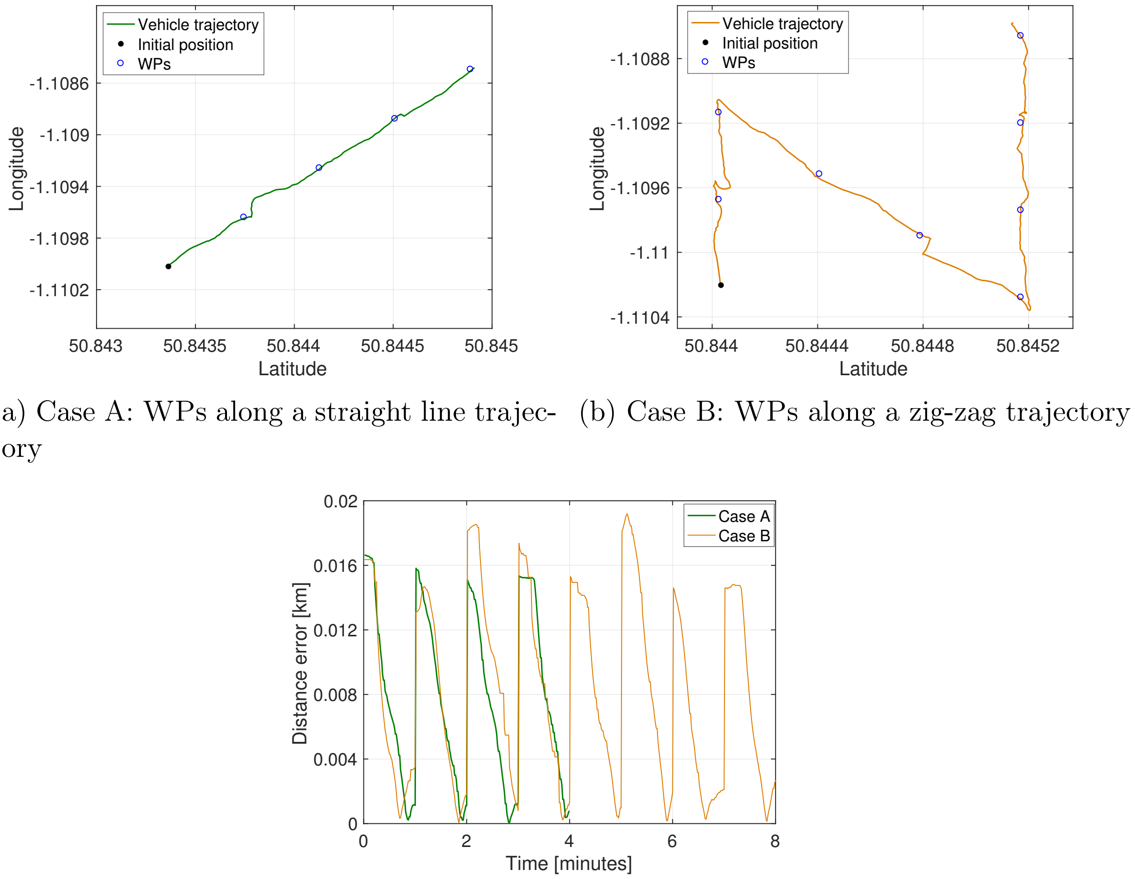
(c) Tracking errors $r_1 = 28, r_2 = 2$

(d) Tracking errors $r_1 = 68, r_2 = 3$

Figure 7.4.1: Synthetic boundary tracking results

the achievable minimum turning radius of the vehicle is higher than the required turning radius necessary to track the constant-depth contour. Tracking, however, is always recovered, after a few oscillations.

To reduce the limiting effect of the control gains, observed in the first set of simulations, the values of the control gains are increased to $r_1 = 68$, $r_2 = 3$, while keeping the remaining parameters and configuration unchanged. The results obtained with this design are shown in Fig. 7.4.1b and Fig. 7.4.1d. This choice of the control gains reduces the minimum turning radius of the vehicle to $R_{min} = 25$ m. Consequently, as it can be seen in Fig. 7.4.1b, tracking of the contour is improved when compared to the result in Fig. 7.4.1a, especially where the vehicle tracks the sharp features of the contour. This is confirmed by the evolution of the sliding variable, shown in Fig. 7.4.1d, which tightly stabilize around zero. Even in this case, tracking is temporarily



(a) Case A: WPs along a straight line trajectory

(b) Case B: WPs along a zig-zag trajectory

(c) Distance from the commanded WP

Figure 7.4.2: Pre-trials results - waypoint tracking in the *Seek Position* mode

lost in two occasions, but the amplitude of the oscillations and the recovery time are significantly reduced.

7.5 Pre-trials

7.5.1 Waypoint navigation

A set of pre-trials was conducted in Portsmouth Harbour, UK. The results from the pre-trials are shown in Fig. 7.4.2. In this set of pre-trials the vehicle was controlled in the *Seek Position* mode described in Section 7.2, and the commanded WPs were provided a-priori, similarly to conventional path planning control techniques. In designing the WPs, the distance between subsequent WPs should be correctly designed. When the vehicle reaches a neighbourhood of the commanded WP, whose amplitude is a configuration parameter, it stops and it may start drifting, depending on the water currents, until a new WP is commanded. In the pre-trials, the WPs are designed for the vehicle moving at a constant speed of ($V \approx 0.3$ m/s). The time interval between two subsequent guidance commands is set to 1 minute. The distance between subsequent WPs is chosen so that the vehicle is capable to reach each one before the next one is commanded. Fig. 7.4.2a shows the trajectory of the vehicle when requested to reach a set of WPs along a straight line. The vehicle successfully reaches a neighbourhood of each WP, whose amplitude of approximately 2 m is determined by the acquisition distance, and then starts moving towards the next one. Similar results, with a set of WPs along a zig-zag trajectory, are shown in Fig. 7.4.2b. The swift changes in the vehicle's direction of movement visible in Fig. 7.4.2b are due to the drifting of the vehicle once it is within the acquisition distance of the sought WP. As this behaviour is not desirable, the prediction of the distance travelled by the vehicle in the time interval between two subsequent guidance commands is of great importance when controlling the vehicle through the *Seek Position* mode. As a rule of thumb, in the following, this will be overestimated, in order to prevent the vehicle to reach within the acquisition distance of the sought WP. The time evolution of the distance between the vehicle's position and the commanded WP in the two pre-trials is shown in Fig. 7.4.2c.

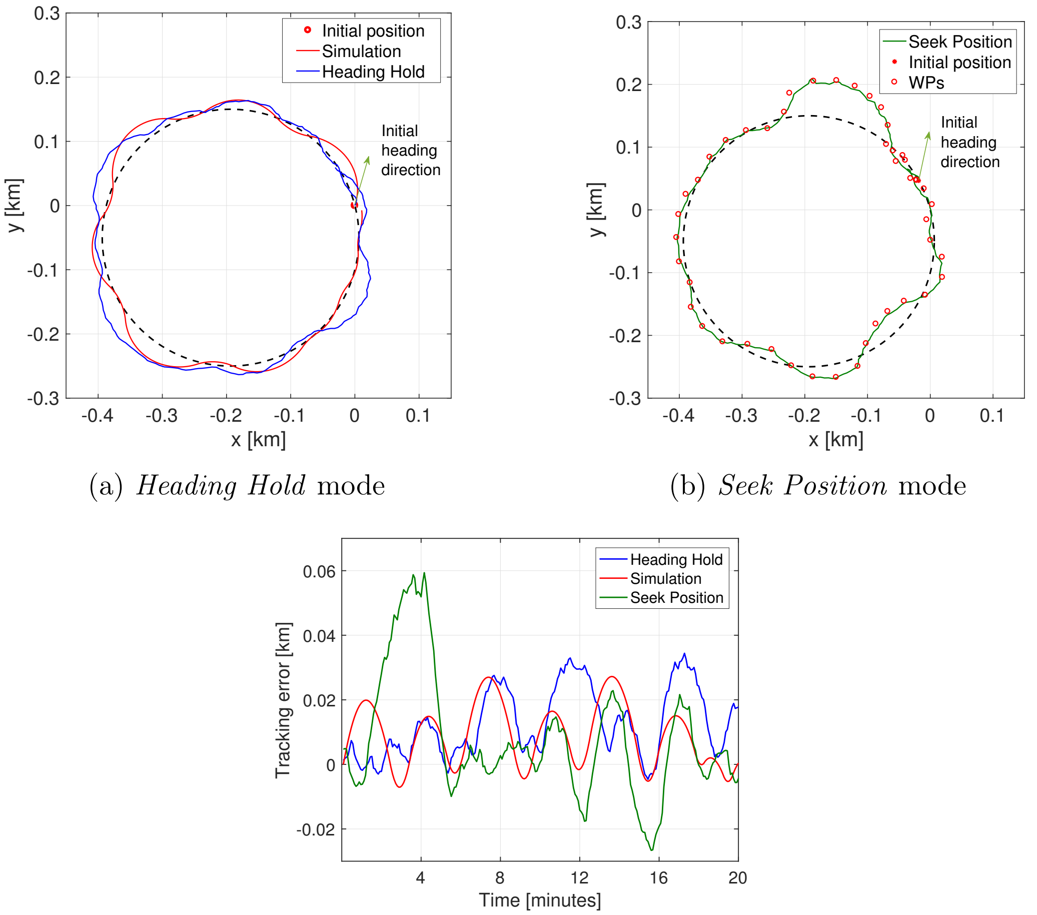
7.5.2 Boundary tracking pre-trials

In this set of pre-trials, the movement of the C-Enduro is controlled by the 'UoE Node' through the implemented boundary tracking algorithm. The vehicle is commanded to track a circular contour. In this case, the sliding variable is defined as:

$$\sigma(t) = r(t) - R \tag{7.5.1}$$

where R is the radius of the tracked circular contour, centred at (x_c, y_c), and

$$r(t) = \sqrt{(x(t) - x_c)^2 + (y(t) - y_c)^2} \tag{7.5.2}$$



(a) *Heading Hold* mode

(b) *Seek Position* mode

(c) Comparison of tracking error

Figure 7.5.1: Pre-Trials results - circumference tracking - comparison of the *Heading Hold* and the *Seek Position* navigation modes

is the virtual point measurement collected by the vehicle. The virtual measurement $r(t)$ is the computed range measurement between the known centre position (x_c, y_c) and the instantaneous position of the vehicle $(x(t), y(t))$, which is measured through the GNSS sensor.

In this set of pre-trials, results are obtained using both the *Heading Hold* and the *Seek Position* commanding modes for the vehicle, introduced in Section 7.2. The commanded quantities in the two modes of operation are the desired direction of movement and the WP respectively, updated every 30 seconds. The quality of the results from both modes is compared.

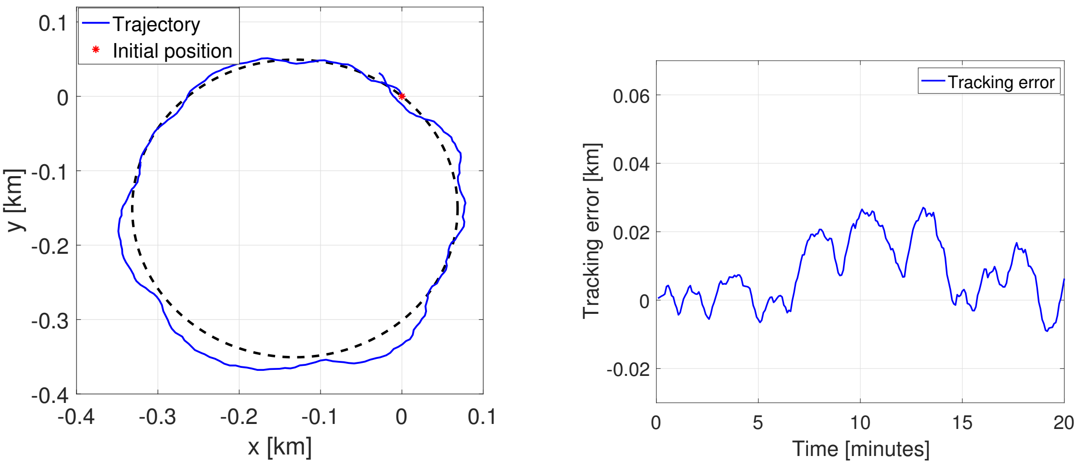
In the *Heading Hold* mode, the commanded quantity is the desired heading. As evident from the kinematic model in (3.2.3), the control input $u(t)$ is the rate of change of the heading. Hence, the desired heading is determined by integrating the control input $u(t)$, obtained by the suboptimal sliding mode guidance law in (3.3.10). The integration is performed using the Euler integration approach for a time period of 30 seconds.

In the *Seek Position* mode, a similar integration procedure is followed in order to obtain the desired WP. Specifically, an Euler integration step is in this case applied to each state of the kinematic model of the vehicle in (3.2.3). The commanded WP is designed accounting for the behaviour of the vehicle when within the acquisition distance. To avoid the sudden changes in the direction of movement of the vehicle observed in Section 7.5.1 and highlighted in Fig. 7.4.2, the length of the integration interval is set to 1.5 times the interval between two subsequent guidance control updates, specifically 45 seconds.
The radius of the tracked circle is set to $R = 200$ m. An additional assumption is that the initial position of the vehicle is on the circumference to be tracked. The thrust of the vehicle is set to 30%, which corresponds to a speed of approximately 1 m/s. The control gains in (3.3.10) are set to $r_1 = 28$, $r_2 = 2$. With these choices, the vehicle can track contours having a radius of curvature $R > R_{min} \approx 120$ m. This radius of curvature is significantly bigger than the practical minimum turning radius of C-Enduro, which is 3 m. However, the radius of curvature is smaller than the radius of the circle to be tracked (200 m). Hence, with this set up, the vehicle is able to track the desired contour. Though the 'UoE Node' publishes the commanded quantity every 5 seconds, a new commanded heading/WP is computed every 30 seconds. This choice is aimed at assessing a worst-case tracking performance.

The results obtained are shown in Fig. 7.5.1. The results obtained in the *Heading Hold* mode are shown in Fig. 7.5.1a; particularly, the black dotted circle is the tracked circle, centred at $(x_c, y_c) = (-0.2, -0.05)$, while the vehicle's trajectory and initial position are shown as a blue line and a red star respectively. The vehicle's initial direction of movement is also indicated with a green arrow. The vehicle successfully tracks the circular contour, on which it is initially deployed. The tracking error $\sigma(t)$ is shown in blue in Fig. 7.5.1c.

The trajectory of the vehicle when navigated using the *Seek Position* mode is shown in green in Fig. 7.5.1b. The initial position of the vehicle is shown as a red star and the commanded WPs are shown as red circles. The tracking error is shown in green in Fig. 7.5.1c, for comparison with the *Heading Hold* mode of navigation. The results obtained in a Matlab simulation are also shown, in order to compare them with the pre-trials results. The trajectory of the vehicle obtained in simulation is shown in red in Fig. 7.5.1a. As visible from Fig. 7.5.1c, the tracking errors from the two navigation modes are comparable, in order of magnitude, to the tracking error obtained from the pure simulation. Even the tracking error obtained in simulation, in fact, is not identically zero. This is due to the slow control action update frequency and to the chosen values of the controller gains, which limit the vehicle's turning capability. The *Heading Hold* scheme gives tighter tracking, with an upper limit on the tracking error of 30 meters. The *Seek Position* mode, instead, has a maximum error of 60 meters. The increased tracking error may be due to the different behaviour of the low-level controllers of the vehicle in the two modes of operation. An additional cause may be the longer integration step used in the *Seek Position* mode and introduced to prevent the vehicle to reach the neighbourhood of the WP defined by the acquisition distance. This suggests that the *Heading Hold* navigation mode should be chosen as the preferred navigation mode.

To further highlight the high performances obtainable through the *Heading Hold* control strategy, the results of an additional pre-trial are shown in Fig. 7.5.2. The configuration of the pre-trial is maintained unchanged with respect to the case in Fig. 7.5.1a. Even in this case, the vehicle successfully tracks the desired circular contour, and the obtained tracking error, shown in Fig. 7.5.2b, is in order of magnitude completely comparable to the tracking error obtained in simulations.



(a) Vehicle's trajectory

(b) Tracking error

Figure 7.5.2: Additional pre-trials results - circumference tracking

7.6 Sea trials results

The sea trials have been completed in Ardmucknish Bay off Dunstaffnage in Scotland ($56°28'58.1''N - 5°25'54.5''W$) in March 2016. A chart of the bay is shown in Fig. 7.6.1. This location has been chosen because of the low sea traffic and the direct access to available support facilities, including a slipway for the launch and recovery of the vehicle. Additionally, the bay has been chosen for its bathymetric features. The shallow water depth contours, in the range $5 - 20$ meters, follow the shore all the way around the bay, performing several turns. Additionally, other contours, such as the 30 m contour, are closed. The Ground Control Station (GCS), shown in Fig. 7.6.1 as an encircled black cross, is the on-shore location, where the 'UoE Node', the VP laptop and the Wi-Fi antenna have been set up. The bay in front of the GCS has a sweep of approximately 2.4 km, which is greater than the range of the used directional Wi-Fi antenna, of the order of 2 km. Consequently, the area of operation of the vehicle was limited by the Wi-Fi communication link.

During the sea trials, the vehicle was operated in the *fixed thrust* mode, with the level of thrust fixed at the 20%. This corresponds to a speed of approximately 0.7 m/s. The control gains in (3.3.10) were set to $r_1 = 28, r_2 = 2$, as in the pre-trials. Because of the lower speed considered here with respect to the pre-trials, this configuration

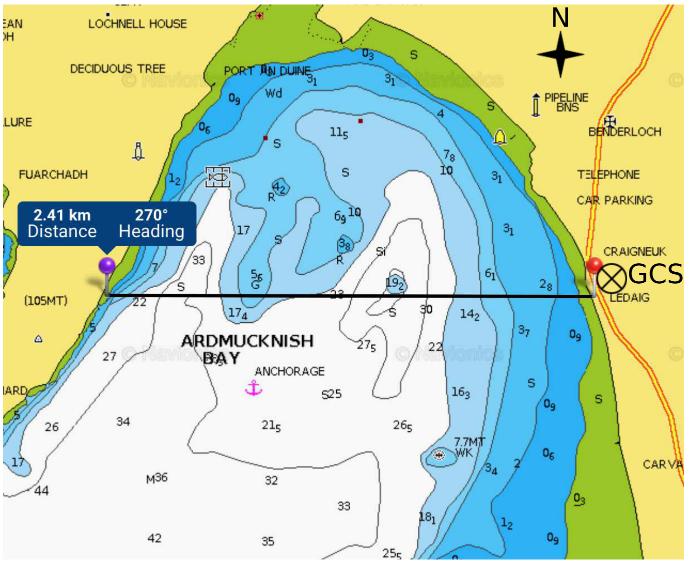
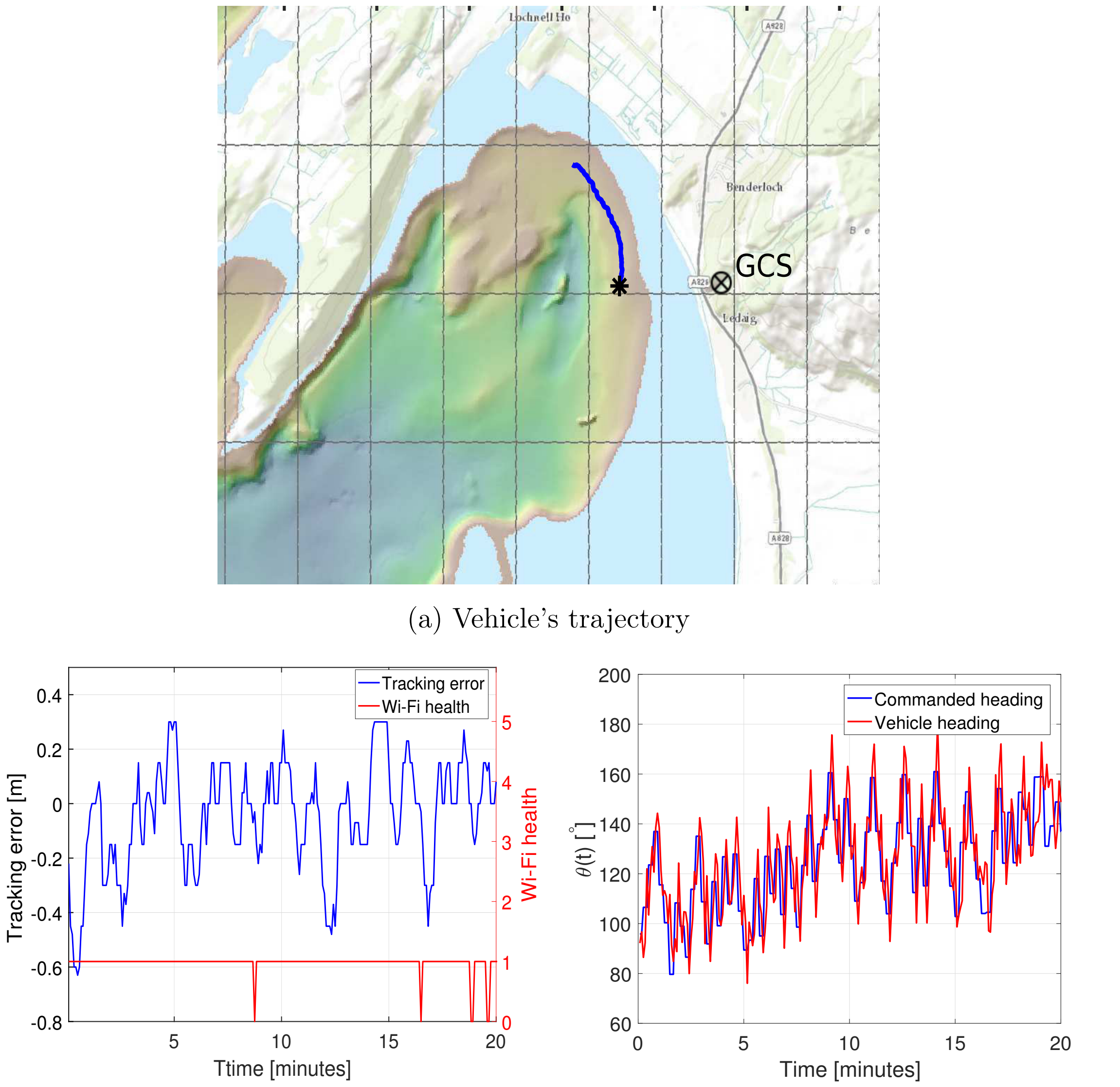


Figure 7.6.1: Ardmucknish Bay chart (obtained from Navionics)

allows the vehicle to track contours having a radius of curvature $R > R_{min} \approx 90$ m. The vehicle was operated in the *Heading Hold* mode and the commanded heading was updated at fixed intervals of 15 seconds.

7.6.1 Trial Result 1 - 12 m contour

The results obtained from a first sea trial are shown in Fig. 7.6.2. In this specific sea-trial, the vehicle was commanded to track a depth contour characterized by $\gamma^* = 12$ m. Prior to initialize the autonomous mode by enabling the 'UoE Node', the vehicle was manually driven to a vicinity of the tracked contour. Once the 'UoE Node' was started, it gained control of the vehicle through the `/control/vehicle/pilot` service and it started controlling its direction through the *Heading Hold* mode. The results shown in Fig. 7.6.2 are limited to the time the 'UoE Node' was active. The trajectory of the C-Enduro is shown on the bathymetry image of Ardmucknish Bay in Fig. 7.6.2a. The initial position of the vehicle when the 'UoE Node' took over control of the vehicle, is indicated as a black star. In the bathymetric image, the color scale represents the water depth, and the full range of the scale is between 0 m and 50 m. Fig. 7.6.2b shows the tracking error and the Wi-Fi health signal. The tracking error, which is the value of the sliding variable $\sigma(t)$, is defined as the difference between the instantaneous depth measurement and the depth to be



(a) Vehicle's trajectory

(b) Tracking error and Wi-Fi health

(c) Commanded heading and vehicle's heading

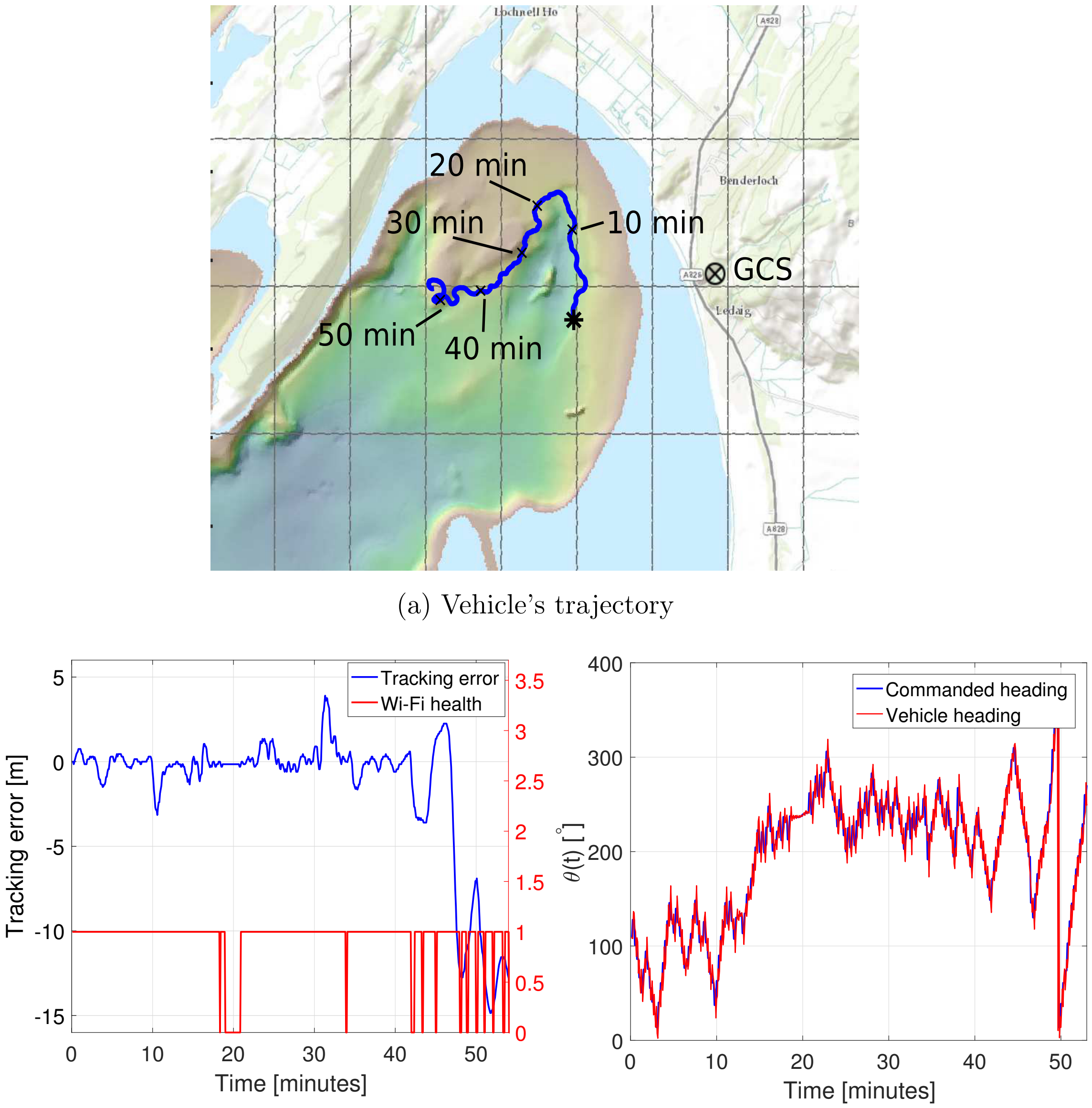
Figure 7.6.2: Sea trial results - 12 m contour tracking

tracked (12 m). The tracking error is, in absolute value, smaller than 0.5 m for the entire duration of the trial. This implies that the guidance commands, defined by the 'UoE Node' through the controller in (3.3.10), successfully drive the vehicle along a contour of 12 m depth, with at most an error of ±0.5 m. From a practical purpose, this shows the effectiveness of the proposed approach. The Wi-Fi health signal is shown in Fig. 7.6.2b. This is at level 1 when the Wi-Fi connection is in good health, while it becomes 0 when the Wi-Fi link is temporarily lost. A few Wi-Fi drop outs are observable towards the end of the sea-trial, but these do not affect the accuracy of boundary tracking in this case. The time history of the commanded heading is shown in blue in Fig. 7.6.2c, where the actual heading of the vehicle is shown in red. It is manifest that the low-level controller successfully steers the ASV in order to make it move according to the commanded heading, which is the guidance command computed by the 'UoE Node'.

A remark on this trial result is that the tracking accuracy could have been enhanced by the shallow seabed conformation in the area surrounding the tracked contour. By looking at the colour scale in Fig. 7.6.2a, in fact, it is possible to observe how the water depth only shows small changes in the area of the trial. This could have reduced the effect of a positional error with respect to the actual location of the tracked contour. In addition, the conformation of the contour is very regular, as it does not show any sharp feature, which may challenge the tracking capability of the vehicle. Consequently, tracking results, in this case, simplified.

7.6.2 Trial Result 2 - 20 m contour

The objective of this second trial was to track a contour of constant depth, fixed at $\gamma^* = 20$ m, which is away from the shallow region previously considered. As it can be seen from Fig. 7.6.3, to precisely track the features of this contour the vehicle has to perform several turns. The trajectory followed by the vehicle is shown in Fig. 7.6.3a, where the vehicle's initial deployment is highlighted with a black star. The depth variations along the trajectory of the vehicle are evident from the variations in the colour scale of the bathymetric image in Fig. 7.6.3a. Considering the steepness of the seabed in the area of the tracked contour, the tracking error, shown in Fig. 7.6.3b, is very satisfactory, being less than 1 m in absolute value for most of the duration of the trial (up to 45 minutes) with the exception of a few peaks, in which it is still less than 4 m. These excursions are justified for two main reasons: the conformation of the tracked contour and the Wi-Fi drop outs. After approximately 10 minutes of the trial, the curvature of the tracked contour demands the vehicle to turn beyond its capability, limited by the present configuration to $R_{min} \approx 90$ m. This sharp feature in the tracked contour is the cause of the spike in the tracking error visible in Fig. 7.6.3b. This, however, is successfully reduced in amplitude and



(a) Vehicle's trajectory

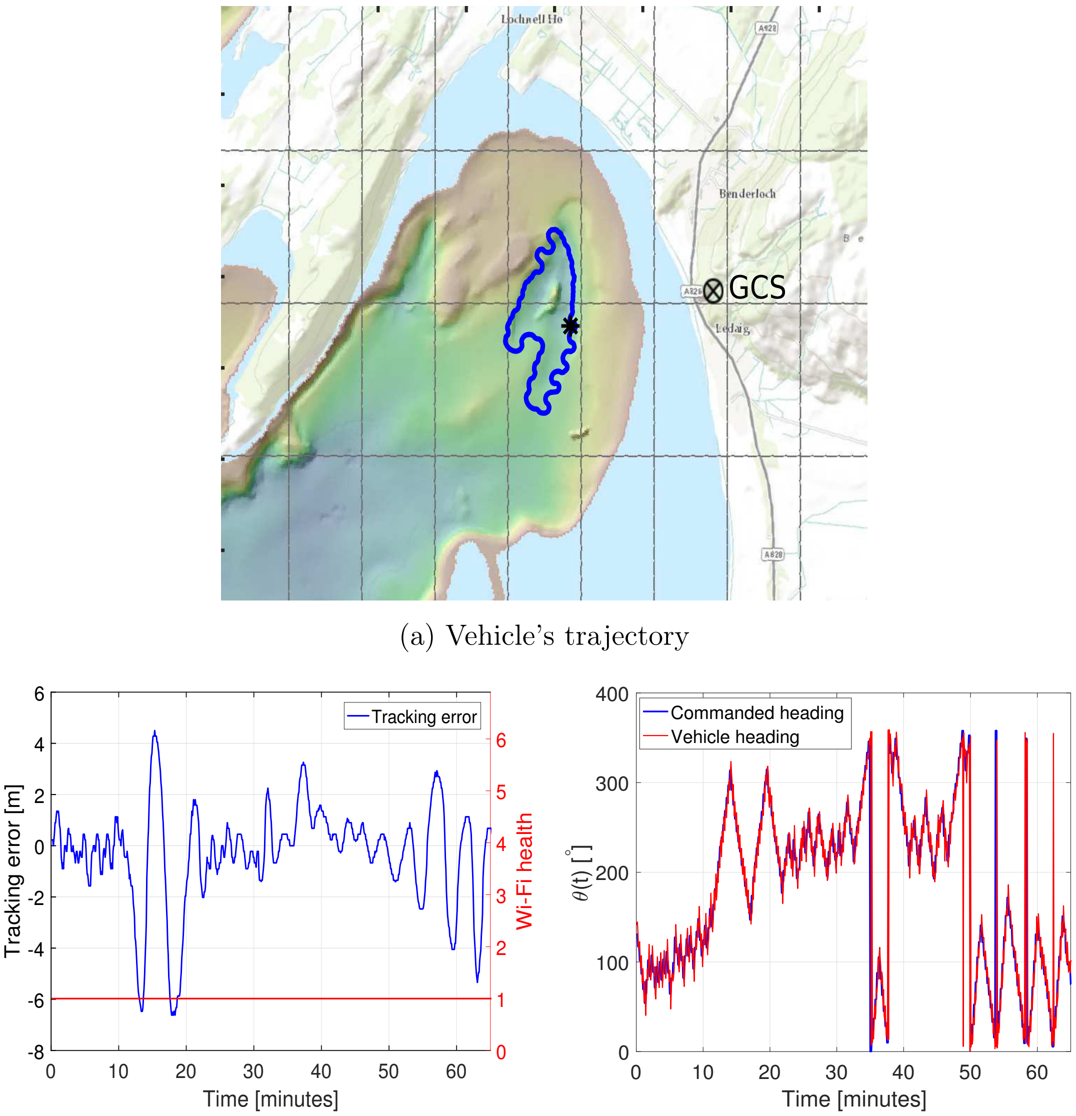
(b) Tracking error and Wi-Fi health

(c) Commanded heading and vehicle's heading

Figure 7.6.3: Sea trial results - 20 m contour tracking

a good tracking accuracy is recovered. As it has been highlighted in the virtual trials in Section 7.4, the controller gains r_1, r_2 can be increased in order to allow the ASV to better track the constant-depth contour along sharp features.

The Wi-Fi health signal is shown in red in Fig 7.6.3b. Wi-Fi drop outs which temporarily affect tracking, can be seen at approximately 35 and 42 minutes of the trial. From 48 minutes onwards, when the vehicle is on the boundary of the Wi-Fi antenna range, several subsequent Wi-Fi drops are observed, which definitely worsen the tracking performances.



(a) Vehicle's trajectory

(b) Tracking error and Wi-Fi health

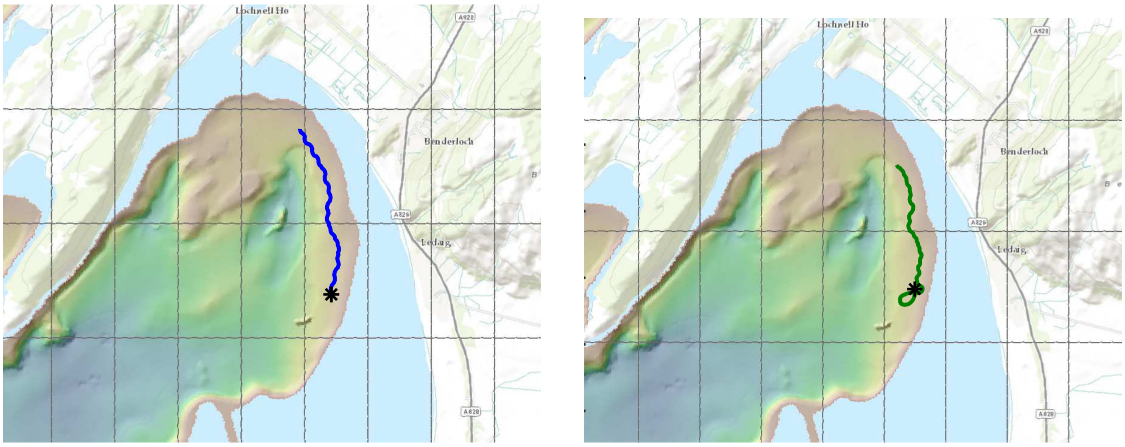
(c) Commanded heading and vehicle's heading

Figure 7.6.4: Sea trial results - 32 m contour tracking

7.6.3 Trial Result 3 - 32 m contour

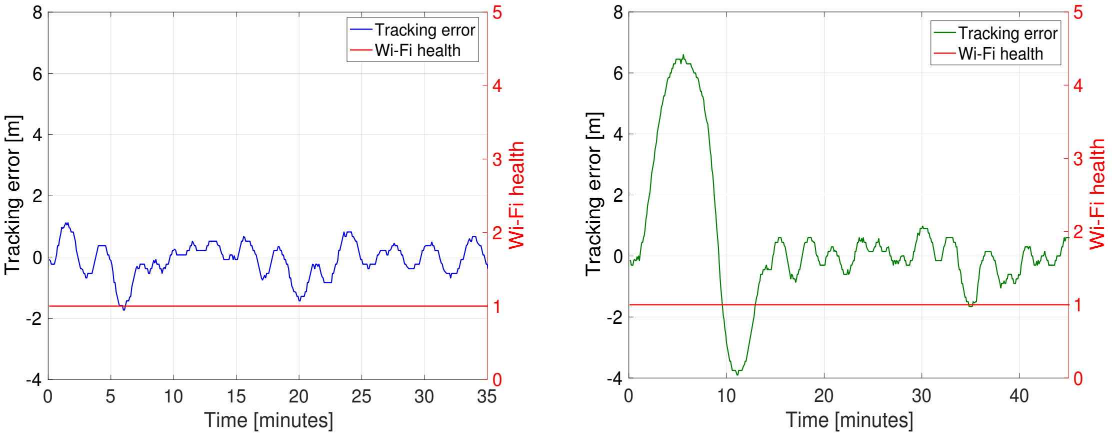
An additional set of sea trials results is shown in Fig. 7.6.4. In this trial, the vehicle was required to track a closed contour, characterized by $\gamma^* = 32$ m. The vehicle's trajectory, in Fig. 7.6.4a, shows how the vehicle successfully tracks the complete closed contour in an anticlockwise direction. The tracking error is shown in Fig. 7.6.4b. After approximatively 10 minutes, the tracking error shows some large oscillations: these are due to the abrupt manoeuvres that the vehicle needs to perform in order to continue tracking the constant depth contour. A similar effect was observed in Fig. 7.6.3a, but the contour tracked here is characterized by an even sharper feature. The vehicle loses tracking temporarily and it performs a sharp turn to turn back and recover tracking. This results in a series of crossings of the tracked contour, each one followed by an overshoot. The succession of overshoots is reduced in amplitude, until tracking of the contour is regained. In fact such behaviour is a feature of the sub-optimal algorithm [84]. The second difficulty encountered during this trial is due to the absence of many points where the water was 32 meters deep in the bottom-right part of the trajectory. After approximately 50 minutes, the vehicle starts oscillating around the contour again. The algorithm, however, allows the vehicle to successfully recover tracking, provided points on the tracked contour are crossed. The overall tracking error is smaller than 5 m in absolute value for the whole duration of the trial and this has been judged satisfactory, especially due the difficulty level of this trial. Finally, Fig. 7.6.4c shows the time evolution of the guidance heading commands, computed by the 'UoE Node', and the vehicle's heading. Even in this case, the low level controllers modify the direction of movement of the C-Enduro, in order to let it move according to the commanded heading.

The presence of external environmental forces, such as wind, water currents and tides, needs to be accounted for in each of the trials. These are assumed to be unknown, but bounded, at the time of the design of the control gains. The obtained results prove the high performances obtainable with the proposed guidance strategy and highlight the robustness of the approach to a certain level of external disturbances. The dependency on the Wi-Fi communication link has been shown to impact on the performances achievable through the boundary tracking algorithm. Embedding the methodology on board of the vehicle would remove the dependence on the Wi-Fi communication link and avoid the impacts of the Wi-Fi drop-outs on the tracking accuracy. Furthermore, Wi-Fi limits the operation of the vehicle within the Wi-Fi range. These considerations have motivated the development of an embedded system, to be directly installed on board of the ASV. The developed embedded system is described in Section 7.7.



(a) Vehicle trajectory- $\theta(0) = 86.7°$ (b) Vehicle trajectory - $\theta(0) = 211.9°$

Figure 7.6.5: 12 m contour tracking - Vehicle's trajectory



(a) Tracking error - $\theta(0) = 86.7°$ - and Wi-Fi health (b) Tracking error - $\theta(0) = 211.9°$ - and Wi-Fi health

Figure 7.6.6: Tracking errors - 12 m contour tracking

7.6.4 Sliding direction

The role of the definition of the sliding variable in determining the sliding direction has been discussed in Section 3.3.2 and has been demonstrated in an additional sea-trial. Particularly, in this case, tracking the desired contour along the bay moving initially towards North[3] from the GCS position is equivalent to moving on the tracked contour in anticlockwise direction; and viceversa. Consequently, defining $\sigma(t) = \gamma(x(t), y(t)) - \gamma^*$, the vehicle will move along the tracked contour towards North.

As a demonstration of this, the vehicle has been required to track a contour characterized by $\gamma^* = 12$ m in anticlockwise direction in two different trials. The initial

[3] The North direction is indicated in Fig. 7.6.1

heading of the vehicle in the two cases is chosen as $\theta(0) = 86.7°$ and $\theta(0) = 211.9°$. In the first case, consequently, the vehicle is initially moving approximately towards North, while it is moving approximately towards South in the second trial. The obtained results are shown in Fig. 7.6.5. Observing the vehicle's trajectories in Fig. 7.6.5a and 7.6.5b respectively, it is manifest how the vehicle successfully tracks the constant depth contour in anticlockwise direction in both cases. When the vehicle's initial heading is directed towards South, however, the vehicle is forced to perform an initial turn, before starting tracking the contour in the desired direction; this is evident from Fig. 7.6.5b. The tracking errors associated with the two trials are shown in Fig. 7.6.6. When the vehicle is initially oriented in the 'right' direction, the vehicle starts tracking the contour immediately and the maximum tracking error is of the order of 1 m, as visible from Fig. 7.6.6a. The turning phase required when the vehicle is initially oriented towards South determines an initial tracking error of the order of 6 m, as visible from Fig.7.6.6b; this is followed by a few oscillations around the tracked contour, the first one determining an overshoot of the order of 4 m. The amplitude of the oscillations around the tracked contour, however, is quickly reduced; once this is verified, in approximately 15 minutes, the tracking error in Fig. 7.6.6b is of the same order of magnitude of the tracking error shown in Fig. 7.6.6a. In conclusion, these trials demonstrate how the direction of movement along the tracked contour can be entirely determined through the choice of the sliding variable definition, and it is independent of the vehicle's initial orientation relative to the tracked contour.

7.7 Embedded system development

An embedded system implementing the boundary tracking algorithm introduced in Chapter 3, has been developed. An embedded system is a computer system with a dedicated function, which is operated within a larger mechanical or electrical system [152]. The developed embedded system implements the guidance strategy necessary to achieve boundary tracking with an ASV. Therefore, the embedded system plays the role of the 'UoE Node' introduced in Section 7.3.

The installation of the embedded system on board of the ASV is the natural development of the networked infrastructure used for the experimental validation of the boundary tracking algorithm in Section 7.5 and Section 7.6. Embedding the 'UoE Node' computation capability on board of the autonomous vehicle would allow the operation of the ASV independently of the Wi-Fi communication link. In this way, the applicability of the proposed strategy would be highly enhanced and a self-contained autonomous system would be obtained.

The embedded system developed in this section has the potential of being directly installed on-board of the autonomous vehicle. The system includes the computation capability necessary to determine the guidance control for the vehicle, and makes use of the instantaneous measurements from the sensors. In this way, the autonomous vehicle could be operated in a fully autonomous way, as per the definition in [19], and the dependency on the Wi-Fi link exploited in the networked infrastructure for the pre-trials and the sea-trials would be relaxed.

The embedded system has been developed on a ECW 281B computer, which is a fanless system with an Intel Celeron J1900 Processor and the Ubuntu 14.04 operating system. The communication infrastructure between the 'UoE Node', the VP and the sensors has been built through a ROS network, whose structure, slightly different from the one presented in Section 7.3, is described in the following section. The embedded system has been developed in C++, with the aid of the C++ package `roscpp`, used to implement the ROS network functionalities.

7.7.1 Embedded system implementation details

The embedded system has been developed in C++ as a ROS package. A ROS package is a coherent collection of files, including both executables and supporting files [153]. The installation of a ROS package on a machine allows the automatic definition of the ROS nodes, their functionalities and their dependencies. Additionally, through the ROS package launch file, a mechanism to automatically start the differ-

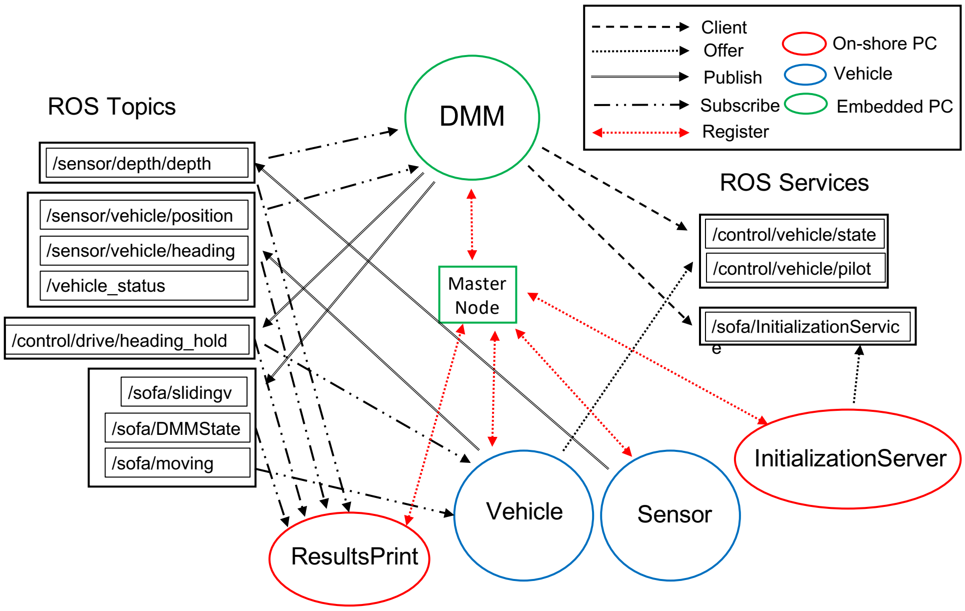


Figure 7.7.1: Modified ROS network

ent nodes, possibly on different machines, and therefore to create the ROS network is provided. Having the ROS package installed on the embedded system therefore allows the automatic set up of the requested ROS network for the trial initialization.

The ROS package is developed in order to make the 'UoE Node' ready for direct installation on the autonomous vehicle. The package includes additional nodes, developed to test the proposed approach in a virtual environment. The results of the virtual experiments obtained through the developed ROS package have been compared with the results obtained through a Matlab simulation, in a code verification test.

The nodes composing the ROS network, along with the respective ROS topics and services, are shown in Fig. 7.7.1. The network includes the following nodes: the 'UoE Node' Decision Making Module (DMM) node, the 'InitializationServer' node, the 'Vehicle' node, the 'Sensor' node and the 'ResultsPrint' node. Fig. 7.7.1 also shows how the nodes are physically distributed between the embedded computer, the on-shore computer and the vehicle.

The 'UoE Node' node implements the boundary tracking guidance strategy, by subscribing to the topics relative to the vehicle's measurements and publishing the commanded guidance control, similarly to Section 7.3. For monitoring purposes, the 'UoE Node' node saves the trial results in a local file, which is stored on the embedded computer. Hence. the 'UoE Node' also subscribes to additional topics, whose content is not used for determining the guidance control. The role of the 'InitializationServer' node, running on the on-shore PC, is unchanged with respect to Section 7.3. The 'InitializationServer' node offers the `/sofa/InitializationService` service, which is requested by the 'UoE Node' node at the beginning of the trial to initialize the trial parameters. The initialization happens through the Wi-Fi link between the embedded computer and the on-shore computer. In the absence of this link, default values of the initialization parameters, stored by the 'UoE Node' node, are used. The 'ResultsPrint' node has been implemented for monitoring purposes. It subscribes to all the available topics, to monitor some quantities of interest during the trial and to store certain quantities of interest in a local file on the on-shore computer for post-processing. The role of the 'Vehicle' and the 'Sensor' nodes is identical to the role of the 'ASVPilot' and the 'Garmin' nodes in Fig. 7.3.1. For testing the embedded system in a virtual environment, these nodes have been built to simulate the presence of a vehicle and a sensor. The 'Vehicle' node mimics the behaviour of the ASV through the kinematic model in (3.2.3). Particularly, the 'Vehicle' node subscribes to the instantaneous commanded heading information, i.e. it subscribes to the `/control/drive/headind_hold` topic, where the 'UoE Node' publishes the guidance command. In this, the 'Vehicle' node behaves exactly as the actual vehicle would do. Inside the 'Vehicle' node, however, the guidance command is used in an Euler integration step applied to the vehicle's kinematic model in (3.2.3), in order to update the vehicle's position and heading. This information is then published by the 'Vehicle' node in the corresponding topics. Similarly, the 'Sensor' node mimics the behaviour of the real sensor and, consequently, publishes the depth measurement at the instantaneous position of the vehicle on the `/sensor/depth/depth` topic. Differently from the real sensor, this node subscribes to the vehicle's position topic and uses this information to construct a virtual measurement.

At the beginning of each trial, the ROS network is established through the launch file belonging to the ROS package. The launch file is capable of starting several nodes at once. The launch file can also start a node remotely on a specific machine/device, identified by its IP address. The launch file firstly initiates the 'ROS Master' node. Subsequently, it establishes the remaining nodes. In the simulated environment, the 'ResultsPrint' node, the 'Vehicle' node, the 'Sensor' node and the 'InitializationServer' node are started on the on-shore computer. The 'ROS Master' node and the 'UoE Node' are started on the embedded computer by the launch file. The launch file also defines the output files on both the on-shore and the embedded computers, where the trial variables are stored. In a real trial making use of the embedded system, the launch file would only create the 'ResultsPrint' node and the 'InitializationServer' node on the on-shore machine and it would start the 'ROS Master' node and the 'UoE Node' on the embedded machine, while the existing 'ASVPilot' node and the 'Garmin' sensor node would be simply interacting on the same ROS network.

7.7.2 Code verification results

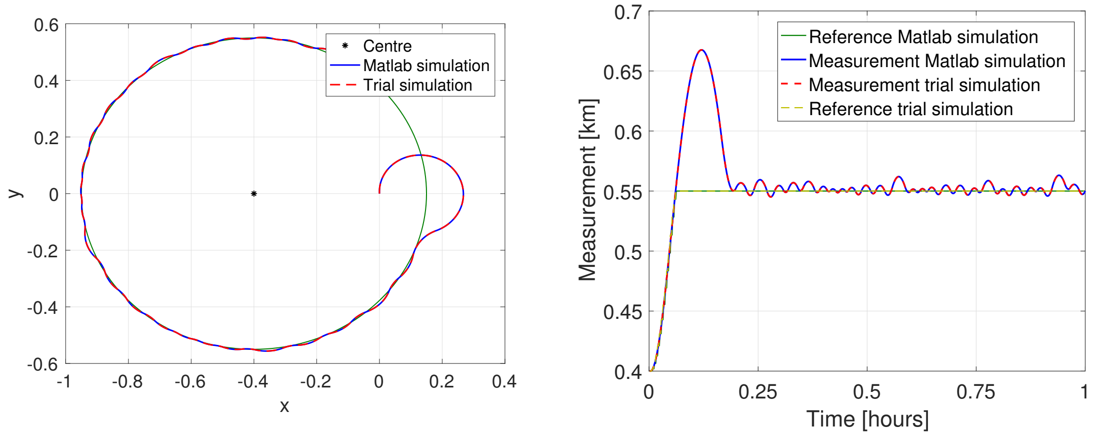
The results obtained through the embedded system in a set of virtual trials are compared with the results obtained, with an identical set up, through a Matlab based simulation. The aim of these tests is the validation of the C++ code and, therefore, of the embedded system functionality.

For the virtual trials set up, the capability of the embedded system to make the vehicle track a circular contour has been tested, similarly to Section 7.5. Particularly, the 'Sensor' node makes use of the position of the vehicle, obtained by subscribing to the relative topic, in order to create a virtual measurement. Specifically, a range measurement between the position of the vehicle and the known position of the centre of the tracked circular contour is computed.

After the execution of the package launch file, the nodes in the network are started. Subsequently, the user is prompted to enter the values for the trial initialization

parameters by the 'InitializationServer' node. The entered parameters are used by the 'UoE Node'. The tracked value of the spatial field is set to $\gamma^* = 0.55$ km; consequently, the vehicle is required to track a circle of radius 0.55 km. The tracked circular contour is centred at the fixed position $(x_c, y_c) = (-0.4, 0)$. Additionally, the user is allowed to initialize the trial length and the level of thrust to be used by the vehicle, which are fixed to one hour and the 20% respectively. The gains of the controller in (3.3.10) are chosen as $r_1 = 28$ and $r_2 = 2$, and the reference trajectory parameter K in (5.4.13) is set to $K = 0.12$.
The results obtained are shown in Fig. 7.7.2. It is manifest how the results obtained through the C++ package in a simulated environment and those obtained through a Matlab simulation are completely equivalent. This perfect alignment confirms the correct implementation of the Euler integration scheme in the embedded system. Additionally, the obtained results perfectly overlap despite the different time execution, as the package simulation happens in real time while the Matlab simulation happen in simulation time. This test, in conclusion, confirms that the approach implementation in C++ correctly represents the approach initially developed in Matlab. Consequently, the successful sea-trials results obtained in Section 7.6 through the networked framework would be repeatable by installing the embedded system on-board of the autonomous vehicle. The embedded system would have the advantage of being independent of the Wi-Fi communication link, thus rendering the framework robust to Wi-Fi losses.

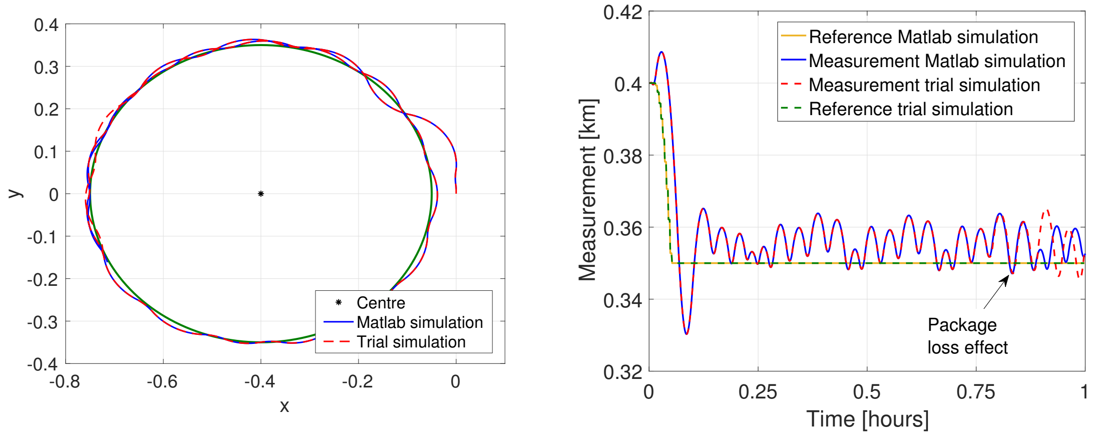
It has been observed that a discrepancy between the results obtained through the ROS package and the Matlab simulation can arise if the communication over the ROS network is imperfect. The communication between the sensors and the 'UoE Node' happens through an Ethernet cable. It is sufficient to have a single package loss to observe the difference in the behaviour of the Matlab simulation and the C++ trial. Despite these differences, however, the implemented scheme is capable of maintaining high performances in terms of tracking accuracy, even following a communication defect. An example of this can be observed from the results in



(a) Vehicle's trajectory

(b) Vehicle's measurement

Figure 7.7.2: Boundary tracking simulation - comparison of the embedded system and the Matlab results



(a) Vehicle's trajectory

(b) Vehicle's measurement

Figure 7.7.3: Boundary tracking simulation - discrepancy between the embedded system and the Matlab results following a communication package loss

Fig. 7.7.3, where the initialization has been maintained as in Fig. 7.7.2, but with $\gamma^* = 0.35$ km and $\sigma(t) = \gamma^* - \gamma(x(t), y(t))$. After approximately 0.8 hours, the data over the ROS network have failed to be updated. Consequently, the perfect equivalence between the Matlab simulation and the C++ trial is lost. However, the performance in terms of tracking accuracy is maintained and boundary tracking is successfully achieved.

7.8 Concluding remarks

The experimental validation of the boundary tracking approach introduced in Chapter 3 with the autonomous surface vehicle C-Enduro is presented in this chapter. The experimental framework set up and the ROS network construction are described in detail. The results obtained from a set of virtual trials are described, before showing the effectiveness of the proposed solution in a set of pre-trials. Finally, the application of the boundary tracking approach to the study of bathymetric features is described and the results obtained in a set of sea-trials are presented. In addition, the development of an embedded system, aimed at being directly installed on the ASV in the future, is described. The validity of the developed embedded system is demonstrated in a simulated environment, purposely created. Specifically, the perfect alignment between the results obtained through the developed embedded system in a simulated environment and those obtainable from the original Matlab simulation is demonstrated.

The work presented in this chapter has great potential for future expansion. Firstly, as a future extension, the designed embedded system could be tested in a set of sea-trials similar to those described in Section 7.6. This would fully demonstrate its effectiveness. Additionally, the source seeking algorithm presented in Chapter 5 could be included in the embedded system, in order to allow the ASV to perform differently, depending on the pre-specified control objective.

Chapter 8

Conclusion and future work

The principal subject of the work in this thesis is the development of techniques for exploring unknown or imprecisely known environmental features with autonomous vehicles. Through the proposed techniques, the enhancement of the autonomy level associated with the deployment of the autonomous vehicles is possible. The control strategies proposed in this thesis belong to two main classes: boundary tracking algorithms, for the identification of the spatial field's boundary, and source seeking algorithms, for the detection of the source of the spatial phenomenon. The approaches are applied to the control of ASVs and AUVs for the exploration of two and three dimensional oceanic features.

The deployment and control of autonomous vehicles for environmental monitoring is a fast growing area of research, and increasing interest is directed towards the development of control techniques characterized by low measurement and computation requirements. The underlying motivation is the willingness to deploy *fully autonomous* vehicles in hazardous environments, for the characterization and the possible resolution of dangerous phenomena. The deployment of fully autonomous vehicles is ultimately aimed at making their operation independent of the presence of a human operator, controlling the vehicle directly or pre-planning its mission.

As oceans constitute the less precisely known part of the environment, and possibly the most dangerous one, the desire to characterize oceanic features with autonomous vehicles has promoted the fast development of different typologies of MAS, which have constituted the first examples of autonomous vehicles. The results presented in this thesis consider a wide variety of oceanic features, such as a sea-surface temperature front, a tracer release and the bathymetry of an area of interest. The proposed boundary tracking and source seeking algorithms are applied to ASVs and AUVs in simulations, using specific kinematic and dynamic models. Additionally, the suboptimal sliding mode boundary tracking algorithm is applied to the ASV C-Enduro for its experimental validation in a set of sea-trials. It is worth mentioning that the applicability of the control strategies proposed in this thesis is not limited to MAS and could, in the future, be extended to the control of UGVs and UAVs, with similar principles.

The results presented in this thesis contribute to the existing literature by introducing novel boundary tracking and source seeking techniques. The suboptimal sliding mode boundary tracking algorithm, originally proposed in [39], is presented in Chapter 3. This algorithm drives a single autonomous vehicle along the boundary of an unknown spatial map, by controlling its direction of movement in the two-dimensional space. The work in Chapter 3 extends the applicability of the algorithm in [39], which was originally applied to a simple and static scenario, to spatial phenomena having a dynamic nature. Additionally, the presence of disturbances affecting the movement of the vehicle is accounted for. The suboptimal sliding mode boundary tracking algorithm is also extended to the study of three-dimensional oceanic features in Chapter 4. Particularly, the unknown boundary of an underwater oceanic feature is explored through the deployment of an underwater glider. The approach is validated in simulations, on the basis of a realistic model of this type of vehicle, accounting for its kinematics and dynamics. For this application and throughout the thesis, the simulations are based on realistic datasets relating to oceanic features, obtained from the Met Office, UK. In addition to extending the applicability of the boundary tracking algorithm in [39] to the three-dimensional space, the approach proposed in Chapter 4 represents a preliminary step towards a different way of operating underwater gliders. Gliders are typically pre-programmed or remotely controlled. Moreover, their low level controllers rely on an estimate of their position, obtained through dead-reckoning techniques and often inaccurate, in order to achieve the predefined mission. Applying the strategy introduced in Chapter 4, the autonomy level associated with the operation of underwater gliders can be enhanced and the dependency on a pre-planned mission can be relaxed. This would also make the operation of underwater gliders independent of the dead reckoning position estimate. The boundary tracking algorithm in Chapter 4, in fact, only requires the instantaneous measurement of the spatial field at the position of the glider and is independent of the position of the glider. The significance of this work in considering different ways of operating underwater gliders makes the validation of this approach in sea-trials one of the recommendations for the future extension of this work.

A two dimensional source seeking algorithm is proposed for a single autonomous vehicle in Chapter 5. The approach is based on sliding mode extremum seeking control techniques and, provided the introduced sufficient conditions are verified, ensures the finite time convergence of the vehicle to a neighbourhood of the sought source. The source seeking algorithm is applicable to both static and dynamic spatial fields. The approach is validated in simulations on a numerical example and on the dynamic dataset describing the evolution of a tracer release over a period of six days. The work in Chapter 5 could be extended by considering the definition of an adaptive reference trajectory. With the current design, in fact, a trade-off between the speed of convergence towards the source and the accuracy in the estimation of the position of the source is required. With an adaptively defined reference trajectory, it may be possible to achieve a fast movement towards the sought source and an accurate estimation of its position.

A different strategy for the study of imprecisely known environmental features with a formation of autonomous vehicles is introduced in Chapter 6. It has been observed how preliminary information about the considered spatial field is typically available prior to deploying the autonomous vehicles. This information, in most of the work in the literature, is used to design pre-planned trajectories to be followed by the autonomous vehicles. In this thesis, the preliminary information is used, differently, in the definition of a probabilistic model of the considered feature. The model is then used in the estimation of the spatial gradient of the feature, useful for the boundary tracking algorithm, and it is updated and improved through the inclusion of real world measurements collected by the vehicles. The approach is validated in simulations, based on the dataset relative to the sea-surface temperature in a tidal mixing front area. In addition, a preliminary application of the approach in collaboration with the Met Office has been performed. Particularly, the fictitious real world measurements collected by the vehicles during the simulation have been included in the Met Office model, in order to investigate the enhancement of its prediction capabilities. This preliminary evaluation has the potential of leading to interesting future results. It would be desirable to evaluate the impact of including in the Met Office model real world measurements collected by vehicle deployed in an area of interest. For instance, autonomous vehicle could be used to sample the areas associated with higher prediction uncertainty. Additionally, the attention should be focussed on the results obtainable in adverse conditions, characterized by the unavailability of classical observation technologies, such as satellites.

The work in Chapter 7 contributes at enhancing the Technology Readiness Level (TRL) of the suboptimal sliding mode boundary tracking strategy and the autonomy level of the deployed autonomous vehicle. This work, in fact, demonstrates the effectiveness of the boundary tracking algorithm presented in Chapter 3 in a set of sea-trials, demonstrating the possibility to monitor a two-dimensional environmental feature without pre-planning or remote controlling of the vehicle. Additionally, the sea trials results constitute a proof of the concept of fully autonomous deployment of an autonomous vehicle. As a further development of this work, an embedded system for future installation directly on-board of the vehicle has been described in Chapter 7. The embedded system has the potential of greatly expanding the area of operation of the vehicle, making it independent of the Wi-Fi link with the ground control station. The experimental validation of the embedded system is therefore an open direction of future research.

In addition to the suggested future extensions of the work in this thesis, many new directions of future research have been identified. As a first direction of research, it has been observed that most of the strategies presented in this thesis could be enriched with adaptation logics. The possibility to adapt the reference trajectory parameters in Chapter 5, for instance, would further improve the convergence to a neighbourhood of the source. Similarly, the possibility to adapt the suboptimal sliding mode controller's gains could be considered in order to optimize the sliding performances in boundary tracking applications. Another interesting extension could consider the probabilistic learning boundary tracking strategy introduced in Chapter 6 and extend it to the study of dynamic fields. This would enhance the applicability of the proposed strategy in field experiments.

In conclusion, the work presented in this thesis highlights the potential of deploying autonomous vehicles for the exploration of unknown or partially known environmental features. The potential of enhancing the autonomy level associated with the operation of Marine Autonomous Systems, additionally, has been highlighted throughout the thesis, and some possible solutions have been proposed.

Appendix A

Sliding mode control

The aim of this appendix is to give a brief introduction to sliding mode control. Additionally, a review of the main second order sliding mode control strategies is given, as a suboptimal second order sliding mode controller and a quasi-continuous second order sliding mode controller are used in this thesis. The interested reader is referred to [83], [84] and [154] for further details.

A.1 Sliding mode control introduction

VSC systems with sliding mode control were originally developed by researchers in Russia in the 1960s [83], but their popularity increased in the seventies, after the publication in English of a book by Itkis [155] and a survey paper by Utkin [156]. VSC systems are a class of systems characterized by a set of feedback control laws: the actual control law is changed during the control process according to a switching function, which depends on the current state of the system.

Sliding mode control belongs to the class of VSC techniques, as it is based on discontinuous feedback control laws. The aim of the discontinuous control laws is to force the system's states to reach, and subsequently to remain on, a specified surface within the state space, the so-called switching or sliding surface.

A.1.1 Concept of sliding mode

The underlying philosophy of sliding mode control is to first define a manifold, the so-called *sliding surface* [83], and to drive the states of the dynamical system onto the manifold, using a discontinuous feedback control. When the states of the system belong to the sliding surface, a sliding motion is obtained.

The sliding manifold is defined as the intersection of a certain number of surfaces, each one imposing a constraint on the system's states. Consequently, the desired behaviour of the system is assigned through the design of the sliding surface and it is obtained during the sliding motion. The sliding surface is typically parametrized as a function of a sliding variable, identified in this thesis as $\sigma(t)$.

The behaviour of the closed loop system controlled through SMC can be divided into two phases: the reaching phase and the sliding phase. The phase in which the VSC drives the system's states towards the sliding surface is the so-called reaching phase. Since the states of the system reach the sliding manifold, they are constrained to remain on it, obtaining a sliding motion. The achievement of a sliding motion coincides with the beginning of the sliding phase. The main advantages characterizing the sliding motion are presented in [83]. Firstly, during the sliding motion, the system behaves as a reduced order system, which appears to be independent of the control. The effect of the control action, rather than to prescribe the dynamic performance, is to ensure that the constraints defined by the sliding surface are met [83]. The second advantage is that, during the sliding motion, disturbances or uncertainties in the input channel are completely rejected. The system is therefore robust with respect to the so-called matched uncertainties, which are uncertainties affecting the input channel. This property makes sliding mode control particularly effective when dealing with uncertain nonlinear systems.

A.1.2 Chattering

Classic sliding mode control can be characterized by high-frequency control switching, the *chattering* effect. Chattering usually takes place during the sliding motion.

In ideal sliding, the system's states precisely belong to the sliding manifold and the control action switching frequency approaches infinity [84]. The switching frequency in practical application is upper bounded; consequently, the system's states perform a zig-zag motion of small amplitude and high frequency around the sliding surface. The amplitude of the zig-zag motion is inversely proportional to the control action switching frequency. The resulting high frequency switching control and the associated zig-zag motion characterize chattering.

Chattering is particularly dangerous in practice because it can lead to damages to the controlled systems [157], [158]. Consequently, strategies to provide smooth and continuous control signals have been investigated in the literature and different chattering avoidance strategies have been proposed. The approaches in [83] and [159] avoid real discontinuities in the control action by changing its design in a vicinity of the sliding manifold. These methods, however, have the drawback of reducing the sliding accuracy, as the sliding variable only lays in a vicinity of the sliding surface [53]. In addition, the robustness property of the sliding motion is lost [83]. Consequently, different methodologies, based on higher order sliding modes, have been proposed in the literature.

Traditional sliding mode control techniques are applied to relative degree one systems. The relative degree of a system is the number of time differentiations of the sliding variable necessary to obtain an explicit appearance of the input variable [160]. Consequently, for relative degree one systems, $\dot{\sigma}(t)$ is a function of $u(t)$. In order to avoid chattering, relative degree one systems are treated as relative degree two systems, in which the controller only appears in $\ddot{\sigma}(t)$. Practically, the discontinuous controller $\dot{u}(t)$ is designed as a virtual controller, so that the actual control $u(t)$ results continuous. As a result, chattering is avoided.

A.1.3 An illustrative example

A simple example is described in this section in order to give some insight to the characteristics of the sliding motion. A pendulum, composed of a light rod and a

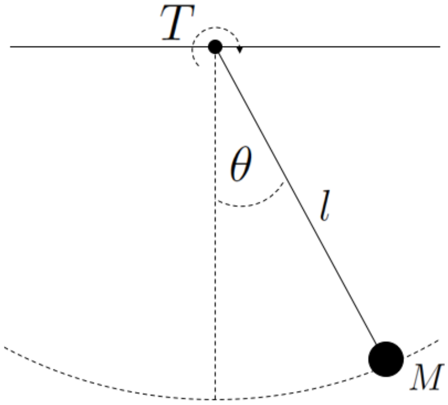


Figure A.1.1: Pendulum schematics

heavy mass, is considered. The pendulum is shown in Fig. A.1.1, where θ is the angular displacement from the vertical direction, M is the mass, l is the length of the rod and T is the torque applied at the suspension point, considered as the control input of the system. The objective of this control problem is to design the torque input T so that the pendulum returns to its vertical equilibrium point, where $\theta(t) = 0$. The pendulum dynamics, obtained neglecting the effects of friction, results:

$$\ddot{\theta}(t) = -\frac{1}{g}\sin\theta(t) + \frac{1}{Ml^2}u(t) \tag{A.1.1}$$

where g is the gravitational acceleration. Through scaling, the dynamics of the pendulum can be captured by the normalized pendulum equation [83]:

$$\ddot{y}(t) = -a_1 \sin y(t) + u(t) \tag{A.1.2}$$

where a_1 is a positive scalar and $y(t) = \theta(t)$ corresponds to the angular displacement. The first step for the design of a sliding mode controller is the choice of the sliding surface. In this case, the sliding surface is defined as:

$$\mathcal{S} = \{(y(t), \dot{y}(t)) : \sigma(t) = 0\} \tag{A.1.3}$$

whereas the sliding variable $\sigma(t)$ is defined as:

$$\sigma(t) = y(t) + \dot{y}(t) \tag{A.1.4}$$

The choice of this sliding surface is justified in [84], where it is shown that a general solution of (A.1.3) and its derivative are given by:

$$y(t) = -y(0)e^{-ct} \tag{A.1.5}$$

$$\dot{y}(t) = -cy(0)e^{-ct} \tag{A.1.6}$$

Hence, $y(t)$ and $\dot{y}(t)$ both converge to zero asymptotically. In the phase plane, which is the $y - \dot{y}$ plane, the sliding surface corresponds of a line having gradient -1 and passing through the origin, that is the diagonal of the 2^{nd} and 4^{th}. quadrants

The VSC law $u(t)$ is chosen to switch between two control structures through:

$$u(t) = \begin{cases} -1 & \text{if } \sigma(t) > 0 \\ 1 & \text{if } \sigma(t) < 0 \end{cases} \tag{A.1.7}$$

Through this logic, the control action to be used at any point $(y, \dot{y})$ in the phase plane is determined by the sign of the sliding variable in (A.1.4). The control action in (A.1.7) can be expressed in compact form as:

$$u(t) = -\text{sign}\ (\sigma(t)) \tag{A.1.8}$$

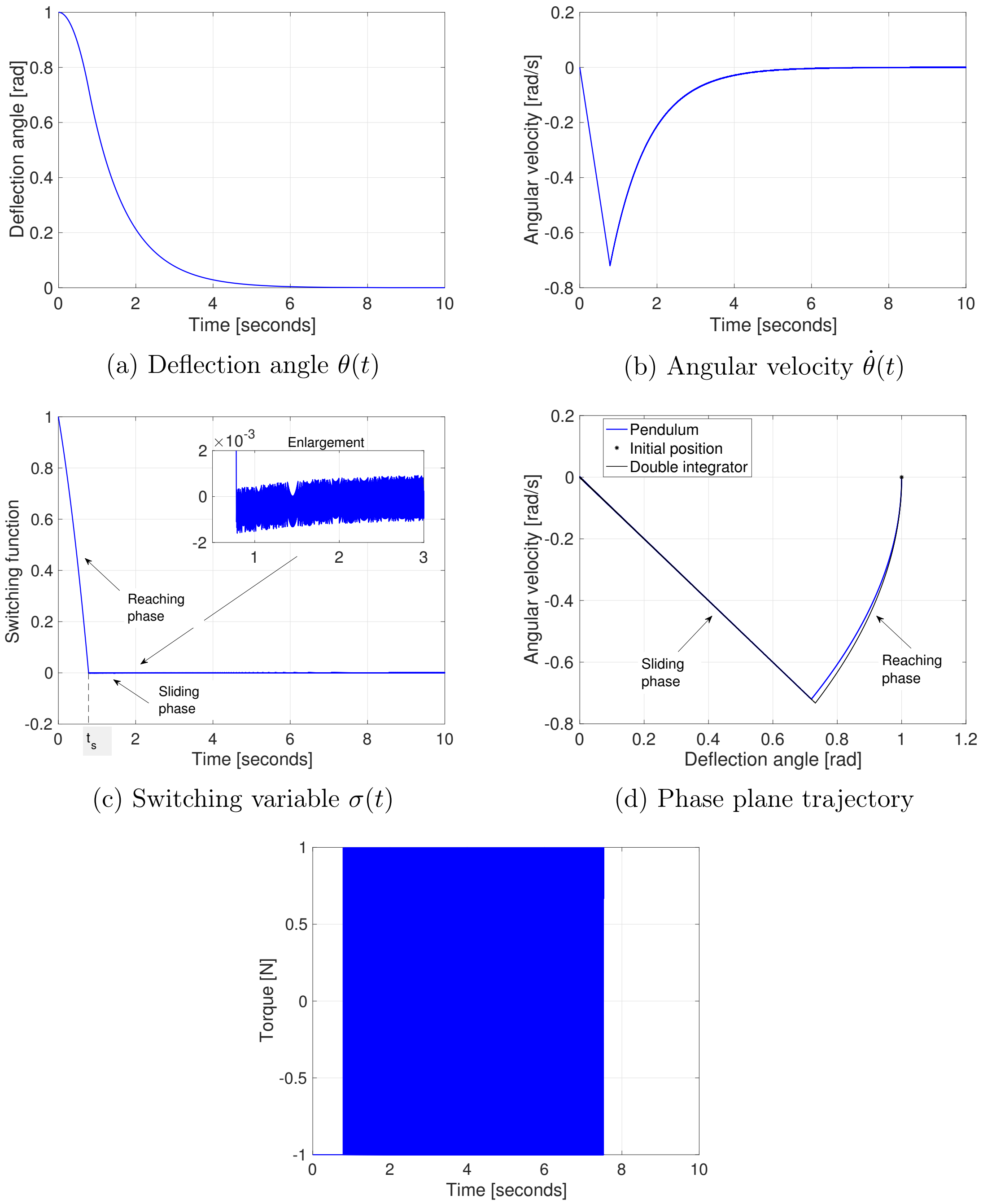
where sign$(\cdot)$ is the signum function.

The behaviour of the system described in (A.1.2), controlled through (A.1.7), has been simulated in a Matlab/Simulink environment (version R2016b) in order to investigate the properties of the sliding motion. The simulation has been initialized with $\theta(0) = 1$ rad, $\dot{\theta}(0) = 0$ and $a_1 = 0.1$. The results obtained are shown in Fig. A.1.2, where the system's states (the deflection angle and the angular velocity), the sliding variable $\sigma(t)$, the phase plane trajectory of the system and the discontinuous feedback control are shown. The settling time for the deflection angle and the angular velocity, shown in Fig. A.1.2a and A.1.2b respectively, is of approximately 5 seconds. The convergence to zero happens without any significant overshoot or oscillation. The exponential decay observable in Fig. A.1.2a, which is typical of first order systems, confirms the order reduction obtained through sliding mode control [83]. This, moreover, is in accordance to (A.1.5) - (A.1.6). Fig. A.1.2c shows the sliding variable in (A.1.4). The reaching phase, in which the control aims to force the system's states on the sliding manifold, and the sliding phase are distinguished. Additionally, the time at which the sliding surface is first met (t_s) is highlighted: a sliding motion is obtained in approximately 1 second. The phase plane trajectory of the closed loop system is shown in Fig. A.1.2d, where the reaching and sliding phases are highlighted. This plot can be used to confirm the stability of the system [85], since the origin of the phase plane is reached by the closed loop system trajectory. The control input $u(t)$ is shown in Fig. A.1.2e and is highly discontinuous. The discontinuity begins after the sliding surface is reached, that is for $t \geq t_s \approx 1$ second, when *chattering* takes place. The switching between the control structures in (A.1.7) happens with high frequency as the system trajectory repeatedly crosses the sliding surface. If infinite switching frequency was possible, the motion would be confined exactly onto the sliding surface, obtaining an ideal sliding. Being the switching frequency bounded, chattering takes place. This is confirmed by the enlargement in Fig. A.1.2c, where the high frequency zig-zag behaviour of $\sigma(t)$ around $\sigma(t) = 0$ is shown.

The pendulum dynamics in (A.1.2) can be interpreted as the dynamics of an ideal double integrator:

$$\ddot{y}(t) = u(t) \tag{A.1.9}$$

where the term $-a_1 \sin y(t)$ is a bounded uncertainty within the nominal dynamics [161]. The trajectory of the nominal double integrator, when controlled as in (A.1.7), is shown in Fig. A.1.2d for comparison. It is possible to observe that, once sliding is established, the nominal double integrator and the normalized pendulum behave identically. This shows the *robustness* of sliding mode control with respect to matched uncertainties: the nonlinear term is in fact treated as a disturbance or uncertainty entering the input channel and it is completely rejected.



(a) Deflection angle $\theta(t)$

(b) Angular velocity $\dot{\theta}(t)$

(c) Switching variable $\sigma(t)$

(d) Phase plane trajectory

(e) Control action $u(t)$

Figure A.1.2: Sliding mode control - pendulum example

The control behaviour in Fig. A.1.2e is highly undesirable in practice, as it could damage the mechanical components and the actuators. A possible solution to the chattering problem, presented in [83], consists in attempting to smooth the discontinuity in the control action in (A.1.8), which is due to the sign function. One

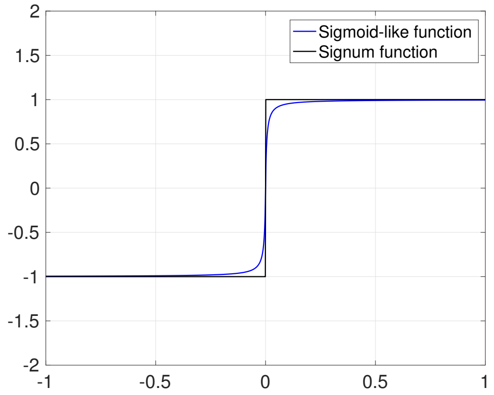


Figure A.1.3: Signum approximation through the sigmoid-like function

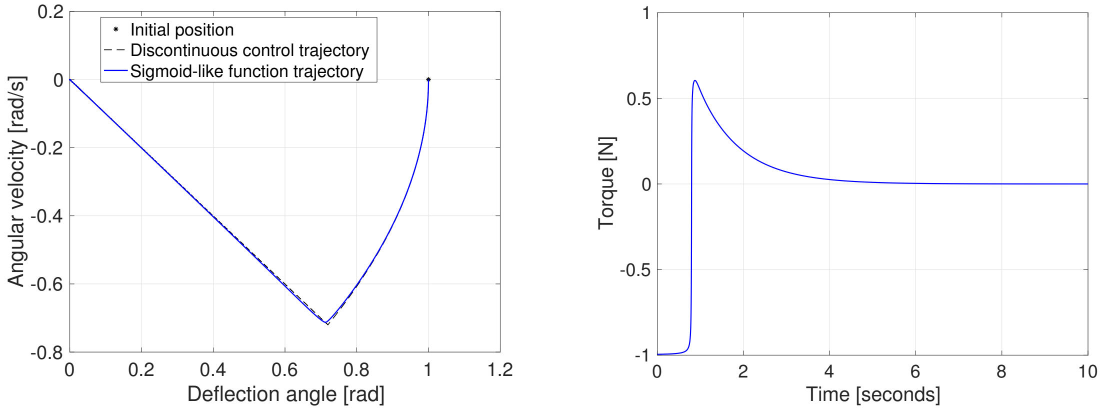
possibility is to approximate the sign function through a sigmoid-like function [83]:

$$\text{sign}(\sigma) \approx \nu_\delta(\sigma) = \frac{\sigma}{(|\sigma| + \delta)} \tag{A.1.10}$$

where δ is a small positive design constant. A graphical comparison between the sign function and a sigmoid-like function with $\delta = 0.005$ is given in Fig. A.1.3. An additional simulation is run, using the same initial conditions as in the previous one and the control law in (A.1.8), with $\nu_\delta(\sigma)$ in place of the sign function. The parameter $\delta = 0.005$ is chosen. The results shown in Fig.A.1.4 are obtained. The phase plane trajectory of the closed loop system in Fig. A.1.4a is undistinguishable from the phase plane obtained with the discontinuous control (which is shown for comparison). The obtained control $u(t)$, as visible from Fig. A.1.4b, is smooth for the entire duration of the simulation. Consequently, the use of sigmoid-like functions is a successful chattering avoidance practice. With this solution, however, ideal sliding is not obtained and the system's states are only driven to a neighbourhood of the sliding surface, obtaining the so-called pseudo-sliding [83].

A.2 Higher order sliding mode control

Conventional sliding mode control provides an effective solution to a wide range of control problems, and it is robust with respect to certain classes of uncertainty [84]. The applicability of conventional sliding modes, however, is limited to systems



(a) Phase plane trajectory (b) Control action $u(t)$

Figure A.1.4: Sliding mode control with sigmoid-like function - pendulum example

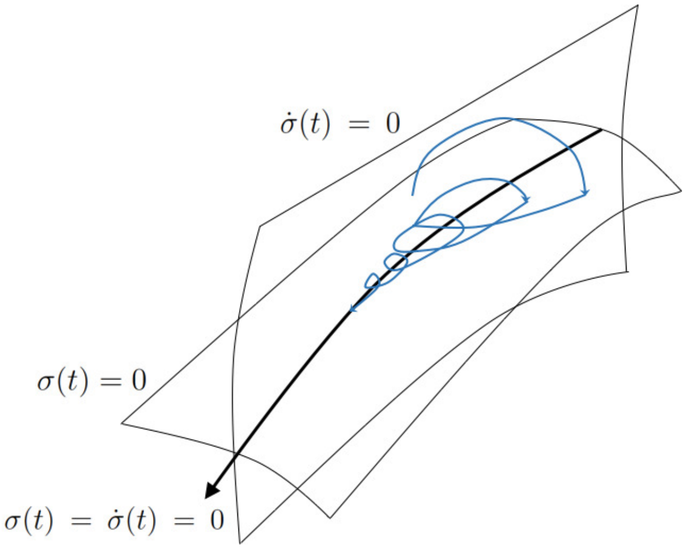


Figure A.1.5: Second order sliding mode trajectory - adapted from [86]

with relative degree one, in which the input signal explicitly appears in the first derivative of the sliding variable. Additionally, classic sliding mode control can be characterized by high-frequency control switching, the chattering effect.

When conventional sliding modes are not applicable, higher order sliding modes should be considered. These methods generalize the classic sliding mode approach by acting on higher order derivatives of the sliding manifold constraint, rather than on its first derivative. Second order sliding mode controllers, for instance, act on the second derivative of the sliding constraint and are thus applicable to relative degree two systems. Higher order sliding modes are also an effective chattering avoidance tool. A second order sliding mode controller can in fact be used for a relative degree one system in order to avoid chattering [84].

If the sliding manifold constraint is defined as $\sigma(t) = 0$, the sliding order is defined as the number of total derivatives of $\sigma(t)$, including the zero one, whose vanishing defines the sliding surface constraint [86]. Consequently, the $r-$th order sliding motion is described through:

$$\sigma(t) = \dot{\sigma}(t) = \ldots = \sigma^{(r-1)}(t) = 0 \tag{A.2.1}$$

Higher order sliding modes are generally characterized by higher information demands with respect to conventional sliding modes, and this constitutes a limitation to their applicability. Typically, an $r-$th order sliding mode controller requires the knowledge of $\sigma(t), \dot{\sigma}(t), \ldots, \sigma^{(r-1)}(t)$ [86].

Second order sliding mode controllers, because of their low information requirements in terms of derivatives of the sliding variable, are amongst the most widely used higher order sliding mode techniques. These are aimed at forcing a second order sliding, characterized by $\sigma(t) = \dot{\sigma}(t) = 0$, as schematically shown in Fig. A.1.5, and are typically designed for relative degree two systems.

Assuming that the controlled system has constant relative degree two, the second derivative of the sliding variable can be expressed in the following generic form [84]:

$$\ddot{\sigma}(t) = \xi(t) + b(t)u(t) \tag{A.2.2}$$

where $\xi(t) = \ddot{\sigma}(t)|_{u=0}$ and $b(t) = \frac{\partial}{\partial u}\ddot{\sigma}(t) \neq 0$ are some unknown smooth functions. It is assumed that the following classical assumptions on the uncertain terms hold globally:

$$|\xi(t)| < C, \quad 0 < K_m \leq b(t) \leq K_M \tag{A.2.3}$$

where C, K_m, K_M are positive constants. Taking the introduced bounds into consideration, (A.2.2) can be reformulated through the following differential inclusion [84]:

$$\ddot{\sigma}(t) \in [-C, C] + [K_m, K_M]u(t) \tag{A.2.4}$$

The problem is to find a feedback control $u(t) = \varphi(\sigma(t), \dot{\sigma}(t))$ such that the trajectories in (A.2.4) converge to the origin $\sigma(t) = \dot{\sigma}(t) = 0$ in finite time.

An overview of the main second order sliding mode control strategies is given in the following sections.

A.2.1 Twisting controller

The twisting controller is the first 2-SM controller proposed in the literature [162]. This algorithm has been named after the trajectory obtained in the phase plane, which perform rotations (twists) around the origin of the $\sigma - \dot{\sigma}$ plane, while converging to the origin itself in finite time. A schematic view of this behaviour is given in Fig. A.2.1. The absolute values of the intersections between the trajectory and the σ, $\dot{\sigma}$ axis are decreasing, as well as the time between two subsequent intersections. This decreasing behaviour can be described through a geometric progression [86].

The twisting controller is defined by:

$$u = -(r_1 \text{sign}(\sigma) + r_2 \text{sign}(\dot{\sigma})), \quad r_1 > r_2 > 0 \tag{A.2.5}$$

with the controller gains r_1, r_2 satisfying the following constraints:

$$\begin{aligned}(r_1 + r_2) &> \frac{(r_1 - r_2)K_M + 2C}{K_m} & \text{(A.2.6)}\\ (r_1 - r_2) &> \frac{C}{K_m} & \text{(A.2.7)}\end{aligned}$$

It is proven that convergence to the phase plane origin, and hence second order sliding, is obtained in finite time. Particularly, the maximum reaching time is [83]:

$$T \leq \sum \frac{|\dot{\sigma}_0|}{(1-q)|K_m(r_1 - r_2) - C|} \tag{A.2.8}$$

where $|\dot{\sigma}_0|$ is the absolute value of the first interception of the phase plane trajectory with the $\dot{\sigma}$ axis and $q < 1$ is the scale factor of the geometric sequence relative to

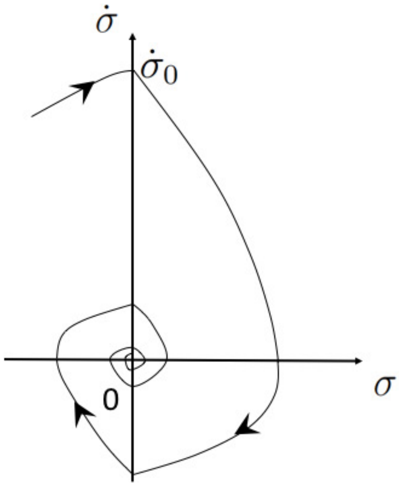


Figure A.2.1: Twisting algorithm trajectory in the phase plane - adapted from [84]

the amplitude of the axis interceptions, defined as:

$$q = \frac{K_M(r_1 - r_2) + C}{K_M(r_1 + r_2) - C} \tag{A.2.9}$$

For the proof of convergence and the detailed derivation of the convergence time, the reader is referred to [84].

The information requirement of the twisting controller is reduced with respect to typical second order sliding mode controllers, for which the knowledge of both $\sigma(t)$ and $\dot{\sigma}(t)$ is required. For the control definition in (A.2.5), in fact, it is sufficient to know the sign of both the sliding variable and its first derivative.

A.2.2 Suboptimal controller

The suboptimal control algorithm is named after its switching logic, which is inspired to the time-optimal control philosophy [53]. The suboptimal controller is given by:

$$u(t) = -r_1 \text{sign}\left(\sigma - \frac{\sigma^*}{2}\right) + r_2 \text{sign}(\sigma^*), \quad r_1 > r_2 > 0 \tag{A.2.10}$$

where σ^* refers to the value of $\sigma(t)$ at the last time instant when $\dot{\sigma}(t) = 0$ was verified. The initial value of σ^* is set to zero and it is updated any time the condition $\dot{\sigma}(t) = 0$ is verified. The finite time achievement of a second order sliding mode is guaranteed if the controller's gains satisfy the constraints [88]:

$$\begin{aligned} r_1 - r_2 &> \frac{C}{K_m} & \text{(A.2.11)} \\ r_1 + r_2 &> \frac{4C + K_M(r_1 - r_2)}{3K_m} & \text{(A.2.12)} \end{aligned}$$

The condition in (A.2.11) is the dominance condition, as it guarantees that the control has sufficient authority to affect the sign of $\ddot{\sigma}(t)$, while the condition in (A.2.12) is the convergence condition, as it guarantees that a second order sliding mode is obtained in finite time.

Two example trajectories of the suboptimal controller in the phase plane are shown in Fig. A.2.2. These represent the two behaviours of the system controlled through the suboptimal sliding mode control, which can be obtained with different values of the controller gains r_1, r_2. Particularly, the phase plane trajectory can perform a series of twisting rotations around the origin (case (a) in Fig. A.2.2), or monotonically converge to the origin without any change in the sign of $\sigma(t)$ (case (b) in Fig. A.2.2). Particularly, as mentioned in [87], the monotonic convergence condition is more restrictive than the one in (A.2.12) and results:

$$r_1 + r_2 > \frac{2C + K_M(r_1 - r_2)}{K_m} \tag{A.2.13}$$

The suboptimal algorithm requires knowledge of both $\sigma(t)$ and $\dot{\sigma}(t)$, necessary to verify if $\dot{\sigma}(t) = 0$ has occurred. In practical implementations, however, the detection of $\dot{\sigma}(t) = 0$ is based on the sign of $\sigma(t), \sigma(t-1), \sigma(t-2)$ and it is determined through a digital peak detector. In this way. the information requirement of the suboptimal sliding mode controller is reduced to the only knowledge of $\sigma(t)$.

A.2.3 Super-twisting controller

The super twisting algorithm has been initially developed for relative degree one systems as a chattering avoidance technique [86], but it can be applied to relative degree two systems in order to obtain a second order sliding mode. The trajectory in

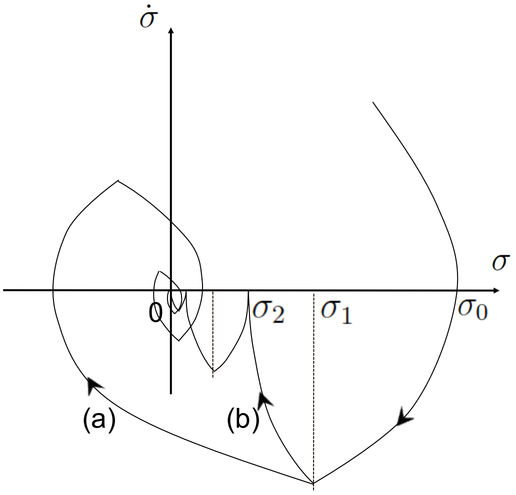


Figure A.2.2: Suboptimal algorithm trajectory in the phase plane - adapted from [84]

the phase plane is in this case characterized by twisting around the origin, similarly to the example trajectory in Fig. A.2.3. The super-twisting controller is constituted of two terms: a term defined through its discontinuous derivative ($u_1(t)$) and a function of the sliding variable ($u_2(t)$). The controller is formulated as:

$$u(t) = u_1(t) + u_2(t) \tag{A.2.14}$$

with:

$$\dot{u}_1(t) = \begin{cases} -u & \text{if } |u| > 1 \\ -W\text{sign}(\sigma) & \text{if } |u| \leq 1 \end{cases} \tag{A.2.15}$$

$$u_2(t) = -\lambda|\sigma|^\rho \text{sign}(\sigma) \tag{A.2.16}$$

The sufficient conditions for the achievement of second order sliding in finite time have been determined in [163] as:

$$W > \frac{C}{K_m} \tag{A.2.17}$$

$$\lambda^2 \geq \frac{4C}{K_m^2}\frac{K_M(W+C)}{K_m(W-C)} \tag{A.2.18}$$

$$0 < \rho \leq 0.5 \tag{A.2.19}$$

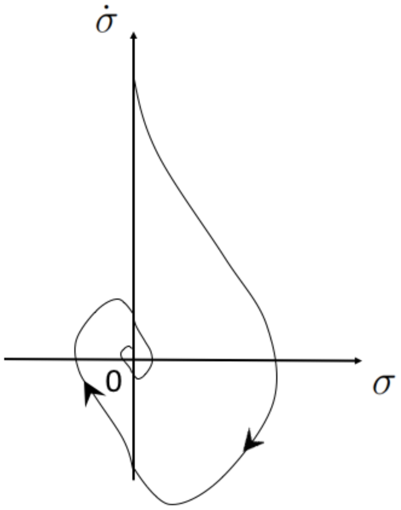


Figure A.2.3: Super-twisting algorithm trajectory in the phase plane - adapted from [84]

The most important characteristic of the super-twisting controller is its independence of the knowledge of the time derivative of the sliding variable $\dot{\sigma}(t)$ and of its sign. Consequently, even the super-twisting algorithm only requires the knowledge of $\sigma(t)$ and it is hence often chosen because of its low information requirements.

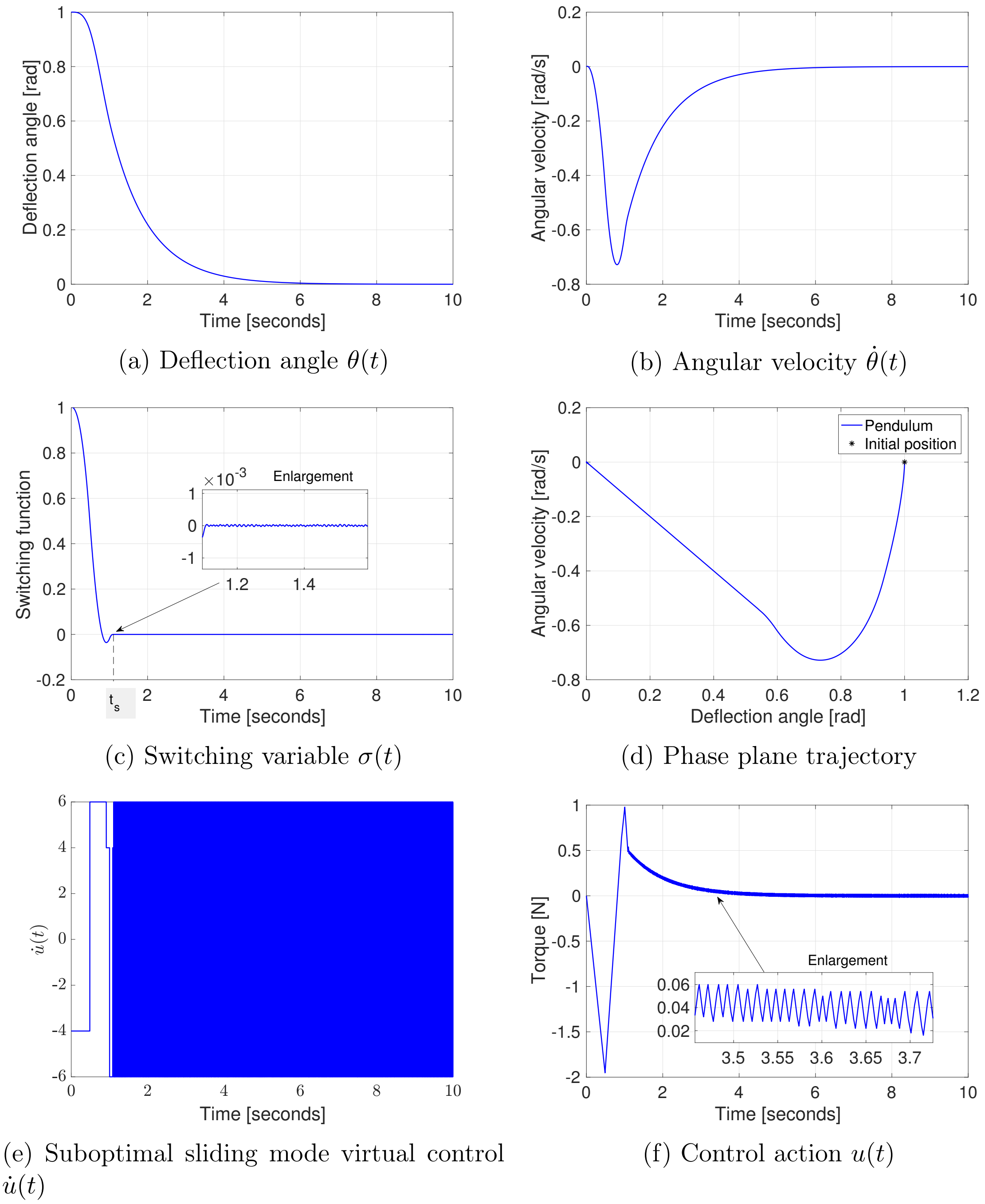
A.2.4 Quasi-continuous controller

The class of quasi-continuous sliding mode controllers has been proposed in [145], for systems with different relative degrees. For a relative degree two system, the quasi-continuous sliding mode controller results:

$$u(t) = -\alpha \frac{\dot{\sigma} + |\sigma|^{1/2} \text{sign}\sigma}{|\dot{\sigma}| + |\sigma|^{1/2}} \tag{A.2.20}$$

where $\alpha > 0$ is a design constant. This controller enforces a second order sliding mode in finite time. The parameter α is chosen specifically for any fixed C, K_m, K_M, typically by computer simulation, avoiding redundantly large estimations of C, K_m, K_M [145].

The controller in (A.2.20) has the peculiarity of being continuous everywhere except that on the two sliding manifold $\sigma(t) = \dot{\sigma}(t) = 0$. It is observed in [84] that the simultaneous fulfilment of the two exact equalities defining the sliding manifold



(a) Deflection angle $\theta(t)$

(b) Angular velocity $\dot{\theta}(t)$

(c) Switching variable $\sigma(t)$

(d) Phase plane trajectory

(e) Suboptimal sliding mode virtual control $\dot{u}(t)$

(f) Control action $u(t)$

Figure A.2.4: Suboptimal sliding mode control - pendulum example

would not happen in practice in the presence of any small noise or disturbance. Consequently, the controller in (A.2.20) is practically continuous everywhere.

The main limitation of the quasi-continuous controller is its high information requirement, as it depends on $\dot{\sigma}(t)$, which is often unknown in practice. If $\dot{\sigma}(t)$ is known or estimated, however, quasi-continuous sliding mode controllers are a valid alternative, thanks to their chattering reduction effect [145].

A.2.5 An illustrative example - continuation

The numerical example presented in Section A.1.3 is continued here. Particularly, a second order sliding mode controller is used here for the relative degree one system in (A.1.2) for chattering avoidance. The second order sliding mode controller is used to define a virtual control acting on the second derivative of the sliding variable $\ddot{\sigma}(t)$. The obtained virtual control $\dot{u}(t)$ is discontinuous, but the actual control $u(t)$, which is obtained through an integration step, results continuous, thus avoiding chattering.

The pendulum simulation has been set up as in Section A.1.3, and a suboptimal sliding mode virtual control $\dot{u}(t)$, obtained as in (A.2.10), has been used. The controller gains have been chosen as $r_1 = 5$, $r_2 = 1$. The obtained results are shown in Fig. A.2.4. The pendulum deflection and angular velocity are shown in Fig. A.2.4a and A.2.4b respectively and successfully converge to zero in finite time. The first interesting result is the evolution of the switching variable, shown in Fig. A.2.4c: even in this case, sliding is obtained in approximately 1 second. During sliding, the high frequency oscillations characterizing classic sliding mode control, which have been highlighted in Fig. A.1.2c, are avoided. The switching function results continuous and it shows some small amplitude oscillations around zero. The virtual control $\dot{u}(t)$ and the actual control $u(t)$ are shown in Fig. A.2.4e and A.2.4f respectively. The virtual control in Fig. A.2.4e shows chattering, with high frequency oscillations starting at the beginning of the sliding phase; this is similar to the behaviour of classic sliding mode controller shown in Fig. A.1.2e. The actual control applied to the pendulum $u(t)$, shown in Fig. A.2.4f, appears continuous. This demonstrates the efficacy of higher order sliding mode controllers as a chattering avoidance tool.

Appendix B

Complete model of an underwater glider

Additional details on the components of the complete model of an underwater glider, introduced in Chapter 4, are given in this appendix.

The complete model of the glider in (4.3.11) has been derived in [104] from first principles, by computing the total energy of the system and applying Newton's second law.

As a first step, the total kinetic energy of the system is computed, accounting for the contributions of the rigid body of the glider, the movable mass $\bar{m}$, the ballast mass m_B and the displaced fluid. The partial derivatives of the total kinetic energy are used to compute the momenta affecting the system. Particularly, the partial derivatives with respect to the vehicle's speed and angular velocity are used to find expressions for the total translational and angular momenta respectively, expressed in the body reference frame. By inverting the expression of the momenta, an explicit expression for the vehicle's speed and angular velocity is obtained; from these, a preliminary version of the model in (4.3.11) is obtained through a differentiation step. As an additional simplification, the original control acting on the position of the movable mass $\bar{m}$, which is a force control input, is transformed into an acceleration control input through a change of variables. Finally, the simplifying hypothesis enumerated in Section 4.3.4 are applied and the complete model in (4.3.11) is obtained.

The reader is referred to [104] for a step by step derivation of the glider model.

The elements of the complete glider model in (4.3.11) are described here. The total mass matrix of the system is obtained through:

$$M = m_H I_d + M_f \ , \tag{B.0.1}$$

where m_H is the hull mass, I_d is the identity matrix, and M_f is the added mass matrix, which accounts for the energy necessary to accelerate the surrounding fluid while the glider translates. The total inertia of the system, including the glider hull component (J_H) and the added mass component (J_f), results:

$$J = J_H + J_f \ . \tag{B.0.2}$$

The added mass inertia matrix J_f is due to the energy necessary to accelerate the fluid while the vehicle rotates and translates. Under the assumption of the glider being symmetric with respect to both the (e_1, e_3) and the (e_1, e_2) planes, introduced in Chapter 4, the added mass and inertia matrices are diagonal.

The complete formulation of the total torque T in (4.3.11) is:

$$\begin{aligned} T &= [J\Omega + \hat{r}_p(\bar{m}(v + \Omega \times r_p + \dot{r}_p))] \times \Omega + (\Omega \times r_p) \times (\bar{m}(v + \Omega \times r_p + \dot{r}_p)) \\ &\quad +(Mv \times v) + \bar{m}\hat{r}_p g R^\top k + T_{ext} - \hat{r}_p[H_{11}(Z_p + \omega_p) + H_{12}(-Z_b)] \qquad \text{(B.0.3)} \end{aligned}$$

while the complete expression of the total force F is:

$$\begin{aligned} F &= [(M + m_B I)v + \bar{m}(v + \Omega \times r_p + \dot{r}_p)] \times \Omega + m_0 g R^\top k + F_{ext} \qquad \text{(B.0.4)} \\ &\quad -[H_{11}(-Z_p + \omega_p) + H_{12}(-Z_b)] - [H_{21}(-Z_p + \omega_p) + H_{22}(-Z_b)] \end{aligned}$$

where H_i, Z_p and Z_b are terms accounting for the inclusion of the acceleration control on the movable mass $\bar{m}$. The change of coordinates, applied to transform the original force control input into an acceleration input, is performed by differentiating the expression of the position vector of the movable mass r_p, and of the ballast mass r_B, obtaining [104]:

$$\begin{aligned}\dot{r}_p &= \frac{1}{\bar{m}}P_p - v - \Omega \times r_p && \text{(B.0.5)}\\ \dot{r}_B &= \frac{1}{m_B}P_B - v - \Omega \times r_B && \text{(B.0.6)}\end{aligned}$$

where P_p and P_B are the movable mass and the ballast mass momenta. Differentiating (B.0.5), the following is obtained:

$$\begin{bmatrix}\ddot{r}_p\\ \ddot{r}_B\end{bmatrix} = Z(x) + H^{-1}(x)\begin{bmatrix}u_p\\ u_B\end{bmatrix} \qquad \text{(B.0.7)}$$

where $H^{-1}(x)$ is the control vector field, $Z(x)$ is the drift vector field, u_p is the input force applied to the movable mass and u_B is the force applied to the ballast mass in order to keep it fixed at $r_B = 0$. Inverting (B.0.7), an explicit relation between the acceleration input and the original input can be obtained:

$$\begin{bmatrix}u_p\\ u_B\end{bmatrix} = H\left(\begin{bmatrix}\ddot{r}_p\\ \ddot{r}_B\end{bmatrix} - Z(x)\right) = H\left(\begin{bmatrix}\omega_p\\ 0\end{bmatrix} - \begin{bmatrix}Z_p\\ Z_b\end{bmatrix}\right) \qquad \text{(B.0.8)}$$

In detail, the drift vector elements are [104]:

$$\begin{aligned}Z_p &= -M^{-1}[[(M + m_B I)v + \bar{m}(v + \Omega \times r_p + \dot{r}_p)] \times \Omega + m_0 g R^\top k + F_{ext}] && \text{(B.0.9)}\\ &\quad -\Omega \times \dot{r}_p + T_{ext} - J^{-1}[(J\Omega + \hat{r}_p(\bar{m}(v + \Omega \times r_p + \dot{r}_p))) \times \Omega + (Mv \times v)\\ &\quad +(\Omega \times r_p) \times (\bar{m}(v + \Omega \times r_p + \dot{r}_p)) + \bar{m}\hat{r}_p g R^\top k] \times r_p\\ Z_b &= -M^{-1}[[(M + m_B I)v + \bar{m}(v + \Omega \times r_p + \dot{r}_p)] \times \Omega + m_0 g R^\top k + F_{ext}] - \Omega \times \dot{r}_p\end{aligned}$$

and

$$H = \begin{bmatrix} M^{-1} - \hat{r}_p J^{-1} \hat{r}_p + \frac{1}{\bar{m}} I & M^{-1} \\ M^{-1} & M^{-1} + \frac{1}{m_B} I \end{bmatrix}^{-1} \tag{B.0.10}$$

The external forces F_{ext} and moments T_{ext} are expressed in the body reference frame. These terms account for the contribution of the hydrodynamic forces and moments. The hydrodynamic forces are typically expressed in the wind reference frame and can be converted to the body frame through the rotation matrix R_{WB} [105]. In particular:

$$F_{ext} = R_{WB} \begin{bmatrix} -D & F_S & -L \end{bmatrix}^\top \tag{B.0.11}$$

where D is the drag force, F_S is the side force and L is the lift force. The hydrodynamic moments with respect to the vehicle body reference frame are:

$$T_{ext} = \begin{bmatrix} M_{DL1} & M_{DL2} & M_{DL3} \end{bmatrix}^\top \tag{B.0.12}$$

In defining the hydrodynamic forces, the effects of the movable rudder should be introduced. In literature, there are only a few examples of glider rudder effects analysis [107], [164]. The main beneficial effect of controlling the vehicle's direction of movement in the lateral plane through a rudder is that the relationship between the rudder deflection angle and the heading deflection is independent of the direction of movement in the vertical plane [104]. With the rudder inclusion from [107], where the rudder deflection is assumed affecting the drag force D, the side force F_S and the moment in the e_3 direction M_{DL3}, the effect of the rudder when moving upwards and downwards results reversed. Consequently, classical aircraft theory has been revised from [105], together with the glider modelling strategy ins [164], in which the effects of a movable rudder on the hydrodynamic forces are considered. In this works, the rudder deflection has been shown to influence the side force F_S and the moments in the e_1 and e_3 directions, M_{DL1} and M_{DL3} respectively. The complete coefficient based expressions for the hydrodynamic forces and moments thus results:

$$\begin{aligned} D &= (K_{D0} + K_D\alpha^2)V^2 && \text{(B.0.13)} \\ F_S &= K_\beta \beta V^2 + K_{\delta_{R\beta}}\delta_R V^2 && \text{(B.0.14)} \\ L &= (K_{L0} + K_L\alpha)V^2 && \text{(B.0.15)} \\ M_{DL1} &= K_{MR}\beta V^2 + K_{Q1}\Omega_1 V^2 + K_{\delta R}\delta_R V^2 && \text{(B.0.16)} \\ M_{DL2} &= (K_{M0} + k_M\alpha + K_{Q2}\Omega_2)V^2 && \text{(B.0.17)} \\ M_{DL3} &= K_{MY}\beta V^2 + K_{Q3}\Omega_3 V^2 + K_{\delta_{R_M}}\delta_R V^2 && \text{(B.0.18)} \end{aligned}$$

where $K_{\delta_{R\beta}}, K_{\delta R}, K_{\delta_{R_M}}$ are the hydrodynamic coefficients corresponding to the rudder position.

Appendix C

Gaussian process models

A Gaussian Process (GP) is a probabilistic model used to describe the probability distribution of unknown functions [126]. A GP is a statistical distribution, characterized by a mean function $\mu(\boldsymbol{x})$ and a covariance function $c(\boldsymbol{x}, \boldsymbol{x}')$:

$$\chi(\boldsymbol{x}) \sim GP(\mu(\boldsymbol{x}), c(\boldsymbol{x}, \boldsymbol{x}')). \tag{C.0.1}$$

In this appendix, some detail about the mean and covariance functions used in this thesis is given. Additionally, the model fitting and the gradient estimation methods introduced in Chapter 6 are demonstrated on a numerical example.

C.1 Mean function

The mean function describes the global trend of the considered data set, and it is often a polynomial function of the input. Let $H_i(\boldsymbol{x}), i = 1, \ldots, p$ be the basis functions. The mean function can be defined as:

$$\mu(\boldsymbol{x}) = \sum_{i=1}^{p} \beta_i H_i(\boldsymbol{x}) \tag{C.1.1}$$

where β_i are the constant coefficients constituting the mean function hyperparameters. A first order polynomial mean function results:

$$\mu(\boldsymbol{x}) = \beta_0 + \beta_1 x + \beta_2 y \tag{C.1.2}$$

while a second order polynomial mean function results:

$$\mu(\boldsymbol{x}) = \beta_0 + \beta_1 x + \beta_2 y + \beta_3 x^2 + \beta_4 y^2 + \beta_5 xy \tag{C.1.3}$$

C.2 Covariance function

The covariance function describes the local trend of the dataset. The covariance function describes the correlation between the output and the input configuration. It is used to describe the smoothness and the stationarity properties of the dataset. The covariance functions considered in this thesis are assumed stationary. A stationary covariance function is a function of $|\boldsymbol{x} - \boldsymbol{x}'|$, where $\boldsymbol{x}$ and $\boldsymbol{x}'$ are two input configuration [128]. This means that the covariance function $c(\boldsymbol{x}, \boldsymbol{x}')$ can be formulated as:

$$c(\boldsymbol{x}, \boldsymbol{x}') = \sigma_{GP}^2 \text{cor } (\boldsymbol{x} - \boldsymbol{x}') \tag{C.2.1}$$

where the hyperparameter σ_{GP}^2 is the variance, constant everywhere, and cor $(\boldsymbol{x} - \boldsymbol{x}')$ is the correlation function. Consequently, a stationary covariance function is invariant to translational movement in the input space and it is a function of solely the distance between two inputs, thus verifying:

$$c(\boldsymbol{x}, \boldsymbol{x}') = c(\boldsymbol{x} + d, \boldsymbol{x}' + d) \tag{C.2.2}$$

where d is a certain displacement. Furthermore, a covariance function is isotropic if it is a function only of $|\boldsymbol{x} - \boldsymbol{x}'|$. In this case, the covariance function results invariant to any rigid motion.

Several choices of covariance functions are available [128]. A non exhaustive list is given here:

- Squared exponential covariance function

$$c(\boldsymbol{x}, \boldsymbol{x}') = \sigma_{GP}^2 e^{-\frac{h^2}{\phi^2}} \tag{C.2.3}$$

- Exponential covariance function

$$c(\boldsymbol{x}, \boldsymbol{x}') = \sigma_{GP}^2 e^{-\frac{h}{\phi}} \tag{C.2.4}$$

- Powered exponential covariance function

$$c(\boldsymbol{x}, \boldsymbol{x}') = \sigma_{GP}^2 e^{-\frac{h^p}{\phi^p}} \tag{C.2.5}$$

- Matérn covariance function

$$c(\boldsymbol{x}, \boldsymbol{x}') = \sigma_{GP}^2 \frac{\left(2\sqrt{\nu}\frac{h}{\phi}\right)^{\nu}}{2^{\nu-1}\Gamma(\nu)} K_{\nu}\left(2\sqrt{\nu}\frac{h}{\phi}\right) \tag{C.2.6}$$

- Matérn covariance function with $\nu = \frac{3}{2}$

$$c(\boldsymbol{x}, \boldsymbol{x}') = \sigma_{GP}^2 \left(1 + \sqrt{3}\frac{h}{\phi}\right) e^{-\sqrt{3}\frac{h}{\phi}} \tag{C.2.7}$$

- Matérn covariance function with $\nu = \frac{3}{2}$

$$c(\boldsymbol{x}, \boldsymbol{x}') = \sigma_{GP}^2 \left(1 + \sqrt{5}\frac{h}{\phi} + \frac{5h^2}{3\phi^2}\right) e^{-\sqrt{5}\frac{h}{\phi}} \tag{C.2.8}$$

where h is the distance between the input configurations $\boldsymbol{x}$ and $\boldsymbol{x}'$, $\nu > 0$ is a design parameter, $K_{\nu}(\cdot)$ is a modified Bessel function of the second order and $\Gamma(\nu) = (\nu-1)!$ is the Gamma function [128]. A graphical comparison of the covariance functions with $\sigma_{GP}^2 = 1$, $\phi = 0.25$ is shown in Fig. C.2.1.

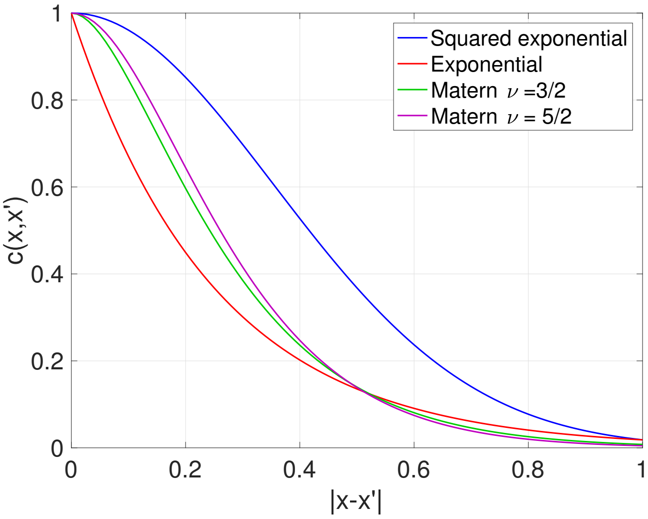


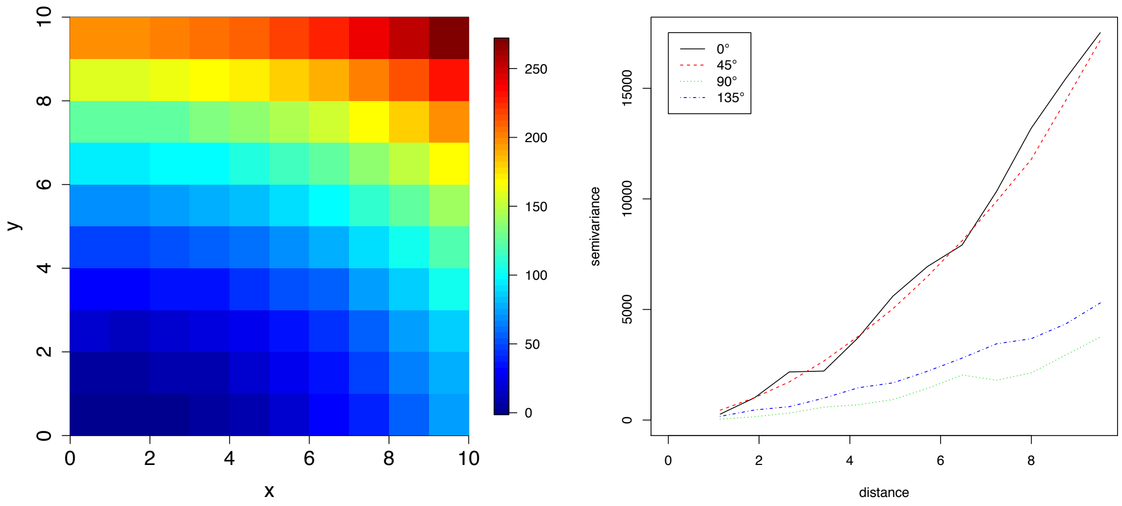
Figure C.2.1: Graphical comparison of covariance functions

C.3 A numerical example

The following numerical function is considered:

$$A(x, y) = x^2 + 2y^2 - 3x \tag{C.3.1}$$

The available forecast information relative to the function in (C.3.1) is defined over a grid region $(x, y) \in [0,\ 10] \times [0,\ 10]$ at 100 locations. In order to fit a probabilistic model to an uncertain forecast model (preliminary information), white noise is added to ten randomly chosen observations in the original dataset. The added noise is distributed as in (6.3.7), with $\sigma_n^2 = 1$. The obtained dataset is shown in Fig. C.2.2a. Before fitting the probabilistic model, the structure of the mean and covariance functions is chosen. A useful method for the choice of the mean function is the evaluation of the isotropy characteristics of the dataset. A dataset is said to be isotropic if the variations are a function of solely the distance between points, and not of the direction of movement [128]. Isotropy is a desirable characteristic because the Gaussian Process fitted to an isotropic dataset is simplified, having an univariate covariance function. The anisotropy associated with the dataset in C.2.2 can be observed from the directional variograms (graphs showing the spatial correlation between data as a function of the separation distance) shown in Fig. C.2.2b. As the



(a) Constant trend

(b) Directional variograms

Figure C.2.2: Numerical example - preliminary dataset

Parameter	β_0	β_1	β_2	β_3	β_4	β_5	σ^2_{GP}	ϕ	σ^2_n
SE	0.137	−3.065	−0.032	1.006	2.003	−0.002	0.047	0.763	0.057
Matérn	0.137	−3.065	−0.032	1.006	2.003	−0.002	0.104	0.239	0.000

Table C.3.1: Numerical example - distribution parameters

variograms are very different from one another, the correlation between data results highly dependent on the direction of movement. This anisotropy can be corrected through the introduction of a trend, which defines the mean part of the fitted model [147]. Fig. C.3.1 shows the directional variograms of the forecast dataset obtained introducing a first order and a second order trend. Data can be judged isotropic when the directional variograms have similar characteristics in terms of nugget, sill and overall behaviour. The dataset which better verifies this condition is the one obtained introducing a second order trend. Consequently, the mean part of the model is chosen as in (C.1.3). Hence, the mean part of the GP is a function of five hyperparameters: $\beta_1, \ldots, \beta_5$.

The obtained GP is characterized as:

$$\chi(\boldsymbol{x}) = \underbrace{\beta_0 + \beta_1 x + \beta_2 y + \beta_3 x^2 + \beta_4 y^2 + \beta_5 xy}_{\mu(\boldsymbol{x})} + Z(\boldsymbol{x}) + \epsilon(\boldsymbol{x}) \ , \tag{C.3.2}$$

where the GP $Z(\boldsymbol{x})$ is modelled through two different covariance functions: the

Metric	Maximized likelihood $\hat{L}$	BIC	AIC
Squared exponential distribution	-34.3	111.8	86.61
Matérn distribution	-34.32	111.8	86.64

Table C.3.2: Numerical example - validation metrics for the comparison of the two models fitted through maximum likelihood

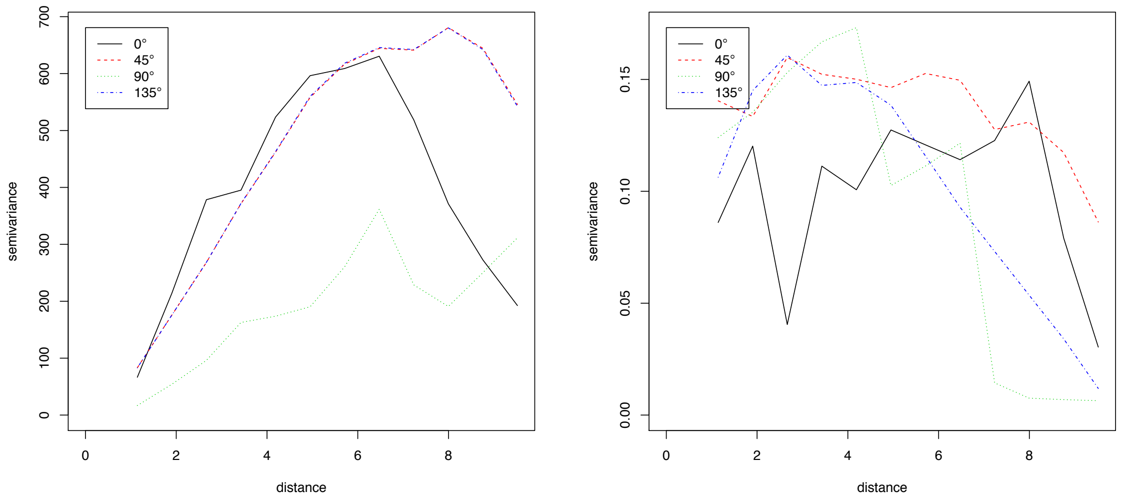
	β_0	β_1	β_2	β_3	β_4	β_5	σ^2_{GP}	ϕ	σ^2_n
Mean	0.135	-3.064	-0.033	1.006	2.003	-0.001	0.042	0.965	2.2215
Median	0.139	-3.062	-0.032	1.006	2.003	-0.001	0.0369	0.950	2.200
Mode	0.136	-3.065	-0.032	1.006	2.003	-0.002	0.0927	0.800	2.400

Table C.3.3: Numerical example - hyperparameters posterior distribution for the model fitted through Bayesian estimation

SE covariance function introduced in (C.2.3) and the Matérn covariance function in (C.2.7). With these choices, two additional hyperparameters are introduced: σ^2_{GP} and ϕ. The final hyperparameter to be estimated is the variance of the white noise affecting the dataset, modelled through $\epsilon(\boldsymbol{x})$: σ^2_n. The values of the hyper parameters of the model in (C.3.2) are obtained in the two cases through the maximum likelihood strategy described in Section 6.3.1, which is solved in the statistic friendly software 'R' [149]. The obtained values for the hyperparameters have been collected in Table C.3.1.

In order to choose the best fitting model, some of the validation metrics introduced in Section 6.3.3 are considered. Particularly, as the models are fitted through the Maximum Likelihood procedure, the maximized likelihood, the BIC and the AIC are compared. The obtained values for the validation metrics are given in Table C.3.2. As mentioned in Section 6.3.3, the model characterized by the highest maximized likelihood should be preferred; additionally, models characterized by smaller values of the BIC and the AIC criteria should be preferred. Consequently, even if the differences are in this case minor, the chosen covariance function is the squared exponential covariance function.

In order to fit the same model though the Bayesian approach described in Section 6.3.2, the prior distribution of the hyperparameters needs to be defined. Particularly, a non informative prior is defined for each hyperparameter, as the mean hyperparameters are described through a flat prior distribution, while the squared



(a) 1^{st} order trend

(b) 2^{nd} order trend

Figure C.3.1: Numerical example - directional variograms with different trend structures.

exponential covariance function hyperparameters are described as reciprocal distributions. The posterior distributions of the hyperparameters are shown in Fig. C.3.2. Additional information about the posterior distribution is given in Table C.3.3.

The external validation metrics introduced in Section 6.3.3 are used in order to select the best fitting model. Four external validation datasets, composed of 121, 196, 441 and 529 observations respectively are built and used for the external validation. The validation metrics are graphically compared in Fig. C.3.3, where m indicates the number of observations in the validation set. The results obtained from a maximum likelihood SE model are represented as red circles, the results from a maximum likelihood Matérn distribution are represented as blue squares and the results from a Bayesian SE distribution are represented as green triangles. Each model performs extremely well in the evaluation of the mean, the variance and the exceedence probability. Consequently, the choice of the best fitting model is entirely based on the RMSE; for each of the validation dataset, the model showing the lowest RMSE is the squared exponential distribution obtained through the Bayesian approach.

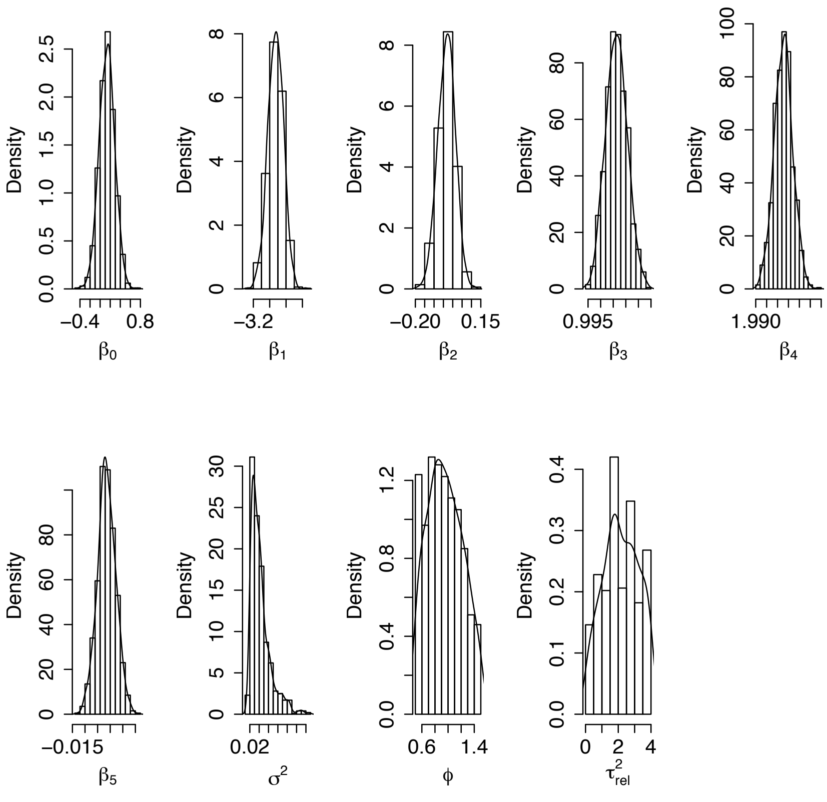


Figure C.3.2: Posterior distribution of the hyperparameters

C.3.1 Gradient estimate construction example

Since the analytical expression in (C.3.1) represents the field and is known, the gradient at each desired location can be evaluated analytically. Specifically, each hyperparameter is chosen to be equal to the mean of the posterior distribution, as given in Table C.3.3. The partial derivatives of the GP in (C.3.2) at each position $k = 1, \dots, 3$ result:

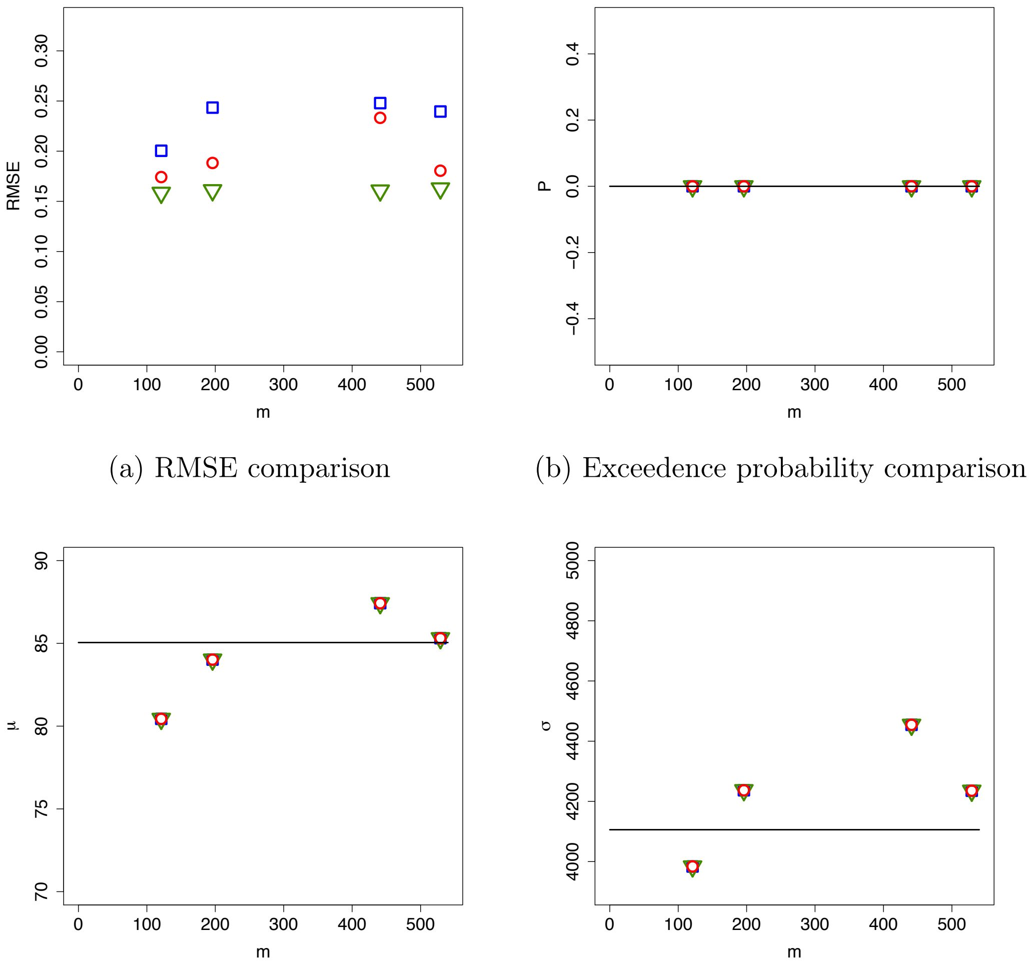
$$\left(\frac{\partial \chi(\boldsymbol{x})}{\partial x}\right)_k = \beta_1 + 2\beta_3 x_k + \beta_5 y_k + Z'_{x,k}(\boldsymbol{x}) \tag{C.3.3}$$

$$\left(\frac{\partial \chi(\boldsymbol{x})}{\partial y}\right)_k = \beta_2 + 2\beta_4 y_k + \beta_5 x_k + Z'_{y,k}(\boldsymbol{x}) \tag{C.3.4}$$

where the GPs $Z'_{x,k}(\boldsymbol{x})$ and $Z'_{y,k}(\boldsymbol{x})$ are defined as:

$$Z'_{x,k}(\boldsymbol{x}) \sim GP\left(0, -\sigma^2_{GP} e^{-\frac{h^2}{\phi^2}} \left(\frac{2}{\phi^2}\right)(x - x_k)\right) \tag{C.3.5}$$

$$Z'_{y,k}(\boldsymbol{x}) \sim GP\left(0, -\sigma^2_{GP} e^{-\frac{h^2}{\phi^2}} \left(\frac{2}{\phi^2}\right)(y - y_k)\right) \tag{C.3.6}$$



(a) RMSE comparison

(b) Exceedence probability comparison

(c) Distribution mean comparison

(d) Distribution variance comparison

Figure C.3.3: External validation results

Results from three different validation cases are provided in Table C.3.4. For each case, three different positions in the grid P_1, P_2 and P_3 are considered, and the assumption is that, at these locations, the measured values are taken from the noise added data base. The gradient at the centroid position is estimated using the surrogate model defined in (C.3.2), through the procedure discussed in Section 6.4. Particularly, the gradient is firstly estimated at each of the three vehicles' position and the gradient at the centroid position is then estimated through an averaging step. The accuracy of the estimation procedure can be appreciated from the comparison between the estimated and the analytically computed gradient values given in Table C.3.4. Additionally, the beneficial effect of the averaging procedure can be highlighted. Considering, for instance, Case 3 in Table C.3.4, the gradient at

	Case 1	**Case 2**	**Case 3**
P_1 (x_1, y_1)	$(7, 1)$	$(2, 3)$	$(1, 5)$
P_2 (x_2, y_2)	$(2, 4)$	$(3, 5)$	$(2, 2)$
P_3 (x_3, y_3)	$(6, 9)$	$(2, 6)$	$(3, 4)$
Centroid (x_c, y_c)	$(5, 4.67)$	$(2.33, 4.67)$	$(2, 3.67)$
Analytically computed gradient	$(7, 18.67)$	$(1.67, 18.67)$	$(1, 14.67)$
Estimated gradient	$(6.99, 18.66)$	$(1.64, 18.67)$	$(0.97, 14.67)$

Table C.3.4: Numerical example - gradient estimation

the centroid position can be estimated applying a single differentiation to the GP in (C.3.2), evaluated at (x_c, y_c). In this way, the partial derivative estimates at the centroid position result:

$$\left(\frac{\partial\chi}{\partial x}, \frac{\partial\chi}{\partial y}\right)|_{(x_c,y_c)} = (0.9545, 14.6510) \ , \tag{C.3.7}$$

while the estimate obtained through the averaging step is:

$$\left(\frac{\partial\chi}{\partial x}, \frac{\partial\chi}{\partial y}\right)|_{(x_c,y_c)} = (0.9626, 14.6535) \ , \tag{C.3.8}$$

Even if the difference between the estimates is minor in this case, the estimate better approximating the real value of the spatial gradient at the centroid position is the one obtained through the averaging procedure. This is because the effect of noise on the collected measurements is smoothed and the effect of spatial correlation is accounted for in the averaging procedure.

Bibliography

[1] M. Dunbabin and L. Marques, "Robots for environmental monitoring: Significant advancements and applications," *IEEE Robotics & Automation Magazine*, vol. 19, no. 1, pp. 24–39, 2012.

[2] G. Muscato, F. Bonaccorso, L. Cantelli, D. Longo, and C. D. Melita, "Volcanic environments: robots for exploration and measurement," *IEEE Robotics & Automation Magazine*, vol. 19, no. 1, pp. 40–49, 2012.

[3] X. Jin and A. Ray, "Coverage control of autonomous vehicles for oil spill cleaning in dynamic and uncertain environments," in *Proceedings of the American Control Conference (ACC)*, pp. 2594–2599, 2013.

[4] C. Ojeda-Martínez, F. G. Casalduero, J. T. Bayle-Sempere, C. B. Cebrian, C. Valle, J. L. Sanchez-Lizaso, A. Forcada, P. Sanchez-Jerez, P. Martín-Sosa, J. M. Falcón, *et al.*, "A conceptual framework for the integral management of marine protected areas," *Ocean & Coastal Management*, vol. 52, no. 2, pp. 89–101, 2009.

[5] ASV, "C-Enduro Long endurance ASV," *Technical Report*, 2015.

[6] J. Manley and S. Willcox, "The wave glider: A persistent platform for ocean science," in *OCEANS 2010*, pp. 1–5, 2010.

[7] L. Willett, "AutoNaut remote USV aims to make waves," *Jane's International Defence Review*, vol. 47, pp. 36–37, 2014.

[8] M. Purcell, C. Von Alt, B. Allen, T. Austin, N. Forrester, R. Goldsborough, and R. Stokey, "New capabilities of the Remus autonomous underwater vehicle," in *OCEANS*, vol. 1, pp. 147–151, 2000.

[9] P. Collar and S. McPhail, "Autosub: an autonomous unmanned submersible for ocean data collection," *Electronics & Communication Engineering Journal*, vol. 7, no. 3, pp. 105–114, 1995.

[10] C. C. Eriksen, T. J. Osse, R. D. Light, T. Wen, T. W. Lehman, P. L. Sabin, J. W. Ballard, and A. M. Chiodi, "Seaglider: A long-range autonomous underwater vehicle for oceanographic research," *IEEE Journal of Oceanic Engineering*, vol. 26, no. 4, pp. 424–436, 2001.

[11] D. C. Webb, P. J. Simonetti, and C. P. Jones, "SLOCUM: An underwater glider propelled by environmental energy," *IEEE Journal of Oceanic Engineering*, vol. 26, no. 4, pp. 447–452, 2001.

[12] J. Sherman, R. E. Davis, W. Owens, and J. Valdes, "The autonomous underwater glider "Spray"," *IEEE Journal of Oceanic Engineering*, vol. 26, no. 4, pp. 437–446, 2001.

[13] F. Zhang, J. Wang, J. Thon, C. Thon, E. Litchman, and X. Tan, "Gliding robotic fish for mobile sampling of aquatic environments," in *Proceedings of the 11th International Conference on Networking, Sensing and Control (ICNSC)*, pp. 167–172, 2014.

[14] H. K. White, P.-Y. Hsing, W. Cho, T. M. Shank, E. E. Cordes, A. M. Quattrini, R. K. Nelson, R. Camilli, A. W. Demopoulos, C. R. German, *et al.*, "Impact of the deepwater horizon oil spill on a deep-water coral community in the Gulf of Mexico," *Proceedings of the National Academy of Sciences*, vol. 109, no. 50, pp. 20303–20308, 2012.

[15] U. R. Sumaila, A. M. Cisneros-Montemayor, A. Dyck, L. Huang, W. Cheung, J. Jacquet, K. Kleisner, V. Lam, A. McCrea-Strub, W. Swartz, *et al.*, "Impact

of the deepwater horizon well blowout on the economics of US Gulf fisheries," *Canadian Journal of Fisheries and Aquatic Sciences*, vol. 69, no. 3, pp. 499–510, 2012.

[16] S. E. Anderson, J. Franko, E. Lukomska, and B. Meade, "Potential immunotoxicological health effects following exposure to COREXIT 9500A during cleanup of the deepwater horizon oil spill," *Journal of Toxicology and Environmental Health, Part A*, vol. 74, no. 21, pp. 1419–1430, 2011.

[17] X. Jin and A. Ray, "Navigation of autonomous vehicles for oil spill cleaning in dynamic and uncertain environments," *International Journal of Control*, vol. 87, no. 4, pp. 787–801, 2014.

[18] S. C. Y. Leong, P. Tkalich, and N. M. Patrikalakis, "Monitoring harmful algal blooms in Singapore: developing a HABs observing system," in *OCEANS 2012*, pp. 1–5, 2012.

[19] A. J. Sørensen and M. Ludvigsen, "Towards integrated autonomous underwater operations," *IFAC*, vol. 48, no. 2, pp. 107–118, 2015.

[20] C. J. Cannell, A. S. Gadre, and D. J. Stilwell, "Boundary tracking and rapid mapping of a thermal plume using an autonomous vehicle," in *OCEANS 2006*, pp. 1–6, 2006.

[21] M. P. Brito, G. Griffiths, and P. Challenor, "Risk analysis for autonomous underwater vehicle operations in extreme environments," *Risk Analysis*, vol. 30, no. 12, pp. 1771–1788, 2010.

[22] R. N. Smith, A. Pereira, Y. Chao, P. P. Li, D. A. Caron, B. H. Jones, and G. S. Sukhatme, "Autonomous underwater vehicle trajectory design coupled with predictive ocean models: a case study," in *Proceedings of the International Conference on Robotics and Automation*, pp. 4770–4777, 2010.

[23] B. Bayat, N. Crasta, A. Crespi, A. M. Pascoal, and A. Ijspeert, "Environmental monitoring using autonomous vehicles: a survey of recent searching techniques," *Current Opinion in Biotechnology*, vol. 45, pp. 76–84, 2017.

[24] R. N. Smith, M. Schwager, S. L. Smith, B. H. Jones, D. Rus, and G. S. Sukhatme, "Persistent ocean monitoring with underwater gliders: Adapting sampling resolution," *Journal of Field Robotics*, vol. 28, no. 5, pp. 714–741, 2011.

[25] J. Kim, P. P. Menon, J. Back, and H. Shim, "Disturbance observer based boundary tracking for environment monitoring," *Journal of Electrical Engineering & Technology*, vol. 12, no. 3, pp. 1299–1306, 2017.

[26] C. Barat and M.-J. Rendas, "Benthic boundary tracking using a profiler sonar," in *Proceedings of the International Conference on Intelligent Robots and Systems*, vol. 1, pp. 830–835, 2003.

[27] P. N. Denbigh, "Swath bathymetry: Principles of operation and an analysis of errors," *IEEE Journal of Oceanic Engineering*, vol. 14, no. 4, pp. 289–298, 1989.

[28] S. Pang, J. Farrell, R. Arrieta, and W. Li, "AUV reactive planning: deepest point," in *OCEANS*, vol. 4, pp. 2222–2226, 2003.

[29] J. Farrell, S. P. S. Pang, and W. L. W. Li, "Chemical plume tracing via an autonomous underwater vehicle," *Journal of Oceanic Engineering*, vol. 30, no. 2, pp. 428–442, 2005.

[30] D. Marthaler and A. L. Bertozzi, "Collective motion algorithms for determining environmental boundaries," in *SIAM Conference on Applications of Dynamical Systems*, pp. 1–15, 2003.

[31] M. A. Sundermeyer and J. R. Ledwell, "Lateral dispersion over the continental shelf: Analysis of dye release experiments," *Journal of Geophysical Research: Oceans (1978–2012)*, vol. 106, no. C5, pp. 9603–9621, 2001.

[32] S. T. Gille, K. Speer, J. R. Ledwell, and A. C. Naveira Garabato, “Mixing and stirring in the Southern Ocean,” *Eos, Transactions American Geophysical Union*, vol. 88, no. 39, pp. 382–383, 2007.

[33] J. G. Bellingham, “New oceanographic uses of autonomous underwater vehicles,” *Marine Technology Society Journal*, vol. 31, no. 3, pp. 34–47, 1997.

[34] A. Joshi, T. Ashley, Y. R. Huang, and A. L. Bertozzi, “Experimental validation of cooperative environmental boundary tracking with on-board sensors,” in *Proceedings of the American Control Conference (ACC)*, pp. 2630–2635, 2009.

[35] A. S. Matveev, H. Teimoori, and A. V. Savkin, “Method for tracking of environmental level sets by a unicycle-like vehicle,” *Automatica*, vol. 48, no. 9, pp. 2252–2261, 2012.

[36] P. Lanillos, J. J. Ruz, G. Pajares, and M. Jesus, “Environmental surface boundary tracking and description using a UAV with vision,” in *IEEE Conference on Emerging Technologies & Factory Automation*, pp. 1–4, 2009.

[37] D. Baronov and J. Baillieul, “Reactive exploration through following isolines in a potential field,” in *Proceedings of the American Control Conference (ACC)*, pp. 2141–2146, 2007.

[38] D. Baronov and J. Baillieul, “Autonomous vehicle control for ascending/descending along a potential field with two applications,” in *Proceedings of the American Control Conference (ACC)*, pp. 678–683, 2008.

[39] P. P. Menon, C. Edwards, Y. B. Shtessel, D. Ghose, and J. Haywood, “Boundary tracking using a suboptimal sliding mode algorithm,” in *Proceedings of the 53rd Conference on Decision and Control (CDC)*, pp. 5518–5523, 2014.

[40] M. Kemp, A. L. Bertozzi, and D. Marthaler, “Multi-UUV perimeter surveillance,” in *Proceedings of the 2004 IEEE/OES Workshop on Autonomous Underwater Vehicles*, pp. 102–107, 2004.

[41] Z. Jin and A. L. Bertozzi, "Environmental boundary tracking and estimation using multiple autonomous vehicles," in *Proceedings of the 46th Conference on Decision and Control (CDC)*, pp. 4918–4923, 2007.

[42] C. H. Hsieh, Z. Jin, D. Marthaler, B. Q. Nguyen, D. J. Tung, A. L. Bertozzi, and R. M. Murray, "Experimental validation of an algorithm for cooperative boundary tracking," in *Proceedings of the American Control Conference (ACC)*, vol. 2, pp. 1078–1083, 2005.

[43] J. Clark and R. Fierro, "Cooperative hybrid control of robotic sensors for perimeter detection and tracking," in *Proceedings of the American Control Conference (ACC)*, pp. 3500–3505, 2005.

[44] J. Clark and R. Fierro, "Mobile robotic sensors for perimeter detection and tracking," *ISA Transactions*, vol. 46, no. 1, pp. 3–13, 2007.

[45] P. P. Menon and D. Ghose, "Simultaneous source localization and boundary mapping for contaminants," in *Proceedings of the American Control Conference (ACC)*, pp. 4174–4179, 2012.

[46] P. P. Menon and D. Ghose, "Boundary mapping of 3-dimensional regions," in *Proceedings of the American Control Conference (ACC)*, pp. 2984–2989, 2013.

[47] D. Marthaler and A. L. Bertozzi, "Tracking environmental level sets with autonomous vehicles," in *Recent developments in cooperative control and optimization*, pp. 317–332, Springer, 2004.

[48] S. Susca, F. Bullo, and S. Martínez, "Monitoring environmental boundaries with a robotic sensor network," *IEEE Transactions on Control Systems Technology*, vol. 16, no. 2, pp. 288–296, 2008.

[49] F. Zhang and N. E. Leonard, "Cooperative filters and control for cooperative exploration," *IEEE Transactions on Automatic Control*, vol. 55, no. 3, pp. 650–663, 2010.

[50] W. Wu and F. Zhang, "Cooperative exploration of level surfaces of three dimensional scalar fields," *Automatica*, vol. 47, no. 9, pp. 2044–2051, 2011.

[51] G. Indiveri, "Kinematic time-invariant control of a 2D nonholonomic vehicle," in *Proceedings of the 38th Conference on Decision and Control (CDC)*, pp. 2112–2117, 1999.

[52] J. Guldner and V. Utkin, "The chattering problem in sliding mode systems," in *Proceedings of the 14th International Symposium on Mathematical Theory of Networks and Systems (MTNS)*, vol. 11, 2000.

[53] G. Bartolini, A. Ferrara, and E. Usani, "Chattering avoidance by second-order sliding mode control," *IEEE Transactions on Automatic Control*, vol. 43, no. 2, pp. 241–246, 1998.

[54] M. Basseville and I. V. Nikiforov, *Detection of abrupt changes: theory and application*, vol. 104. Prentice Hall Englewood Cliffs, 1993.

[55] K. Krishnanand and D. Ghose, "Glowworm swarm optimization for simultaneous capture of multiple local optima of multimodal functions," *Swarm intelligence*, vol. 3, no. 2, pp. 87–124, 2009.

[56] M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: Active contour models," *International Journal of Computer Vision*, vol. 1, no. 4, pp. 321–331, 1988.

[57] C. G. Mayhew, R. G. Sanfelice, and A. R. Teel, "Robust source-seeking hybrid controllers for autonomous vehicles," in *Proceedings of the American Control Conference (ACC)*, pp. 1185–1190, 2007.

[58] C. G. Mayhew, R. G. Sanfelice, and A. R. Teel, "Robust source-seeking hybrid controllers for nonholonomic vehicles," in *Proceedings of the American Control Conference (ACC)*, pp. 2722–2727, 2008.

[59] J. Cochran and M. Krstic, "Nonholonomic source seeking with tuning of angular velocity," *IEEE Transactions on Automatic Control*, vol. 54, no. 4, pp. 717–731, 2009.

[60] J. Cochran, A. Siranosian, N. Ghods, and M. Krstic, "3-D source seeking for underactuated vehicles without position measurement," *IEEE Transactions on Robotics*, vol. 25, no. 1, pp. 117–129, 2009.

[61] C. Zhang, D. Arnold, N. Ghods, A. Siranosian, and M. Krstic, "Source seeking with non-holonomic unicycle without position measurement and with tuning of forward velocity," *Systems & Control Letters*, vol. 56, no. 3, pp. 245–252, 2007.

[62] C. Manzie and M. Krstic, "Extremum seeking with stochastic perturbations," *IEEE Transactions on Automatic Control*, vol. 54, no. 3, pp. 580–585, 2009.

[63] S. Liu and M. Krstic, "Stochastic source seeking for nonholonomic unicycle," *Automatica*, vol. 46, no. 9, pp. 1443–1453, 2010.

[64] S. J. Liu, P. Frihauf, and M. Krstic, "Stochastic source seeking with tuning of forward velocity," in *Proceedings of the 31st Chinese Control Conference (CCC)*, pp. 4424–4429, 2012.

[65] Y. Pan, K. D. Kumar, and G. Liu, "Extremum seeking control with second-order sliding mode," *SIAM Journal on Control and Optimization*, vol. 50, no. 6, pp. 3292–3309, 2012.

[66] A. S. Matveev, H. Teimoori, and A. V. Savkin, "Navigation of a unicycle-like mobile robot for environmental extremum seeking," *Automatica*, vol. 47, no. 1, pp. 85–91, 2011.

[67] A. S. Matveev, M. C. Hoy, and A. V. Savkin, "Extremum seeking navigation without derivative estimation of a mobile robot in a dynamic environmental field," *IEEE Transactions on Control Systems Technology*, vol. 24, no. 3, pp. 1084–1091, 2016.

[68] E. Bıyık and M. Arcak, "Gradient climbing in formation via extremum seeking and passivity-based coordination rules," *Asian Journal of Control*, vol. 10, no. 2, pp. 201–211, 2008.

[69] N. E. Leonard and E. Fiorelli, "Virtual leaders, artificial potentials and coordinated control of groups," in *Proceedings of the 40th Conference on Decision and Control (CDC)*, vol. 3, pp. 2968–2973, 2001.

[70] R. Bachmayer and N. E. Leonard, "Vehicle networks for gradient descent in a sampled environment," in *Proceedings of the 41st Conference on Decision and Control (CDC)*, vol. 1, pp. 112–117, 2002.

[71] P. Ögren, E. Fiorelli, and N. E. Leonard, "Cooperative control of mobile sensor networks: Adaptive gradient climbing in a distributed environment," *IEEE Transactions on Automatic Control*, vol. 49, no. 8, pp. 1292–1302, 2004.

[72] B. J. Moore and C. Canudas-de Wit, "Source seeking via collaborative measurements by a circular formation of agents," in *Proceedings of the American Control Conference (ACC)*, pp. 6417–6422, 2010.

[73] C. Smith, *The automatic computation of maximum likelihood estimates.* Scientific Department, National Coal Board, 1962.

[74] S. J. Liu and M. Krstic, "Introduction to extremum seeking," in *Stochastic Averaging and Stochastic Extremum Seeking*, pp. 11–20, Springer, 2012.

[75] Y. Tan, W. Moase, C. Manzie, D. Nešić, and I. Mareels, "Extremum seeking from 1922 to 2010," in *Proceedings of the 29th Chinese Control Conference (CCC)*, pp. 14–26, 2010.

[76] M. Arcak, "Passivity as a design tool for group coordination," *IEEE Transactions on Automatic Control*, vol. 52, no. 8, pp. 1380–1390, 2007.

[77] H. Bai, M. Arcak, and J. T. Wen, "Group coordination when the reference velocity is available only to the leader: An adaptive design," in *Proceedings of the American Control Conference (ACC)*, pp. 5400–5405, 2007.

[78] L. B. Arranz, A. Seuret, and C. C. De Wit, "Translation control of a fleet circular formation of AUVs under finite communication range," in *Proceedings*

of the 48th Conference on Decision and Control held jointly with the 28th Chinese Control Conference CDC/CCC, pp. 8345–8350, 2009.

[79] R. Carona, A. P. Aguiar, and J. Gaspar, "Control of unicycle type robots tracking, path following and point stabilization," *Proceedings of IV Jornadas de Engenharia Electrónica e Telecomunicações e de Computadores*, pp. 180–185, 2008.

[80] A. De Luca and G. Oriolo, "Modelling and control of nonholonomic mechanical systems," in *Kinematics and dynamics of multi-body systems, CISM Lecture Notes*, no. 360, pp. 277–342, Springer, 1995.

[81] M. Bibuli, G. Bruzzone, M. Caccia, and L. Lapierre, "Path-following algorithms and experiments for an unmanned surface vehicle," *Journal of Field Robotics*, vol. 26, no. 8, pp. 669–688, 2009.

[82] H. Teimoori and A. V. Savkin, "Equiangular navigation and guidance of a wheeled mobile robot based on range-only measurements," *Robotics and Autonomous Systems*, vol. 58, no. 2, pp. 203–215, 2010.

[83] C. Edwards and S. Spurgeon, *Sliding mode control: theory and applications.* Taylor & Francis, London, 1998.

[84] Y. Shtessel, C. Edwards, L. Fridman, and A. Levant, *Sliding mode control and observation.* Springer, New York, 2014.

[85] H. K. Khalil and J. Grizzle, *Nonlinear systems.* Prentice Hall, New Jersey, 1996.

[86] L. Fridman and A. Levant, "Higher order sliding modes," *Sliding Mode Control in Engineering*, vol. 11, pp. 53–102, 2002.

[87] G. Bartolini, A. Pisano, E. Punta, and E. Usai, "A survey of applications of second-order sliding mode control to mechanical systems," *International Journal of Control*, vol. 76, no. 9-10, pp. 875–892, 2003.

[88] G. Bartolini, A. Pisano, and E. Usai, "Digital second-order sliding mode control for uncertain nonlinear systems," *Automatica*, vol. 37, no. 9, pp. 1371–1377, 2001.

[89] M. Rubagotti, A. Estrada, F. Castaños, A. Ferrara, and L. Fridman, "Integral sliding mode control for nonlinear systems with matched and unmatched perturbations," *IEEE Transactions on Automatic Control*, vol. 56, no. 11, pp. 2699–2704, 2011.

[90] E. Blockley, M. Martin, A. McLaren, A. Ryan, J. Waters, D. Lea, I. Mirouze, K. Peterson, A. Sellar, and D. Storkey, "Recent development of the Met Office operational ocean forecasting system: an overview and assessment of the new global FOAM forecasts," *Geoscientific Model Development Discussions*, vol. 6, pp. 6219–6278, 2013.

[91] R. Pingree and D. Griffiths, "Tidal fronts on the shelf seas around the British Isles," *Journal of Geophysical Research: Oceans*, vol. 83, no. C9, pp. 4615–4622, 1978.

[92] J. Hopkins, P. Challenor, and A. G. Shaw, "A new statistical modeling approach to ocean front detection from SST satellite images," *Journal of Atmospheric and Oceanic Technology*, vol. 27, no. 1, pp. 173–191, 2010.

[93] R. E. Davis, C. C. Eriksen, and C. P. Jones, "Autonomous buoyancy-driven underwater gliders," *The technology and applications of autonomous underwater vehicles*, pp. 37–58, 2002.

[94] O. Schofield, J. Kohut, D. Aragon, L. Creed, J. Graver, C. Haldeman, J. Kerfoot, H. Roarty, C. Jones, D. Webb, *et al.*, "Slocum gliders: Robust and ready," *Journal of Field Robotics*, vol. 24, no. 6, pp. 473–485, 2007.

[95] G. Griffiths, P. Stevenson, A. T. Webb, N. W. Millard, S. D. McPhail, M. Pebody, and J. R. Perrett, "Open ocean operational experience with the Autosub-1 autonomous underwater vehicle," in *International Symposium on Unmanned Untethered Submersible Technology*, pp. 1–12, 1999.

[96] D. L. Rudnick, R. E. Davis, C. C. Eriksen, D. M. Fratantoni, and M. J. Perry, "Underwater gliders for ocean research," *Marine Technology Society Journal*, vol. 38, no. 2, pp. 73–84, 2004.

[97] C. Jones, E. Creed, S. Glenn, J. Kerfoot, J. Kohut, C. Mudgal, and O. Schofield, "Slocum gliders - a component of operational oceanography," in *Proceedings of the 14th International Symposium on Unmanned Untethered Submersible Technology*, 2005.

[98] C. Jones, D. Webb, S. Glenn, O. Schofield, J. Kerfoot, J. Kohut, D. Aragon, C. Haldeman, T. Haskin, A. Kahl, *et al.*, "Slocum glider expanding the capabilities," *17th InternationalSymposium on Unmanned Untethered Submersible Technology 2011 (UUST 2011)*, 2011.

[99] R. N. Smith, P. Cooksey, F. Py, G. S. Sukhatme, and K. Rajan, "Adaptive path planning for tracking ocean fronts with an autonomous underwater vehicle," in *Experimental Robotics*, pp. 761–775, Springer, 2016.

[100] E. Fiorelli, N. E. Leonard, P. Bhatta, D. A. Paley, R. Bachmayer, and D. M. Fratantoni, "Multi-AUV control and adaptive sampling in Monterey Bay," *IEEE Journal of Oceanic Engineering*, vol. 31, no. 4, pp. 935–948, 2006.

[101] R. N. Smith, Y. Chao, P. P. Li, D. A. Caron, B. H. Jones, and G. S. Sukhatme, "Planning and implementing trajectories for autonomous underwater vehicles to track evolving ocean processes based on predictions from a regional ocean model," *International Journal of Robotics Research*, vol. 29, no. 12, pp. 1475–1497, 2010.

[102] A. Stuntz, J. S. Kelly, and R. N. Smith, "Enabling persistent autonomy for underwater gliders with ocean model predictions and terrain-based navigation," *Frontiers in Robotics and AI*, vol. 3, pp. 1–17, 2016.

[103] H. C. Woithe, D. Boehm, and U. Kremer, "Improving Slocum glider dead reckoning using a doppler velocity log," in *OCEANS 2011*, pp. 1–5, 2011.

[104] J. G. Graver, *Underwater gliders: Dynamics, control and design.* PhD thesis, Princeton University New Jersey, USA, 2005.

[105] B. L. Stevence and F. L. Lewis, *Aircraft control and simulation.* John Wiley and Sons, New York, 1992.

[106] P. Jantapremjit and P. Wilson, “Guidance-control based path following for homing and docking using an autonomous underwater vehicle,” in *Proceedings of the IEEE/MST Oceans Conference and Exhibition Conference and Exhibition*, pp. 1–6, 2008.

[107] I. Abraham and J. Yi, “Model predictive control of buoyancy propelled autonomous underwater glider,” in *Proceedings of the American Control Conference (ACC)*, pp. 1181–1186, 2015.

[108] N. Mahmoudian, *Efficient motion planning and control for underwater gliders.* PhD thesis, Virginia Polytechnic Institute, 2009.

[109] N. Mahmoudian, J. Geisbert, and C. Woolsey, “Approximate analytical turning conditions for underwater gliders: Implications for motion control and path planning,” *IEEE Journal of Oceanic Engineering*, vol. 35, no. 1, pp. 131–143, 2010.

[110] D. Kowalyshyn, “Buoyancy driven underwater glider,” Master’s thesis, Swarthmore College, 2014.

[111] N. Mahmoudian and C. Woolsey, “Underwater glider motion control,” in *Proceedings of the 47th Conference on Decision and Control (CDC)*, pp. 552–557, 2008.

[112] R. Bachmayer, J. G. Graver, and N. E. Leonard, “Glider control: a close look into the current glider controller structure and future developments,” in *OCEANS*, vol. 2, pp. 951–954, 2003.

[113] P. Bhatta, *Nonlinear stability and control of gliding vehicles.* PhD thesis, Princeton University, New Jersey, 2004.

[114] A. De Marco, E. L. Duke, and J. S. Berndt, “A general solution to the aircraft trim problem,” in *AIAA Modeling and Simulation Technologies Conference and Exhibit*, pp. 6703–6743, 2007.

[115] F. Zhang, *Modeling, design and control of gliding robotic fish.* PhD thesis, Michigan State University, 2014.

[116] Mathworks, “Fmincon r2012a documentation.”

[117] N. E. Leonard and J. G. Graver, “Model-based feedback control of autonomous underwater gliders,” *IEEE Journal of Oceanic Engineering*, vol. 26, no. 4, pp. 633–645, 2001.

[118] A. Marzocchi, J. J.-M. Hirschi, N. P. Holliday, S. A. Cunningham, A. T. Blaker, and A. C. Coward, “The North Atlantic subpolar circulation in an eddy-resolving global ocean model,” *Journal of Marine Systems*, vol. 142, pp. 126–143, 2015.

[119] G. Madec and M. Imbard, “A global ocean mesh to overcome the North Pole singularity,” *Climate Dynamics*, vol. 12, no. 6, pp. 381–388, 1996.

[120] T. Rossby and C. Flagg, “Direct measurement of volume flux in the Faroe-Shetland Channel and over the Iceland-Faroe Ridge,” *Geophysical Research Letters*, vol. 39, no. 7, 2012.

[121] H. Yu and U. Ozguner, “Extremum-seeking control via sliding mode with periodic search signals,” in *Proceedings of the 41st Conference on Decision and Control (CDC)*, vol. 1, pp. 323–328, 2002.

[122] A. S. Matveev, A. A. Semakova, and A. V. Savkin, “Technical facts about dynamic scalar fields underlying algorithms of mobile robots navigation for tracking environmental boundaries and extremum seeking,” *arXiv preprint arXiv:1608.04553*, 2016.

[123] S. Wright and J. Nocedal, “Numerical optimization,” *Springer Science*, vol. 35, no. 67-68, p. 7, 1999.

[124] A. F. Shchepetkin and J. C. McWilliams, "The Regional Oceanic Modeling System (ROMS): a split-explicit, free-surface, topography-following-coordinate oceanic model," *Ocean Modelling*, vol. 9, no. 4, pp. 347–404, 2005.

[125] R. N. Smith, J. Das, Y. Chao, D. A. Caron, B. H. Jones, and G. S. Sukhatme, "Cooperative multi-AUV tracking of phytoplankton blooms based on ocean model predictions," in *Proceedings of the IEEE/MST Oceans Conference and Exhibition*, pp. 1–10, 2010.

[126] D. J. MacKay, "Introduction to Gaussian Processes," *NATO ASI Series F Computer and Systems Sciences*, vol. 168, pp. 133–166, 1998.

[127] P. Boyle, *Gaussian Processes for regression and optimisation.* PhD thesis, Victoria University of Wellington, 2007.

[128] C. E. Rasmussen, *Gaussian Processes for machine learning.* MIT Press, London, 2006.

[129] M. L. Stein, *Interpolation of spatial data: some theory for kriging.* Springer Science & Business Media, 2012.

[130] P. J. Diggle and P. J. Ribeiro, "Bayesian inference in Gaussian model-based geostatistics," *Geographical and Environmental Modelling*, vol. 6, no. 2, pp. 129–146, 2002.

[131] R. D. C. Team, *R: A Language and Environment for Statistical Computing.* R Foundation for Statistical Computing, Vienna, Austria, 2008.

[132] J. A. Nelder and R. Mead, "A simplex method for function minimization," *The computer journal*, vol. 7, no. 4, pp. 308–313, 1965.

[133] J. C. Lagarias, J. A. Reeds, M. H. Wright, and P. E. Wright, "Convergence properties of the Nelder–Mead simplex method in low dimensions," *SIAM Journal on optimization*, vol. 9, no. 1, pp. 112–147, 1998.

[134] P. J. Diggle, J. Tawn, and R. Moyeed, "Model-based geostatistics," *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, vol. 47, no. 3, pp. 299–350, 1998.

[135] M. W. Davis and C. Grivet, "Kriging in a global neighborhood," *Mathematical Geology*, vol. 16, no. 3, pp. 249–265, 1984.

[136] F. A. Gema and J. Mateu, *Spatial and spatio-temporal geostatistical modeling and kriging*. John Wiley & Sons, Chichester, 2015.

[137] R. J. Larsen and M. L. Marx, *An introduction to mathematical statistics and its applications*, vol. 2. Prentice-Hall Englewood Cliffs, NJ, 1986.

[138] L. S. Bastos and A. O'Hagan, "Diagnostics for Gaussian Process emulators," *Technometrics*, vol. 51, no. 4, pp. 425–438, 2009.

[139] P. Ögren, E. Fiorelli, and N. E. Leonard, "Cooperative control of mobile sensor networks: Adaptive gradient climbing in a distributed environment," *IEEE Transactions on Automatic Control,*, vol. 49, no. 8, pp. 1292–1302, 2004.

[140] J. Marshall, M. E. Broucke, and B. Francis, "Formations of vehicles in cyclic pursuit," *IEEE Transactions on Automatic Control*, vol. 49, no. 11, pp. 1963–1974, 2004.

[141] J. M. Soares, A. P. Aguiar, A. M. Pascoal, and M. Gallieri, "Triangular formation control using range measurements: An application to marine robotic vehicles," *IFAC Proceedings*, vol. 45, no. 5, pp. 112–117, 2012.

[142] S. Daingade, A. Sinha, A. V. Borkar, and H. Arya, "A variant of cyclic pursuit for target tracking applications: theory and implementation," *Autonomous Robots*, pp. 1–18, 2015.

[143] A. Jain, D. Ghose, and P. P. Menon, "Achieving a desired collective centroid by a formation of agents moving in a controllable force field," in *Proceerings of the Indian Control Conference (ICC)*, pp. 182–187, 2016.

[144] R. Sepulchre, D. Paley, and N. E. Leonard, "Stabilization of planar collective motion: All-to-all communication," *IEEE Transactions on Automatic Control*, vol. 52, no. 5, pp. 811–824, 2007.

[145] A. Levant, "Quasi-continuous high-order sliding-mode controllers," *IEEE Transactions on Automatic Control*, vol. 50, no. 11, pp. 1812–1816, 2005.

[146] J. Waters, D. Lea, M. Martin, D. Storkey, and J. While, "Describing the development of the new FOAM-Nemovar system in the global 1/4 degree configuration," *Technical Report, Met Office*, 2013.

[147] D. L. Zimmerman, "Another look at anisotropy in geostatistics," *Mathematical Geology*, vol. 25, no. 4, pp. 453–470, 1993.

[148] P. J. Ribeiro Jr, O. F. Christensen, and P. J. Diggle, "geoR and geoRglm: software for model-based geostatistics," in *Proceedings of DSC*, vol. 2, 2003.

[149] P. J. Ribeiro Jr and P. J. Diggle, "geoR: A package for geostatistical analysis," *R news*, vol. 1, no. 2, pp. 14–18, 2001.

[150] T. Chung, L. Cremean, W. B. Dunbar, Z. Jin, E. Klavins, D. Moore, A. Tiwari, D. Van Gogh, and S. Waydo, "A platform for cooperative and coordinated control of multiple vehicles," in *Recent Developments in Cooperative Control and Optimization*, pp. 75–104, Springer, 2004.

[151] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, "ROS: an open-source Robot Operating System," in *ICRA workshop on open source software*, vol. 3, p. 5, Kobe, Japan, 2009.

[152] E. White, *Making Embedded Systems: Design Patterns for Great Software.* O'Reilly Media, Inc., 2011.

[153] J. M. O'Kane, "A gentle introduction to ROS," *University of South Carolina*, 2014.

[154] V. I. Utkin, *Sliding modes in control and optimization.* Springer Science & Business Media, 2013.

[155] U. Itkis, *Control systems of variable structure.* Wiley, New York, 1976.

[156] V. Utkin, "Variable structure systems with sliding modes," *IEEE Transactions on Automatic control*, vol. 22, no. 2, pp. 212–222, 1977.

[157] G. Bartolini, A. Ferrara, and E. Usai, "On multi-input second order sliding mode control of nonlinear systems with uncertainty," in *Proceedings of the 38th Conference on Decision and Control (CDC)*, pp. 4245–4250, 1999.

[158] A. Levant, "Chattering analysis," *IEEE Transactions on Automatic Control*, vol. 55, no. 6, pp. 1380–1389, 2010.

[159] H. Lee and V. I. Utkin, "Chattering suppression methods in sliding mode control systems," *Annual Reviews in Control*, vol. 31, no. 2, pp. 179–188, 2007.

[160] A. Isidori, *Nonlinear control systems.* Springer Science & Business Media, London, 2013.

[161] S. Spurgeon, "Sliding mode control: a tutorial," in *Proceeding of the European Control Conference (ECC)*, pp. 2272–2277, 2014.

[162] L. Levantovsky, "Second order sliding algorithms: their realization," *Dynamics of Heterogeneous Systems*, pp. 32–43, 1985.

[163] A. Levant, "Sliding order and sliding accuracy in sliding mode control," *International Journal of Control*, vol. 58, no. 6, pp. 1247–1263, 1993.

[164] S.-S. Fan, C.-J. Yang, S.-L. Peng, K.-H. Li, Y. Xie, and S.-Y. Zhang, "Underwater glider design based on dynamic model analysis and prototype development," *Journal of Zhejiang University: Science C*, vol. 14, no. 8, pp. 583–599, 2013.