D 2017

U.PORTO
FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

NAVIGATION ALGORITHMS FOR SENSOR-LIMITED AUTONOMOUS UNDERWATER VEHICLES

JOSÉ LUÍS DA ROCHA MELO

TESE DE DOUTORAMENTO APRESENTADA
À FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO EM
ENGENHARIA ELECTROTÉCNICA E DE COMPUTADORES

U.PORTO

FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

Navigation Algorithms for Sensor-limited Autonomous Underwater Vehicles

José Luís da Rocha Melo

Supervisor: Aníbal Castilho Coimbra de Matos

Doctoral Program in Electrical and Computer Engineering

January 2017

Faculdade de Engenharia da Universidade do Porto

Navigation Algorithms for Sensor-limited Autonomous Underwater Vehicles

José Luís da Rocha Melo

Dissertation submitted to Faculdade de Engenharia da Universidade do Porto
to obtain the degree of

Doctor Philosophiae in Electrical & Computer Engineering

January 2017

To my parents.

Abstract

Navigation is one of the key elements in modern robotics, as the ability that autonomous vehicles must have to correctly understand their position and attitude within the environment is determinant for the success of the different applications. This thesis will address different topics related with Underwater Navigation, always with the goal of extending current levels of navigation autonomy of sensor-limited AUVs.

Sensor-limited AUVs refers to systems equipped with only a limited set of sensors, usually a combination of low-accuracy inertial sensors, together with depth sensor and compass, but also low information sonar sensors, like an altimeter. For these type of vehicles, very common nowadays, the problem of navigation is even more challenging. This thesis is divided in two different parts, corresponding to the two general and very up to date research topics that have been addressed. The first part addresses the topic of long term and long range navigation of sensor-limited AUVs. On the other hand, the second part is devoted navigation in operations with multiple vehicles. Common to all the work here presented is the use of Bayesian Estimation techniques as a tool to address the problems under consideration.

Regarding the first topic, information of local natural features of the environment can be extracted and used to aid the navigation of sensor-limited AUVs. By doing so, a completely on-board and autonomous navigation of AUVs can be achieved, without the need for external aiding devices or support vessels. Two aspects were considered. First, the development of a sensor-based guidance strategy for Bottom Following was addressed. Bottom Following, described as the manoeuvre that enables AUVs to follow a trajectory similar to the topography of the bottom, can be very interesting for a number of applications related with inspection of the sea bottom. This Bottom Following strategy was later complemented by the development of suitable a vertical controller for the MARES AUV, using Eigenstructure Assignment. The second aspect is related to Terrain Based Navigation, and how information of the terrain can be used to bound the navigation errors that otherwise could arise. A comparative analysis of the different aspects that can have an influence in the Particle Filter based solutions for this problem. Additionally a new Data-Driven Particle Filter was proposed that is more precise and efficient when compared to existing algorithms.

Regarding the second topic addressed, of enabling operations with multiple vehicles, the focus is on developing an Acoustic Navigation strategy that allows the simultaneous navigation and tracking of multiple vehicles. The motivation for this is the need for extending existing Acoustic Navigation schemes for scenarios with multiple vehicles, while maintaining the same level of functionality. The Multiple AUV Tracker, an algorithm that is able to track multiple vehicles, even if they are emitting otherwise undistinguishable acoustic signals was developed. Building up on this tracker, a LBL acoustic network is proposed that enables the navigation of multiple vehicles. At the core of this network is the aforementioned tracker, but also the use of synchronous-clock

One-Way-Travel-Time techniques to provide pseudo-range measurements to the vehicles.

Keywords: AUV Navigation; Bottom Following; Terrain Based Navigation; Tracking of Multiple AUVs; LBL Acoustic Positioning;

Resumo

A navegação é um elemento chave na robótica moderna, já que qualquer veículo autónomo necessita de consigar estimar corretamente a sua posição e atitude relativamente ao meio em que se encontra. Só assim será possível que concluam com sucesso as missões a que estão destinados. Esta tese aborda diferences tópicos, mas todos eles visando aumentar os atuais níveis de autonomia de Veículos Autónomos Subaquáticos com limitações sensoriais.

Veículos Autónomos Subaquáticos com limitações sensoriais é a designação utilizada quando se refere a uma classe de veículos equipada com um conjunto limitado de sensores, normalmente uma combinação de sensores inerciais de baixa precisão, sensor de profundidade e bússola, mas também sonares de baixa resolução. Para este tipo de veículos, hoje em dia muito comum, o problema da navegação é um problema ainda mais desafiante. Os algoritmos aqui apresentados têm como motivação dar resposta a duas questões pertinentes no campo da navegação subaquática. Por um lado, procura-se desenvolver os algoritmos que permitam a navegação deste tipo de veículos por períodos e/ou distâncias mais alargadas. Por outro lado, procura-se permitir a navegação e seguimento simultâneo de múltiplos veículos, usando para isso redes acústicas. Transversal a estes dois temas é o uso de técnicas de Estimação Bayesiana para abordar os problemas em consideração.

Em relação ao primeiro tema enunciado, é comum usar características naturais locais, por forma a aumentar a autonomia de navegação do veículo. Em concreto, informação sobre a batimetria do terreno pode ser utilizada para permitir uma navegação mais autónoma. Quando tal acontece, os tradicionais métodos de navegação e localização acústica podem até ser dispensados, aumentando desde logo o grau de autonomia dos veículos. Informação sobre o terreno pode ser utilizada, por exemplo, para permitir o Seguimento de Fundos. Será apresentado uma estratégia de *guidance* baseada em informação batimétrica obtida por um altímetro, permitindo que os veículos sigam sigam trajetórias que se assemelhem à topografia do fundo. Esta manobra é particularmente apreciada em missões relacionadas com a inspeção de fundos. Esta estratégia é complementada pelo desenvolvimento de um controlador vertical para o Veículo Autónomo Subaquático MARES, usando técnicas de *Eigenstructure Assignment*. O segundo aspeto relacionado com o uso de características naturais, é o desenvolvimento de algoritmos eficientes de navegação baseada em terrenos. Será feita uma análise comparativa sobre o efeito que os diversos parâmetros dos Filtros de Partículas podem ter em algoritmos deste género. Adicionalmente, será proposto um Filtro de Partículas *data-driven*, que é mais preciso e eficiente que algoritmos já existentes.

Em relação ao segundo tema da tese, relativo a operações com múltiplos veículos, é apresentada uma estratégia de Navegação Acústica que permite, simultaneamente, a navegação e o seguimento externo de múltiplos veículos. A motivação para este estratégia advém da necessidade de alagar as técnicas de Navegação Acústica já existentes para casos em que existem múltiplos veículos, mas mantendo o mesmo nível de funcionalidades já existente. Inicialmente é apresentado um algoritmo capaz de fazer o seguimento de vários veículos, a emitir sinais acústicos semelhantes. Para além disso, é proposto uma nova rede de posicionamento acústico LBL que permite, simultaneamente, a navegação e seguimento de múltiplos veículos, usando apenas sinais acústicos semelhantes. Na base de funcionamento deste rede está, para além do algoritmo de seguimento já referido, o uso de relógios sincronizados. Dessa forma, cada um dos veículos poderá obter medidas de distância, usando o chamado *One-Way Travel Time* dos sinais acústicos utilizados.

Palavras-chave: Navegação de AUVs; Seguimento de fundos; Navegação Baseada em Terrenos; AUV Navigation; Bottom Following; Terrain Based Navigation; Seguimento de múltiplos veículos; Posicionamento acústico por LBL;

Acknowledgments

Going through all the pages of my thesis, I feel a sense of pride for what I have accomplished. If I could go back in time, I certainly could have made some things differently. But, as someone once told me, my work was earnest.

It was already six years ago since I made the decision of returning to my Porto and to FEUP to start my PhD. And what a long journey this has been, undoubtedly longer than what I ever expected. Looking back, I am truly grateful for all the things I learnt, all the people I have met and all the places I have been to. Nevertheless, this was a bumpy journey, where at times it was difficult to find the strength and the motivation to keep going. This section is for all of those who kept me going during all this time.

First and foremost I need to thank to my supervisor, Professor Aníbal Matos. He was the one who first invited me to return to Porto and collaborate with him when my plans were pointing in different directions. And as if that is not enough, I am also grateful for the motivation and guidance that I needed throughout all these years. I never cease to be amazed by his apparently never ending knowledge about pretty much everything.

I also would like to thank all my colleagues that accompanied me on all these years. First of all to the ones working in our lab, that have to deal with me daily. I would literally not be able to do as much without you guys! I also need to thank all the other people from CROB, that daily have to put up with me, always helping with that extra coin when I just don't have enough for that much needed coffee!

On a more institutional note, I also need to acknowledge FCT for the scholarship that supported most of my work, and also INESC-TEC, for accepting me as a researcher and for the continuous support throughout this journey.

Finally, I am undoubtedly thankful for the continuous support of the ones who are closer to me, not only during this journey but more importantly in life. All my friends, the ones I have known since ever and the ones who only recently I came across, the ones who are in town and those who live further. I would have never made it without you.

And of course, my family. Always there, ready to support, help and comfort me unconditionally and in anyway they can, even if sometimes they don't quite understand my choices. I love you.

José Luís da Rocha Melo

Contents

List of Figures

List of Tables

List of Abbreviations

ASV	Autonomous Surface Vehicle
AUV	Autonomous Underwater Vehicle
DD	Data-Driven
DTM	Digital Terrain Model
DVL	Doppler Velocity Log
EA	Eigenstructure Assignment
GNSS	Global Navigation Satellite System
GPS	Global Positioning System
IMU	Inertial Measurement Unit
INS	Inertial Navigation System
LBL	Long Baseline
KF	Kalman Filter
MSL	Mean Sea Level
PF	Particle Filter
OWTT	One-Way Travel Time
PHD	Probability Hypothesis Density
RFS	Random Finite Set
SIR	Sequential Importance Resampling
SMC	Sequential Monte-Carlo
TBN	Terrain Based Navigation
ToF	Time-of-Flight
TWTT	Two-Way Travel Time
UUV	Unmanned Underwater Vehicle

Part I

Introduction

Chapter 1

Introduction

Navigation, is a broad and extensive field of study, and one of the key issues in modern robotics. In the early days of mobile robotics Navigation has been defined as the process of measurement and computation necessary to determine the present and probable future positions of a vehicle (Draper et al., 1965). On a more contemporary definition, nowadays Navigation encompasses all the sub-problems related to moving a robot from one point to the other, thus including different tasks like sensing, state estimation, perception, robot self localization and collision avoidance, mapping or even trajectory planning and situational awareness (Kendoul, 2012). Notwithstanding, for the Marine Robotics community the term Navigation is loosely applied to the self-localization task only, and that is the understanding that will be followed throughout the remainder of this thesis.

Autonomous Underwater Vehicles (AUVs) are becoming a reliable and cost-effective solution for performing a variety of underwater tasks in a fully automated way. Minimization of the risk of operations in dangerous scenarios, increasing the reliability of the operations, and accomplishing them more quickly are, generally speaking, what is sought when relying on autonomous vehicles. Among the main tasks to be performed by AUVs are bathymetric tasks, environmental surveying, surveillance and patrolling, or even mine countermeasures operations. The use of such vehicles means not only that the assigned tasks can be performed in a more cost-effective way, but also enables operations in challenging scenarios, in a way that would not be safe, or even possible, for human intervention.

The level of autonomy achieved by AUVs is chiefly determined by their performance in three areas, namely energy autonomy, navigation autonomy and decision autonomy (Hagen et al., 2009). Navigation autonomy can then be described as the ability to navigate precisely and positioning with little or non-significant error over extended periods of time. The ability autonomous vehicles must have to correctly understand their position and attitude within the environment is determinant for the success of different applications. For field robotics, the Global Positioning System (GPS) has revealed to be an invaluable asset to resolve the global position of a given robot. In fact, modern GPS techniques can provide the global position in terms of latitude and longitude with centimetre accuracy, which is considered to be adequate to the vast majority of applications. However in GPS denied environments there are no general obvious solutions to the navigation problem.

The use of GPS technology is not possible underwater, as electromagnetic radiations are strongly attenuated in aquatic environments. As such, alternative techniques must be derived to provide accurate navigation capabilities to underwater vehicles. This thesis will address different topics concerning extending current levels of navigation autonomy of AUVs. While the developed methods are appropriated to be used by other kinds of Marine Vehicles, like Remotely Operated Underwater Vehicles (ROVs) for example, the focus will be on AUVs which pose more difficult and challenging problems.

1.1 Motivation

Autonomous long range and long term operations, without the need for human intervention, is still a challenging goal for robotic underwater vehicles. The ability to perform long-term and long-range missions would dramatically influences the level of autonomy of these vehicles. However, so far this kind of autonomy is only accessible to high-end autonomous vehicles, that can navigate for long periods relying only on their own high-precision sensors and equipments.

A diversity of Autonomous Underwater Vehicles have already been developed, in both research or industrial contexts. Naturally, existing AUVs have very different characteristics, from their physical attributes to the sensors that they carry. Such differences are related to the envisioned applications and operational environments for each of the vehicles. In the context of this thesis, the algorithms that will be presented refer to sensor-limited AUVs.

Sensor-limited AUVs have been referred in the literature as autonomous vehicles equipped with only a limited set of sensors characterized by their low accuracy. These kind of vehicles usually employ a combination of low-accuracy inertial sensors, together with depth sensor and compass, but also low information range sensors, like an altimeter or a Doppler Velocity Log (DVL)(Meduna, 2011). In some cases the authors also consider the use of underwater modems, allowing them to acoustically communicate with other vehicles in the team (Teck, 2014). Sensor-limited AUVs are also often vehicles that have smaller dimensions and can withstand lower depths. Available on-board computing power also has some limitations. Naturally, the biggest advantage of sensor-limited AUVs comes in terms of their cost, which is significantly lower when compared to high-end vehicles.

Opposed to this, other class of vehicles that can be considered is the class of sensor-rich vehicles. In this class, vehicles are equipped with multiple state-of-the-art navigation sensors that allow the vehicles to successfully perform long range missions. Examples of such sensors are Ring Laser Gyroscopes and Vibrating Beam Accelerometers, together with high precision clock sources. Vehicles using this kind of navigation sensors can perform long missions without requiring direct position measurements updates from external aiding sensors, that bound the navigation errors. Moreover, such class of vehicles are also equipped with high-resolution sonar sensors that provide unrivalled perception of the surrounding environment.

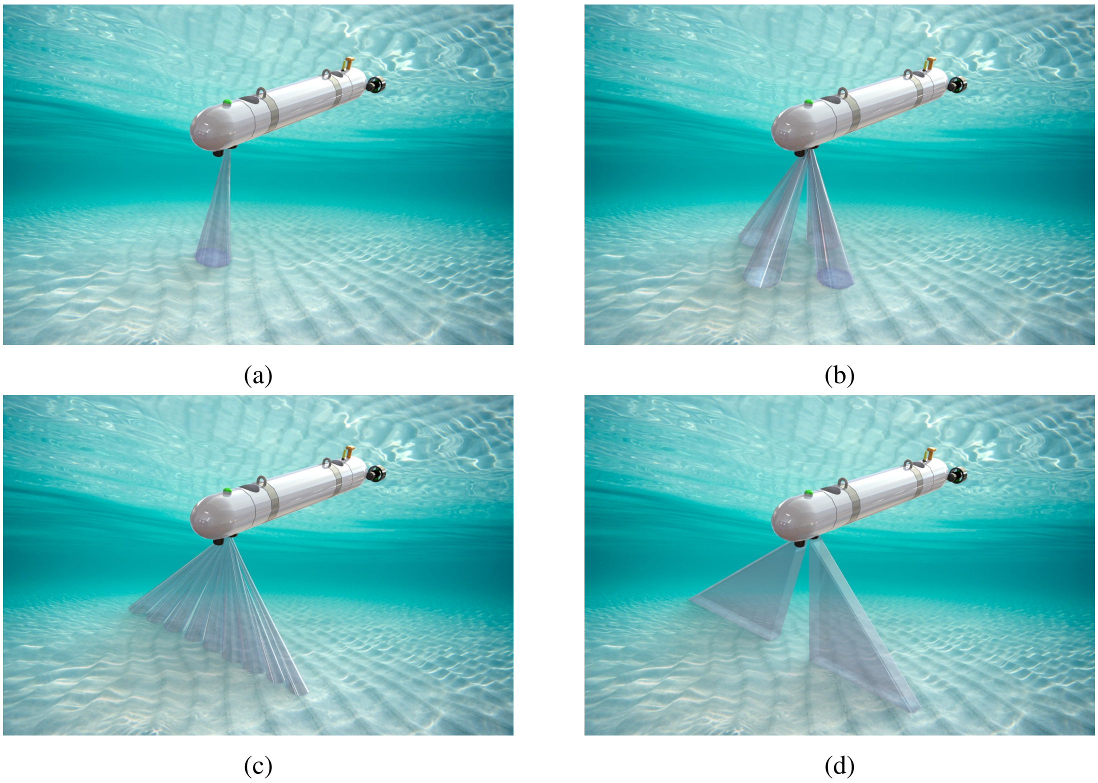


Figure 1.1: Footprint comparison for the different sonar sensors: altimeter (1.1a), DVL (1.1b), Multibeam (1.1c), and Sidescan Sonar (1.1d)

Figure 1.1 presents a representation of different sonar sensors available, where it is possible to compare the differences of the sensors footprint. As it can be seen, Altimeters and DVLs have rather smaller footprints, thus providing less information about the terrain. By comparison, Multibeams and Sidescan Sonars have larger footprints, providing a swath of readings in a single ping. Sensor-limited systems often implies the use of the former kind of sensors, and that will be the understanding throughout this thesis.

The current state-of-the-art for the navigation of sensor-limited AUVs consists on the combination of Inertial Navigation techniques with Acoustic Navigation systems, that can provide external position measurements. Inertial Navigation left alone is known to drift over time, and such drifts are particularly visible in sensor-limited vehicles. On the other hand, Acoustic Navigation always requires the use of external acoustic beacons, that need to be deployed and naturally confine the area of operation of the vehicles, thus limiting their autonomy in terms of range and duration of the missions. Therefore, methods are needed that enable long-term and long-range operations for sensor-limited systems.

At the same time, and following current trends in robotics, there has been an increasing interest on enabling operations with multiple vehicles. As such, there are already significant research efforts directed towards the development of algorithms for the control and coordination of multiple marine vehicles, for the completion of a common goal. Even though there is an extensive and growing literature on cooperative control theory of fleets of vehicles, only a few examples of operations with multiple AUVs demonstrated in water have emerged on the literature. Nevertheless, with the increase of such multi-vehicle cooperating missions of underwater vehicles, the problem of navigation and localization of multiple vehicle becomes even more relevant. However, current state-of-the-art navigation algorithms, that combine both Inertial and Acoustic Navigation techniques still face some challenges when trying to cope with navigation and tracking of multiple vehicles.

The algorithms presented in this thesis try to contribute to this two general research topics in the field of underwater navigation: on one hand, providing alternatives to existing techniques that allow for long term and long range missions and, on the other hand, develop the necessary algorithms that enable operations with multiple vehicles using existing Acoustic Navigation techniques.

1.2 Thesis outline

The topics addressed in this thesis are roughly related with two different aspects of Navigation for sensor-limited Autonomous Underwater Vehicles: enabling long term and long range navigation on one hand, and navigation of multiple vehicles on the other. These topics, which were just outlined in the previous section, are collected in nine chapters and grouped in the four different parts that compose this thesis. Despite that, each chapter is self-contained, and can be read independently, without losing context. The organization of this thesis reflects a logical presentation of the different algorithms but also, almost to its full extent, the chronological order on which all the different topics were studied.

The first part, in which the current chapter is included, is an introductory part, and is composed of two different chapters. The current chapter presents a brief introduction and motivation to the work presented ahead, together with a list of the publications that directly derived from the work presented throughout this thesis. After that, Chapter 2 provides some background and literature review on the different techniques for underwater navigation. Additionally, the roots of Bayesian Filtering will also be stated. Nowadays a fundamental tool for navigation applications, various realizations of the Bayes Filters will be used in the different chapters that follow.

The second part of this thesis is devoted to the study of navigation methods that enable long-term long-range missions. Such methods normally resort to using natural features of the environment in order to improve the performance and autonomy of the vehicles. This second part of the thesis is composed by three chapters. Chapter 3 presents a Bottom Following and Estimation strategy for sensor-limited systems. The goal in this chapter is to derive an algorithm that would allow an AUV to follow a trajectory that closely resembles the profile of the bottom. Motivated by this, on Chapter 4 a dedicated pitch-depth MIMO controller, suitable for Bottom Following missions, is derived. The purpose of this controller is to be combined with the Bottom-Following approach of the previous chapter, enabling smooth trajectories for the AUV. Finally, Chapter 5 the last one of this part, deals with the problem of Terrain Based Navigation for sensor-limited AUVs. Besides providing a detailed overview on the existing state-of-the-art approaches using Particle

Filters, this chapter also formulates and studies the performance of a new Data-Driven Particle Filter for Terrain Based Navigation.

The third part of this thesis is devoted to the development of acoustic navigation techniques suitable for dealing with operations with multiple vehicles, and is composed by two distinct chapters. Chapter 6 presents a multiple vehicle tracker, that is able to acoustically track multiple AUVs. For many applications that require the use of AUVs, the ability to track in real time the trajectory of the vehicles is of uttermost importance. The Multiple AUV Tracker, presented in this chapter, is able to address the aforementioned problem. Building up on this tracking algorithm, Chapter 7 proposes a new Acoustic Navigation Long Baseline scheme that is able to cope with the simultaneous navigation and external tracking of multiple vehicles. The proposed approach is able to expand all the features of traditional LBL networks to scenarios involving multiple vehicles scenarios.

Finally, the fourth and last part of the thesis consists of Chapter 8, which concludes the thesis, presenting a summary of the achieved contributions and some reflections and suggested directions for future research work.

1.3 Original Contributions

As a result of this thesis, different original contributions have resulted in several publications, presented in both national and international conferences. Some of the contributions have also been submitted to journals of interest. In what follows, a chronological list of the articles submitted and accepted or in the process of review will be listed.

- *A bottom-following problem approach using an altimeter*, José Melo and Aníbal Matos, Proceedings of the 12th International Conference on Autonomous Robot Rystems and Competitions, Braga, Portugal, 2012

  - This article reflects the preliminary work to the Bottom Estimation problem. This work is also one of the outcomes of a project work for Mobile Robotics course, one of the elective courses attended that were part of the Doctoral Program.

- *A bottom-following problem approach using an altimeter*, José Melo and Aníbal Matos, Revista Técnico-Científica robótica.pt, nº 91, April 2013.

  - The conference article with the same name, mentioned above, was selected to be published in this Portuguese journal.

- *Bottom estimation and following with the MARES AUV*, José Melo, Aníbal Matos, Proceedings of the MTS-IEEE Conference Oceans'12, Virginia Beach, USA, October 2012.

  - This article is an extension of the articles mentioned above, adding the feature of estimation of the slope of the bottom. Additionally, strong experimental results are provided. Chapter 3 is based on this article.

- *On the use of Particle Filters for Terrain Based Navigation of sensor-limited AUVs*, José Melo, Aníbal Matos, Proceedings of the MTS-IEEE Conference Oceans'13, Bergen, Norway, June 2013.

  - This article explores and studies state-of-the-art Particle Filter algorithms for Terrain Based Navigation of sensor-limited AUVs. Chapter 5 is based on this article.

- *A PHD Filter for Tracking Multiple AUVs*, José Melo, Aníbal Matos, Proceedings of the MTS-IEEE Conference Oceans'14, St. John's, Canada, September 2014.

  - This article is a preliminary work that discusses the feasibility of an acoustic tracker for multiple AUVs, based on a Sequential Monte Carlo Probability Hypothesis Density Filter. Parts of this article are included on Chapter 6.

- *Survey on Advances on Terrain Based Navigation for Autonomous Underwater Vehicles* José Melo, Aníbal Matos, Submitted to Ocean Engineering, Elsevier, September 2015

  - This article constitutes a thorough review on Terrain Bases Navigation for Autonomous Underwater Vehicles. It resulted from the literature review on this topic.
  - Currently under revision

- *Towards LBL Positioning Systems for Multiple Vehicles*, José Melo, Aníbal Matos, Proceedings of the MTS-IEEE Conference Oceans'16, Shanghai, China, April 2016.

  - This article proposes a new LBL acoustic network that enables simultaneous localization and tracking of multiple vehicles. Chapter 7 is loosely based on this article.
  - Was submitted and accepted to the Student's Poster Competition of the IEEE/MTS Ocean's 16 Conference in Shanghai, China. Was awarded a 2^{nd} prize on the competition.

- *Tracking Multiple AUVs* José Melo, Aníbal Matos, Submitted to Autonomous Robots, May 2016

  - This article is an extension of previous preliminary work on tracking multiple AUVs. In this article different features of the tracker are introduced, and robust experimental validation results are presented. The final part of Chapter 6 is based on this article.
  - Currently under revision.

- *Data-Driven Terrain Based Navigation of AUVs* José Melo, Aníbal Matos, Submitted to Journal of Navigation, September 2017
  - This article formulates the Data-Driven Particle Filter, and studies its performance is studied when applied to the problem of Terrain Based Navigation of AUVs. The final part of Chapter 5 is based on this article.
  - Currently under revision.

Chapter 2

Underwater Navigation

2.1 Introduction

The problem of localization is one of the most fundamental tasks for the navigation of mobile robotics. For outdoor ground-based applications, the problem of navigation for autonomous vehicles is mostly solved with the use of Global Positioning System (GPS) based techniques. Modern GPS receivers can provide the global position of a vehicle in terms of latitude and longitude with centimetre level precision, but also its velocity, which is found to be precise enough for the vast majority of applications. However, for underwater environments the use of GPS technology is not possible due to the strong attenuation that affects electromagnetic radiations in underwater mediums. Thus, alternative techniques must be derived for the position determination of underwater Robots.

Underwater navigation can be roughly subdivided in three main branches: Inertial Navigation, Acoustic Navigation, and the more recently emerged Geophysical Navigation. While these different techniques can be used as standalone, they are usually combined into more robust navigation solutions, designed to be capture the strengths of each individual method. Up to recently, the state-of-the-art approach has been combining acoustic and inertial navigation techniques. However, the requirement for using whether acoustic beacons or a support vessel is extremely inconvenient, and even impracticable for some applications. While resurfacing and using a GPS receiver can partially solve for this problem, such behaviour is sometimes undesirable, like during deep water surveys, or even be impossible, if the vehicle is navigating under an iced surface for example. The challenge is then to provide methods that allow AUVs to operate autonomously in highly unstructured environments, and combining Inertial Navigation with Geophysical Navigation methods, might be the way to go.

The advantages and disadvantages of the different underwater navigation techniques have been summarized in Table 2.1. For reference, the remainder of this section provides a brief overview on the existing navigation techniques for Autonomous Underwater Vehicles. A more comprehensive survey, covering both the technology and the algorithms, was provided at different times and by

	Advantages	**Disadvantages**
Inertial Navigation	Self-contained	Drifts over time
Acoustic Navigation	Robust position fixing	Low update rate, deployment of beacons
Geophysical Navigation	Self-contained	Scarcity of maps

Table 2.1: Summary of advantages and disadvantages of the different underwater navigation techniques

different authors, as for example by Kinsey et al. (2006), Stutters et al. (2008) or more recently Paull et al. (2014).

2.2 Inertial Navigation

Inertial Navigation is, as the name indicates, based on inertial principles and uses measurements from Inertial Measurement Units (IMUs) to obtain estimates of both position and velocity using dead-reckoning techniques. Dead reckoning is the process of recurrently estimating a navigation solution of a vehicle using a previously known position and orientation and integrating the vehicle's velocity and acceleration. Inertial Navigation is a self-contained navigation technique with a very good short-term accuracy but its position and velocity estimates are known to drift in time and diverge due to the dead-reckoning of noisy and biased signals.

Modern IMUs consist on a set of accelerometers and gyroscopes, that measure the specific force and angular velocity, respectively. By using a triad of each of these sensors mounted along the different coordinated axis it is possible to obtain, by integration, three-dimensional position and attitude estimates, with good short time accuracy and high update rates. Sometimes IMUs also contain a triad of magnetometers, that measure Earth's magnetic field and allow the estimation of heading. Without external corrections, inertial navigation is known to produce position estimates that will drift and grow unbounded over time. This is mostly caused by the noise and bias present on the signals coming from the IMUs, which are continuously integrated to obtain position and heading estimates.

The fact that Inertial Navigation is self-contained, in the sense that it neither emits nor receives any external signal, is one of its most significant strengths, making it a stealthy navigation solution, immune to interference or jamming. However, due to the dead-reckoning nature of the process, the navigation errors obtained with this method are known to increase and grow unbounded with time, in an extent that is heavily dependent on the accuracy of the sensors used. According to Groves (2013), inertial sensors can be roughly divided in four categories with respect to their levels of maximum horizontal position drifts, as specified on Table 2.2.

While with the marine and navigation grade IMUs it is possible to have AUVs to perform short-term missions with negligible navigation errors, IMUs in this classes are characterized for having dimensions, cost and power requirements that are not compatible the majority of the up-to-date autonomous vehicles. IMUs of the remaining classes, or for longer periods of activity, the use

Grade	Position Drift	Application
Marine	$<$ 1.8km/day	submarine, spacecraft
Navigation	$\sim$ 1.5km/hour	airliner, military aircraft
Tactical	>15km/hour	guided weapons, UAVs
Automotive	n.a.	AHRS

Table 2.2: Inertial Navigation Systems performance categories

of navigational aids is therefore required. For IMUs of the automotive class, obtaining position estimates only from inertial measurements is hardly possible, and this kind low-cost sensors are used mostly for developing Attitude and Heading Reference System (AHRS).

Inertial Navigation Systems (INS) are responsible for solving the standard inertial navigation equations, using data coming from the IMU. Modern INS are also responsible to fuse such information with data from external sensors in an optimal way. In the case of underwater vehicles, sensors like Doppler Velocity Loggers (DVL) or barometric depth sensors are usually employed. For underwater vehicles, the use of acoustic navigation schemes, detailed on the following subsection, has become a standard solution to obtain external positional aids.

2.3 Acoustic Navigation

Acoustic Navigation embraces a number of techniques that rely on the exchange of acoustic signals between a set of acoustic beacons, and one or more vehicles, with the objective of determining the position of the latter. Acoustic Navigation techniques are able to provide position navigational aids, usually in form of slant ranges between each of the beacons or transponders of the acoustic network. These ranges are obtained from the time-of-flight of each the acoustic signals. This is obviously dependent on knowing the velocity of propagation of a sound wave in the water for a given location, which of around 1500 m/s, but varies according to different factors like temperature, depth or salinity. Additionally, in some configurations bearing measurements can also be derived by comparing the time differences of arrival (TDOA) of the same signal when detected by each of the transponders. However, Acoustic Navigation techniques are characterized by a low-update rate. Additionally, it requires the deployment of the acoustic beacons in the area of operation, or in alternative the use of a support vessel, which represents an important drawback, as it significantly affects the costs of such operations.

Broadly speaking, three distinct Acoustic Navigation schemes exist, namely the Long Baseline (LBL), the Short Baseline (SBL), and the Ultra Short Baseline (USBL). If appropriate acoustic interrogation protocols are used, all of these systems can be used for simultaneous external tracking of the vehicle and also to provide relative navigational aids to the vehicles. A detailed review of the different Acoustic Navigation schemes, their individual strengths and their disadvantages has been provided by Vickery (1998).

Acoustic System	Baseline Length
LBL	$>$ 100m to $\sim$ <2000 m
SBL	$\sim$ 20m to 50m
USBL	< 10cm

Table 2.3: Baseline length for the different types of Acoustic Navigation systems

Conventionally, SBL is a ship-based acoustic positioning systems, as the transponder array, consisting on at least 3 individual transponders, is commonly mounted on the hull of a support vessel. One of the main reasons for this is the dimensions of the baseline, that is the distance between the beacons, as indicated in Table 2.3. USBL is yet another acoustic positioning system very similar to SBL. However, while SBL is usually characterized for having the transducers mounted on the hull of the vehicle, quite distant from each other, in USBL systems the transducers are closely mounted to each other. Because of that, USBL arrays do not need to be mounted on a ship hull. In fact, recently USBL techniques have been used for navigation purposes in small-sized AUVs, as for example reported by Morgado et al. (2006)

SBL and USBL can both provide range and bearing between two acoustic beacons. For both SBL and USBL the Time-of-Flight of the acoustic signals is used to compute the ranges between beacons, but only if the velocity of propagation of a sound wave in the water is known. Obtaining bearing, however is done differently for SBL and USBL. On SBL systems, bearing is derived from the time differences of arrival (TDOA), as a ping (transmitted acoustic signal) is detected on each of the transceivers. On the other hand, due to its relatively small baseline, USBL systems compute bearing by comparing the phase of a given ping between individual elements of an acoustic array. Loosely speaking, both these systems have a low level of complexity and don't require the deployment of any additional transponders, but they need to go through a detailed calibration process to obtain optimal precisions on positioning.

Differently from the above methods, LBL systems need to have an array of acoustic beacons deployed on the sea floor, in specific predetermined locations within the are of operation. The position estimates are based on the obtained ranges between the vehicle and the set of beacons. For operating in this configuration, usually four beacons need to be deployed prior to the operation. By employing multilateration techniques, using the ranges to each beacon, it is possible to obtain an estimate of the relative position of the vehicle. In some specific configurations the number of beacons can be smaller but ambiguity with respect to three-dimensional position of the vehicle might arise.

Compared to the previous systems, one of the main advantages of LBL systems is its very good position accuracy, which is independent of the operational depths. Maybe for that reason, LBL is perhaps the most popular of all the acoustic positioning systems. However, the cost and time needed to set up a network, and the later recovery of the beacons, can be quite cumbersome, particularly in adverse environments, and this is still one of the major disadvantage of such systems. To overcome deployment of the beacons on the sea floor, the use of GPS-enabled intelligent buoys has been proposed, in what has been called Inverted LBL. With the use of such systems, the transponders of the bottom are replaced by floating buoys which carry the acoustic transducers. Due to the fact that such devices also carry GPS, calibration of the system can be significantly simplified.

Both traditional LBL or the Inverted LBL require the use of multiple acoustic beacons, deployed prior to operations whether on the sea bottom, or on the surface. In order to simplify the deployment and calibration process, different researchers have focused on developing methods that use only a single acoustic beacon. To name only a few, for example Larsen (2000) proposed the Synthetic LBL method, which combined dead-reckoning with range measurements from a single acoustic source to provide sub-meter positioning accuracy. Later on, LaPointe (2006) presented a simulation study on the use of Virtual LBL, a single beacon navigation systems on which a vehicle position is determined by advancing multiple ranges from a single transponder along the vehicles dead reckoning track. More recently, the work by Ferreira et al. (2010a) addressed the problem of simultaneous localization and control of an AUV using a single acoustic beacon for homing scenarios.

Recent advances in underwater communication topics have also brought Acoustic Modems to play a relevant part in underwater navigation capabilities. Several authors have proposed different frameworks to enable Cooperative Navigation. In Cooperative Navigation, teams of AUVs localize themselves more accurately by sharing position estimates and uncertainty. The interested reader is referred to the works of for example Bahr et al. (2009) or Fallon et al. (2010), among others. However, such approaches require a data link between the vehicles. While this is a interesting approach, acoustic communications are still characterized by small bandwidth, low data rates and high latency and, particularly for shallow waters and adverse environmental conditions, reliable underwater communications can be quite challenging for long distances.

2.3.1 LBL Acoustic Networks

For operating both traditional LBL or the Inverted LBL, usually four beacons need to be deployed prior to the operation. As previously mentioned, by simple multilateration of the ranges it is possible to obtain an estimate of the relative position of the vehicle. Different spherical navigation algorithms can be implemented. In some specific configurations the number of beacons can be smaller but ambiguities with respect to three-dimensional position of the vehicle might arise. For example when a precise depth sensor is present in the vehicle, it is possible to use a network of only three beacons. Additionally, if the trajectories of the vehicle are guaranteed not to cross the baseline, and operational depths are negligible when compared to the baseline, then it is possible to operate in a configuration with only two beacons, as for example presented by Melo and Matos (2008).

The exchange of acoustic signals between beacons and vehicle must follow a predefined protocol. Commonly, the AUVs are responsible to interrogate, in a predefined sequence, all the beacons that are part of the network. Different beacons are independently addressed using distinguishable acoustic signals, usually by using different frequency modulated acoustic signals. In this way,

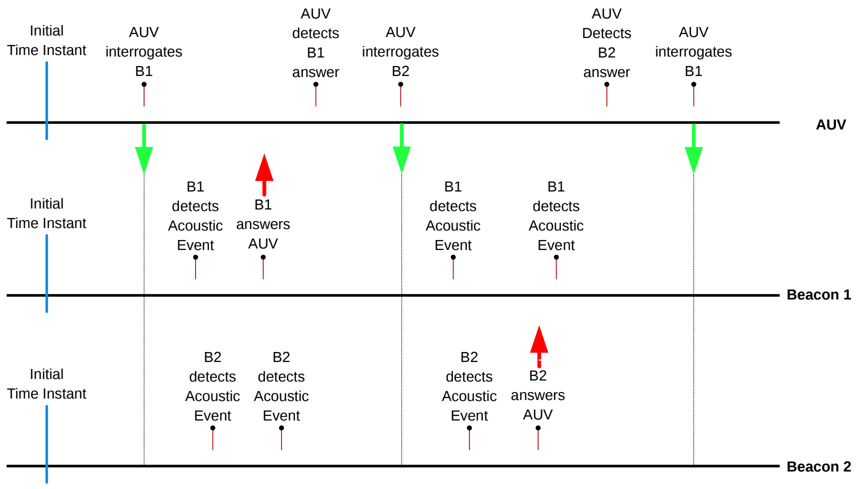


Figure 2.1: LBL Interrogation protocol of an AUV operating in a network with 2 buoys

all the the nodes of the acoustic network can be unambiguously distinguished. A diagram with a timeline representation of such protocol, can be found in Figure 2.1. After being interrogated, a beacon should would for a predefined "turn around" amount of time, and then reply by emitting an adequate acoustic signal. By following this protocol, it is easy for all the devices that compose the network to compute the Time-Of-Flight (ToF) between vehicle and beacons. After computing the ToF, it is then possible for the vehicle to estimate its slant range to each of the beacons of the network, given that the speed of sound is known. For obvious reasons, this ToF, which is the elapsed time since the interrogation of the vehicle until reply was received, can also be referred to as the Two-Way-Travel-Time (TWTT). Current state-of-the-art navigation algorithms, combining both Inertial and Acoustic Navigation techniques, fuse the obtained slant ranges with inertial measurements, yielding an optimal estimate of the vehicle position and attitude.

It is reasonable to expect that in a near future new applications, requiring the operation of multiple AUVs, will arise. Thus, the pertinence of this issue becomes obvious. Nevertheless, due to its nature, conventional LBL acoustic navigation schemes like the one just described still face some challenges when trying to cope with navigation and tracking of multiple vehicles.

2.3.2 One-Way-Travel-Time LBL Navigation

LBL is perhaps the most popular of all the acoustic positioning systems, mostly due to its precision, reliability and robustness. However, scaling up these system for operations with multiple vehicles has no easy and standard solution. LBL systems provide navigational aids to AUVs in terms of ranges to each one of the beacons that compose the network. Standard implementations depend on the AUV independently sending a query acoustic signal to each one of the beacons and waiting for their reply. The time difference between sending the query and receiving the reply is then converted to the slant range between AUV and beacon.

Standard LBL systems already require the use of signals with different frequencies to address each acoustic beacon. Using the same strategy for different vehicles is difficult, mostly due to the scarcity of available frequencies. Moreover, the use of such gimmick leads to the necessary increment in overall complexity of system, mostly due to the need to develop dedicated hardware, e.g. finely tuned filters. It also has non-negligible cost and set up time. Alternatively, time multiplexing schemes, similar to the Time Division Multiple Access (TDMA) method, could also be envisioned. In this case, the different vehicles would need to wait for its slot time to be able to interrogate the acoustic beacons. However, the big drawback of such approach is that the interval of time in between two successive interrogation slots can easily become too long, thus degrading the navigation accuracy of the system.

A simple solution has been proposed for scaling LBL algorithms to provide navigational aids to multiple vehicles, which is named One-Way-Travel-Time (OWTT) Acoustic Navigation. OWTT techniques require clock synchronization between all the vehicles and beacons present in the acoustic network. This generally implies having low-drift clocks in all of the systems. For this arrangement, the beacons are configured to periodically send acoustic signals at predefined time instants (e.g. once per second), while the vehicles are purely passive. OWTT systems rely on a protocol which defines the exact instant when each of the beacons is supposed to emit its acoustic signal. Because all the clocks are synchronized, the time elapsed from the time instant the beacon is supposed to transmit until each of the vehicles detects it, also known as the one-way-travel-time, can be directly converted to ranges. In some configurations the beacons are required to transmit signals distinguishable between each other. Alternatively, and because the number of the beacons is small (usually lower than 4), Time Division Multiple Access (TDMA) techniques can also be used. An early example of OWTT navigation of multiple vehicles was provided by Eustice et al. (2007).

2.4 Geophysical Navigation

In the broad sense, Geophysical Navigation refers to the use of geophysical features, like the terrain profile, Earth's magnetic field or even Earth's gravity field to estimate position of a vehicle. These methods are referred to, respectively, as Terrain Based Navigation, Geomagnetic Based Navigation, and Gravity Based Navigation. As stated by Leonard et al. (1998), at its foundation this kind of navigation relies on matching sensor data with an *a-priori* environment map, under the assumption that there is sufficient spatial variation in the parameters being measured to permit accurate localization. Some authors have a more strict understanding of the term Geophysical Navigation, and use it only to designate navigation algorithms based on geopotential fields, like the gravitational or magnetic fields (Teixeira, 2007).

Perhaps the biggest advantage of these map-based navigation methods is their ability to estimate the global position of a vehicle. This compares to the relative positioning characteristics of Acoustic Navigation methods. Moreover, these methods are completely on-board navigation systems, without the need for external devices, granting the vehicles a large operational range (Meduna, 2011). This can be particularly interesting when there is the need to perform covert or stealthy missions. Notwithstanding, there are still some open issues that need to be addressed when resorting to this navigation systems. For example, geopotential fields scale non-linearly with the distance to the respective sources, which can represent a problem with non-trivial solution. At the same time, map representation, computation and management issues can arise when dealing with magnetic or gravity fields, due to their multidimensionality nature.

2.4.1 Terrain Based Navigation

Underwater Terrain Based Navigation (TBN), to which research efforts have been recently focused, is a potentially powerful self-contained solution for long-range navigation of AUVs. Similarly to the use of GPS or even Acoustic Navigation, TBN uses information of the variations of the terrain to bound the error growth of inertial navigation, thus increasing the long term estimation accuracy of the position of the vehicle. TBN is also sometimes referred to as Terrain Relative Navigation (TRN) or Terrain Aided Navigation (TAN). Even though these terms are interchangeable, in the remainder of this thesis only the Terrain Based Navigation denomination will be used. In this introductory subsection, a brief description of Terrain Based Navigation for underwater vehicles is given, but a more detailed overview is provided in Section 5.

Terrain Based Navigation is a self-contained technique, in the sense that no external aiding signals or devices are needed. This is in fact a great advantage when comparing to Acoustic Navigation. Because of this unique feature, TBN has the potential to dramatically improve the autonomy of AUVs, given that appropriate terrain maps of the areas to navigate are available. TBN for underwater environments is fairly recent, at least when compared to aerial techniques. The main differences between the methods developed for the two environments are mostly related to the sensors used and, naturally, the vehicle dynamics.

While aerial TBN has been focused mostly on the use of single beam sensors, usually radar or laser altimeters, the underwater community has been focused on using dense sensors. Since the early days, the use of TBN for underwater vehicles has been highly focused towards the use of powerful multiple beam sonars, able to map large areas of terrain within a single measurement acquisition step, providing in this way a high resolution perception of the environment. Experimental validation of such approaches was also consistently coupled with the use of high-grade INS. Lately some research efforts have been directed on a rather opposite approach, which is the application of TBN techniques to sensor limited systems. Since the larger group of existent AUVs is, by far, equipped with low accuracy inertial and bathymetric sensors, the rational of such approach is evident. Meduna et al. (2008) developed a tightly-coupled TBN filtering framework for sensor-limited vehicles tightly integrated with the on-board navigation system. The approach, which basically relies on the estimation of critical sensor errors, envisioned the use of low information range sensors like DVL or altimeters. The success of the approach was also experimentally demonstrated in different scenarios like long range navigation and return-to-site missions. The use of DVLs or single beam sonars for TBN purposes has also been reported by other authors (Donovan, 2011; Kim and Kim, 2014).

One of the main factors contributing to the success of terrain navigation systems is terrain variability. Large variability in the terrain is known to positively contribute to the convergence of the estimation algorithms, and flat areas are expected to present poor performance. Even though TBN techniques have been demonstrated to perform well, in some situations the filter can converge to the wrong location, especially if the vehicle is operating in particularly uninformative terrain. Ånonsen (2010) predicated that TRN occasionally converges to overconfident, incorrect solutions when operating for an extended period of time over featureless terrain. Noting that, Dektor and Rock (2012) analysed the causes for the filter failure when in flat terrain. According to the author, the cause of false fixes in information-poor regions is the assumption that the terrain is uncorrelated. Dektor and Rock (2014) further address this issue by developing an adaptive variance algorithm, which is dependant on the amount of map errors, sensor errors, and information of the terrain. On a complementary line of work Houts et al. (2012) designed a robust framework for failure detection and recovery for terrain navigation . This framework builds up on a series of diagnostic checks and, in particular, on checking whether the measurements predicted by the estimate agree with the observed measurements.

2.4.2 Geomagnetic and Gravity Based Navigation

Geomagnetic Based Navigation is in fact a very similar problem to TBN, with the main difference being the features present in the pre-surveyed digital map. While for TBN the map represents terrain elevation, in geomagnetic navigation Earth's Geomagnetic field is used. Comparing with TBN, Geomagnetic Navigation is still a very recent topic, and literature addressing the topic is still scarce. While for TBN sonars are used to perceived the bottom, Geomagnetic Navigation relies only on magnetic sensors, which are passive sensors, to obtain measurements of the local Geomagnetic Field. This is an obvious advantage particularly for military operations, which sometimes are required to be stealthy.

Given the similarity between Terrain Based Navigation and Geomagnetic Navigation, all the matching techniques could, in principle, be applied to the latter, up to some minor differences. Different authors have addressed the application of Geomagnetic Navigation to underwater environments, namely Mu et al. (2007), Ren et al. (2008), Zhao et al. (2009) or Wang et al. (2010). Nonetheless, only a few conclusive experimental results were reported, for example by Rice et al. (2004). Geomagnetic Navigation techniques can also use artificial magnetic fields that present a stable magnetic signature, permitting a more precise navigation. Teixeira (2007) demonstrated an efficient implementation of a Geophysical Navigation algorithm by introducing the concept of maps of invariant gradients of the geomagnetic field. The method was based on measuring the vertical gradient of the Geomagnetic Field, thus contributing to the elimination of important sources of errors.

Analogously to Terrain Based Navigation and Geomagnetic Based Navigation, using the Earth's Gravity Field can possibly provide a valid resource when trying to obtain an absolute positioning method. By using gradiometers it is possible to measure Earth's local Gravity Field and, thus, derive a similar navigation method. However, such state-of-the-art devices, accurate enough to distinguish between vehicle accelerations, Coriolis acceleration and the gravity itself are still at a cost that is in most cases impracticable. A few studies have been done regarding the requirements and performance evaluation for Gravity Based Navigation (Jircitano et al., 1990; Lee et al., 2015). Also, some authors proposed the simultaneous use of both Gravity Gradient and Terrain information to obtain more robust and accurate navigation solutions (Liu et al., 2009). However, up to the authors knowledge, no successful implementation of systems based on the Gravity gradient was reported.

2.5 Bayesian Estimation

The objective of any estimation procedure is to obtain the value of a parameter x, given an observation of an experimental outcome y. This has been mathematically stated most notably as the Bayes theorem:

$$p(x|y) = \frac{p(y|x)p(x)}{p(y)} \tag{2.1}$$

Bayesian Estimation adapts the Bayes theorem to a statistical paradigm. In the Bayesian Estimation framework everything that is unknown is considered a stochastic variable, described by its respective probability distribution. The goal of Bayesian Estimation is then to compute the posterior distribution of the state vector given a set of observations.

At the core of Bayesian Estimation framework is the Bayes Filter, a general probabilistic tool that has been extensively used to recursively estimate the state of linear and nonlinear stochastic systems, $\mathbf{x}_t$, using noisy measurements as observations, $\mathbf{z}_t$. Under markovian and mutually independence assumptions, using the Bayes formula (2.1) and the law of total probability, the following Bayes Filter recursion can be obtained:

$$p(\mathbf{x}_t|\mathbf{z}_t) = \frac{p(\mathbf{z}_t|\mathbf{x}_t)p(\mathbf{x}_t|\mathbf{z}_{t-1})}{p(\mathbf{z}_t|\mathbf{z}_{t-1})} \tag{2.2}$$

$$p(\mathbf{z}_t|\mathbf{z}_{t-1}) = \int p(\mathbf{z}_t|\mathbf{x}_t)p(\mathbf{x}_t|\mathbf{z}_{t-1})d\mathbf{x}_t \tag{2.3}$$

$$p(\mathbf{x}_t|\mathbf{z}_{t-1}) = \int p(\mathbf{x}_t|\mathbf{x}_{t-1})p(\mathbf{x}_{t-1}|\mathbf{z}_{t-1})d\mathbf{x}_{t-1} \tag{2.4}$$

(2.2) is the measurement update equation, where the term $p(\mathbf{z}_t|\mathbf{z}_{t-1})$ is a normalization constant, evaluated as in (2.3). The measurement update equations are determined by the measurement model, $p(\mathbf{z}_t|\mathbf{x}_t)$ and the prior, $p(\mathbf{x}_t|\mathbf{z}_{t-1})$. On the other hand, Equation (2.4) corresponds to the time update equation, and this is determined by the state transition density, $p(\mathbf{x}_t|\mathbf{x}_{t-1})$.

The Bayes Filter is not analytically solvable for the general case, mostly due to the complexity involved on evaluating the aforementioned integrals. However, under some assumptions, different realizations of the Bayes Filter can be implemented. Examples of this are parametric filters like the Kalman Filter (KF), the Information Filter (IF) or even the Extended Kalman Filter (EKF), and non-parametric approaches like the Particle Filter (PF) or the Point Mass Filter (PMF). For details on the theoretical derivation of such filters, the interested reader should refer to works by Bergman (1999), Arulampalam et al. (2002), Chen (2003), and the references therein.

Bayesian filters have become a *de facto* standard for sensor fusion in integrated navigation systems. Morice et al. (2009) stated that the strength of Bayesian filtering techniques is that they allow the fusion of information from multiple sensors, taking into account both the sensor measurements and the accuracy of the sensors. Moreover, these probabilistic algorithms also allow to explicitly deal with uncertainties that might exist.

The Bayes Filter can be optimally implemented as a Kalman Filter under the assumptions that the state and process noise are mutually independent Gaussian distributions, and both the state transition probability and observation function can be represented by means of a linear stochastic equation. When this is not the case, Extended Kalman Filters are able to deal with non-linear problems by using an approximation based on the first-order Taylor series expansion around a working point. However, the Taylor expansion requires the computation of Jacobian matrix, which sometimes is non-trivial, particularly when in the presence of highly non-linear systems. In such cases, the computed gradient can negatively influence the accuracy of the approximation, and therefore result in poor estimates.

The Sigma Point Kalman filter (SPKF), also known as Unscented Kalman filter (UKF), is a nonlinear realization of the Bayes filter that relies on the use of only a few deterministically chosen sigma points, which are then propagated through the existing nonlinearities, to yield a new sigma point approximation of the posterior density. However, due to is characteristics and simplicity, non-parametric realizations of the Bayes Filters, like the Particle Filter or the Point Mass Filter have gained more attention in the recent approaches. When in presence of strong non-linear systems, the use of a full Bayesian non-parametric filter is advised. While those are in general more accurate in representing non-linear systems and simple to implement, its main drawback is related with their computational complexity, known to scale exponentially with the state vector's dimension.

The Particle Filter, also sometimes referred to as Sequential Monte Carlo Method, is a numerical approximation to the Bayesian filter for non-linear systems. The key idea in the Particle Filter is to represent the required posterior density function by a set of random samples with associated weights, and then compute estimates based on these samples and weights (Arulampalam et al., 2002). It can be demonstrated that as the number of samples, or particles, increases, the discrete weighted approximation becomes an equivalent representation to the usual posterior density. The particle filter is also known to be asymptotically optimal in a minimum mean square sense.

The Point Mass Filter is yet another non-parametric realization of the Bayes Filter, even though not as popular as the PF. In this case, the posterior density is assumed to be represented by a set of point masses, ordered in a grid. The continuous PDF is obtained by integrating over the masses of the grid. Among the main advantages of such filters is their ability to solve the Bayesian Filter in an asymptotically optimal way. Even though the size of the grid is usually fixed, efficient implementations of the PMF use grid adaptation mechanisms to automatically adjust the grid mesh, which is particularly useful in situations when a high gradient PDF exist. This grid-adaptation mechanism can also be regarded as a trade-off between algorithm performance and computational requirements.

The Rao-Blackwellized Kalman Filter (RBPF), also known as Marginalized Particle Filter, is another widely used realization of the Bayes Filter. The RBPF is a combination of the Particle Filter and the Kalman Filter, which can be used when the underlying non-linear system contains a linear sub-structure, subject to Gaussian noise. The Particle Filter is used to address the non-linear sub-structure, while simultaneously the Kalman Filter is used address the linear sub-structure. In such cases, using an RBPF is often more efficient that the plain non-linear filters mentioned before.

When dealing with estimation, it is always convenient to establish a lower bound on estimation error that is possible to achieve. According to Bergman (1999) such a bound can then be used to quantify the fundamental performance level that can be reached for the currently studied estimation problem. The Cramér-Rao Lower Bound (CRLB) is known to set a lower limit on the mean square estimation error of every estimator. Therefore, the CRLB is used to assess the performance of a filter by evaluating how far from the CRLB the estimates are.

2.5.1 State Space representation

The state-space representation is a natural framework to formulate the general problems of navigation of autonomous vehicles. A generic continuous-time stochastic filtering problem can by described in a dynamic state-space form according to the following equations (Chen, 2003):

$$\dot{\mathbf{x}}_t = f(t, \mathbf{x}_t, u_t, w_t) \tag{2.5a}$$

$$\mathbf{y}_t = g(t, \mathbf{x}_t, u_t, v_t) \tag{2.5b}$$

In (2.5) $\mathbf{x}_t$ refers to vector of state variables, u_t refers to the system inputs, and t denotes time. wt and v_t represent process noise and measurement noise, respectively. Despite its simplicity, the above model is rather general and can be applied in most of the applications. The function $f(t, \mathbf{x}_t, u_t, w_t)$ describes the system dynamics, and determines how the states evolve over time. On the other hand $g(t, \mathbf{x}_t, u_t, v_t)$ is the measurement equation and describes how observations of the system are made, and how do they relate to the system state, $\mathbf{x}_t$. The process and measurement noise components are generally unknown and modelled as stochastic processes.

From (2.5), and considering noise of additive nature, a basic discretized state-space model for a general non-linear time-invariant system, can be expressed by the following difference equations:

$$\mathbf{x}_k = f(\mathbf{x}_{k-1}, u_k) + w_k \tag{2.6a}$$

$$\mathbf{z}_k = g(\mathbf{x}_k) + v_k \tag{2.6b}$$

The difference equations (2.6) try to capture the fundamental idea of the general navigation problems, on which new observations $\mathbf{z}_k$ are used to correct the the system state, $\mathbf{x}_k$. The characteristics of such problem are described by both the state transition equation (2.6a) and measurement model equation (2.6b). Evidently, these are all system dependant, usually modelling position and attitude states of the vehicle. In order to address the limitations of sensor-limited systems, the system state-space representation is usually augmented to accommodate the estimation of different noise sources that affect the system. In this way, it is possible to overcome, up to some extent, the limitations of such systems.

When the dynamical system has a state space representation as in (2.6), the densities $p(\mathbf{x}_k|\mathbf{x}_{k-1})$ and $p(\mathbf{z}_k|\mathbf{x}_k)$ of the Bayes Filter can be easily computed if the noise densities w_k and v_k are known, as indicated by (2.7). Plugging this densities in a Bayesian framework then becomes straightforward.

$$p(\mathbf{x}_k|\mathbf{x}_{k-1}) = p_{w_k}(\mathbf{x}_k - f_t(\mathbf{x}_{k-1}, u_k, k)) \tag{2.7a}$$

$$p(\mathbf{z}_k|\mathbf{x}_k) = p_{v_k}(\mathbf{z}_k - h(\mathbf{x}_k, k)) \tag{2.7b}$$

2.6 Experimental Systems

This section will briefly introduce the autonomous marine vehicles that have been used for the experimental validation of the algorithms that will be presented in the following chapters. The vehicles used, one AUV and a set of two ASVs, were fully designed by INESC TEC researchers, and have been in continuous development and improvement process for the past several years. Besides the vehicles, the beacons and acoustic navigation systems that will be used in Chapter 6 and 7 will also be briefly described.

2.6.1 MARES

The AUV used to experimentally support, at least partly, the work developed in this thesis is the MARES AUV. This vehicle, depicted in Figure 2.2, is a torpedo-shaped, highly modular, small sized sensor-limited AUV, with about 1.5m long, 32kgs of weight, and propelled by four thrusters.

The thrusters are grouped in two pairs, two horizontal ones, located at the tail of the vehicle, and two vertical ones. This particular configuration of the thrusters allows for an almost decoupled control of the horizontal and vertical motion of the vehicle. Therefore, surge velocity and yaw can be generically controlled by the horizontal thrusters, while heave and pitch of the vehicle are controlled by the vertical ones. Such vehicle feature can be particularly appreciated in some contexts. A more detailed description of the vehicle can be found in Cruz and Matos (2008).

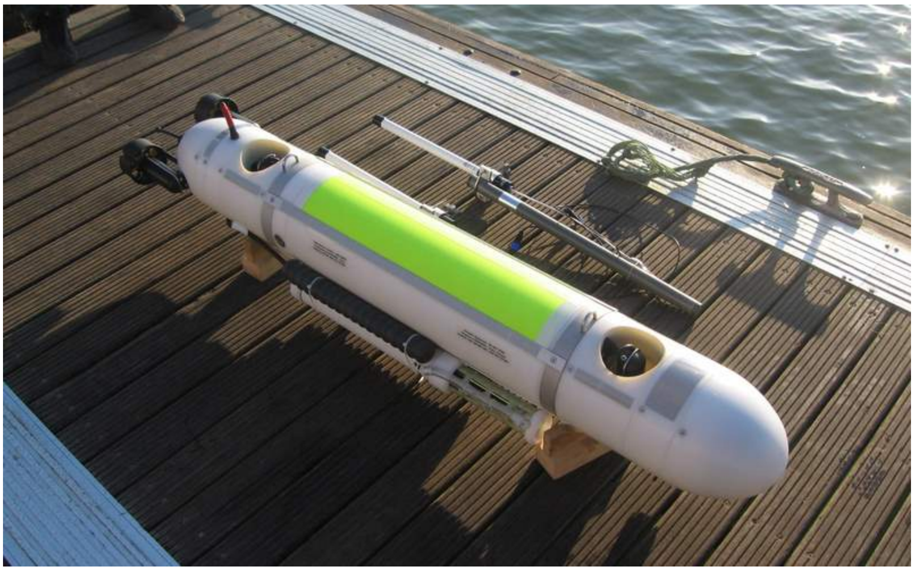


Figure 2.2: The MARES AUV

2.6.2 AUV surrogates

When using and experimentally validating different algorithms for AUVs, it is sometimes difficult to obtain an independent verification of the vehicles trajectory. This is particularly evident when trying to assess the performance of Acoustic Navigation systems. For that reason, in some trials it can be more appropriate to use Autonomous Surface Vehicles (ASVs) as AUV surrogates, providing an easier approach to independently establish ground-truth data. The idea behind such idea that the ASVs can easily mimic AUVs navigating at a constant depth. By doing so it is possible, on one hand, to experimentally validate some of the algorithms developed for sensor-limited AUVs, and on the other hand have the necessary ground-truth data of the position of the vehicles, provided by the GPS receivers that equip the ASVs.

In some of the field trials described ahead two ASVs were used, namely the ASVs Gama and Zarco, depicted in Figure 2.3. Gama and Zarco are two small sized catamaran based craft, designed to operate in quiet waters, and can reach speeds of up to 2m/s. These vehicles can be remotely operated or autonomously perform pre-programmed missions. The vehicles are equipped with a set of navigation instruments, including a high-precision GPS receiver, which provide an accurate positioning level, a WiFi link for real time connection with shore, and the necessary acoustic transceiver. For more details regarding these vehicles, the interested reader should refer to Cruz et al. (2007)

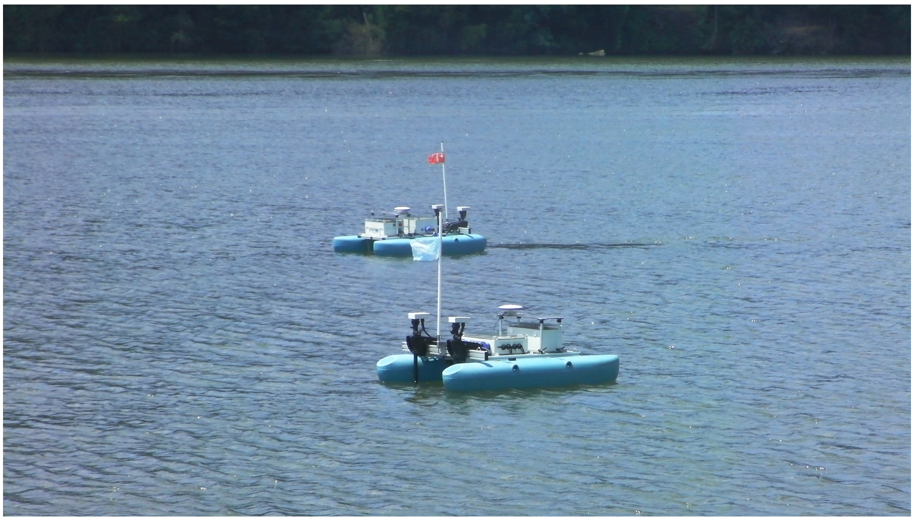


Figure 2.3: The ASVs used in the field trials

2.6.3 Navigation Beacons

The experimental validation of the Acoustic Navigation methods that will be presented in the chapters ahead requires the use of navigation beacons. The acoustic Navigation Beacons used for that purpose are depicted in Figure 2.4. These navigation beacons are man portable acoustic navigation buoys, fully envisioned and designed by INESC TEC researchers, and have been thoroughly described in Almeida et al. (2016a). Beacons based in buoys are appropriate for acoustic networks based on Inverted LBL, on which the beacons are deployed at the sea surface.

In terms of their hardware characteristics, the beacons feature an embedded computer running, GPS, WiFi and a serial radio for long range communications. Besides providing position information for the vehicles, GPS also provides a high-precision synchronized clock source. They are also fitted with a pack of rechargeable batteries, and an acoustic transducer. This transducer, that enables the emission and detection of the acoustic signals, is controlled by a proprietary Acoustic Control System, that will be described in the following subsection.

2.6.4 Acoustic Systems

An acoustic navigation network is usually composed of nodes of two kinds, the acoustic beacons and the vehicles, as detailed above. Nevertheless, all the nodes are required to be equipped with an Acoustic Navigation System, that will be briefly described in this section.

The acoustic system here presented is composed of an acoustic transducer and an Acoustic Control System. The transducer, operating in frequencies in the range of 20 to $30KHz$ range, allows the emission and reception of the acoustic signals. On the other hand, the transmission and detection of the signals is handled by a proprietary Acoustic Control System (ACS), composed of a set of electronic boards, that can be seen in Figure 2.5. Besides the signal conditioning and filtering electronics, these boards also implement precision timing electronics that enable an accurate measurement of the Time-Of-Flight of the acoustic signals.

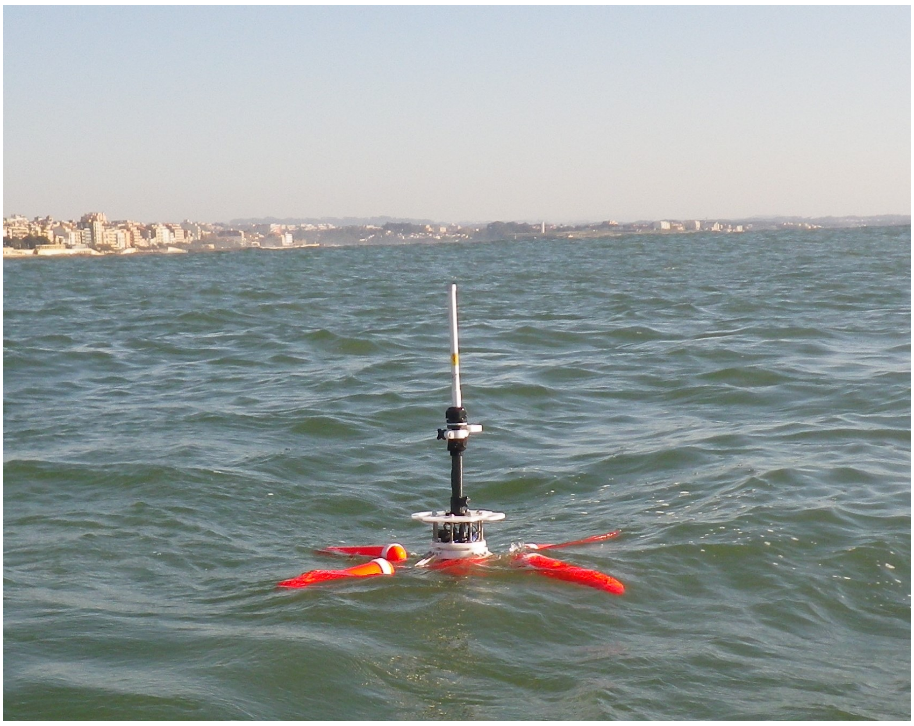


Figure 2.4: The man portable acoustic navigation beacon buoys

The system is flexible enough to enable measuring both the One-Way-Travel-Time or the Two-Way-Travel-Time, depending on the desired configuration. A complete characterization of the range measurements that can be obtained by this system has been discussed in Almeida et al. (2016b). Under optimal conditions, precisions of up to 15*cm* can be achieved.

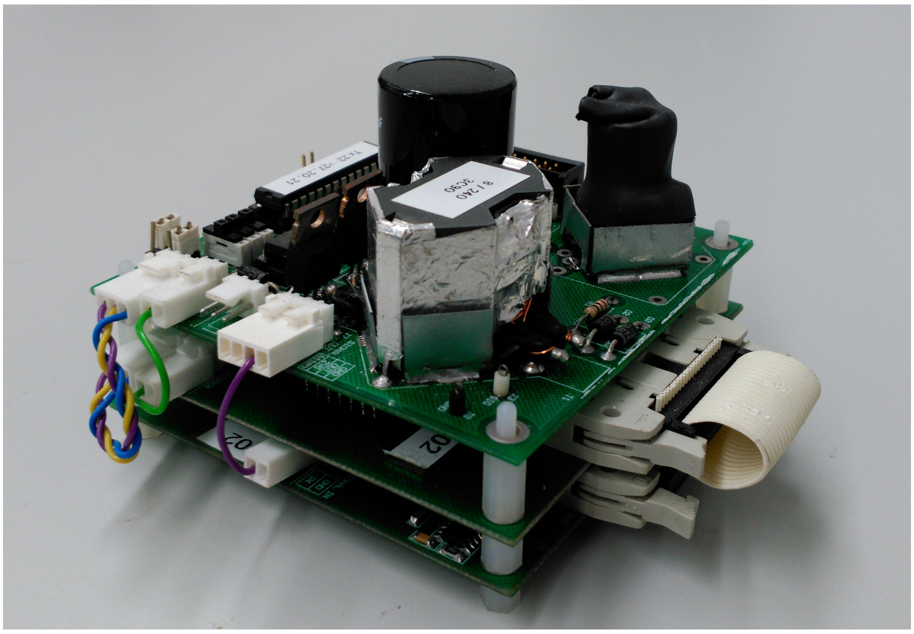


Figure 2.5: Acoustic Boards responsible for controlling the emission and detection of the signals

Part II

Navigation using Natural Features

Chapter 3

Bottom Estimation and Following

This chapter addresses the problem of bottom following by an Autonomous Underwater Vehicle in an environment which is not previously known. In particular, the focus is on integrating a reactive behaviour based on environment sensing, with the on-board navigation software of the MARES AUV. For this, a guidance algorithm will provide the necessary pitch and depth references to the control layer of the vehicle. While range to the seabed can be measured with an altimeter, the pitch reference values are based on an online estimation of the slope of the seabed. By doing so, it is possible to control the vehicle in a way that it will always maintain a constant attitude towards the bottom, and the trajectory followed will remain parallel to bottom, regardless of it's profile.

3.1 Introduction

Traditional applications for the use of Autonomous Underwater Vehicles (AUVs) are mostly related with bathymetric tasks, where the objective of mapping the bottom of the river or sea is achieved by using advanced ultrasound equipment. However, other applications for these vehicles have been envisioned, especially in open waters environments where the benefits of using them are more dramatic. Nowadays, AUVs are already being used for variety of missions, including the inspection of the bottom, inspection of underwater structures and even remote environmental sensing within oceanographic expeditions.

Performing visual inspection of the bottom with an AUV obviously requires the vehicles to navigate closely to the bottom. With poor lighting conditions and turbid water, the bottom of the sea is usually a quite adverse environment for image acquisition. Whenever this is necessary, the vehicle needs to navigate as close to the bottom as possible in order to obtain satisfactory results. Such inspection tasks would also greatly benefit if the trajectories of the vehicle closely resemble the profile of the bottom. In that way bottom features would be depicted according to their natural size and orientation ratio, decreasing distortion and other disturbances that otherwise may affect the final images. In known environments navigating close to the bottom does not represent a challenging navigation problem, as it is easy to plan ahead a given trajectory. In most cases, however, it is not possible to known in advance the profile of the bottom and the problem of having an autonomous vehicle navigating close to the bottom becomes non-trivial, and could even put in danger the safety of the vehicle.

The problem of Bottom Following by an AUV, or seabed tracking, was described in one of the initial works on the topic by Bennett et al. (1995) as "maintaining a fixed altitude above an arbitrary surface whose characteristics may or may not be known". Throughout time different authors have addressed this topic in the literature. For example, in a series of articles, Caccia et al. (1997, 1999, 2003) provided a very interesting and detailed study on the use of high-frequency pencil beam profiling sonar for bottom following scenarios. The solutions proposed were based on a multi-hypothesis Extended Kalman Filter, for motion and environment estimation techniques, and a Lyapunov-based guidance system. Yoerger et al. (2000) described a survey where bottom-following tasks were extensively done to gather high-precision detailed data from the seabed with the help of an acoustic range finder.

Gao et al. (2008) proposed the use of the potential field method to derive a controller that addresses both the problems of bottom avoidance and bottom following with a two-level hierarchical control approach. The first step includes the motion planning phase, on which the commanded pitch angle is generated based on measurements from altimeter and depth sensor; the subsequent execution control phase, on the other hand, is responsible for the regulation of the stern rudder to track pitch references using for that a linear sliding mode controller. Silvestre et al. (2009) proposed a different approach on which the bathymetric characteristics ahead of the vehicle are measured by two echo sounders and taken into account and preview control theory is used to develop a suitable bottom following controller. Adhami-Mirhosseini et al. (2011) adopted a different strategy, by converting the bottom-following problem into a trajectory tracking problem. In the described method, first a Fourier series expansion of the seabed profile is obtained, and then used with the nonlinear output regulation framework to address the seabed tracking problem.

The method presented in this chapter is different on what can be found in elsewhere in the literature, in the sense that is not a control-based approach, but instead a guidance based approach. At the same time, it was important to develop a strategy suitable for being used by sensor-limited systems, with only a very simple sonar sensor, like an altimeter. The derived strategy relies on using altitude measurements given by a sonar to estimate the altitude to the bottom and it's slope, and then generate the respective control variables. By doing so, the trajectory performed by the AUV in the vertical plane will closely resembles the profile of the bottom. Performing this kind of trajectory is of particular importance for applications where acquiring images of the bottom is needed.

The chapter is organized as follows. Section 3.2 gives a brief presentation of the MARES AUV architecture and controller structure. This is relevant to understand the sensor-based approached followed. Section 3.3 will further discuss the necessary robust outlier removal mechanism introduced, in order to obtain a smooth measurement set. Section 3.4 deals with the different real-time slope estimation algorithms studied, and Section 3.5 details the computation of the suitable control variables. The results obtained with the proposed solution, both by simulation and in real missions, are presented in Section 3.6 and finally some conclusions are discussed in the last section.

3.2 System Architecture

The guidance algorithm proposed in this chapter is to be integrated in the on-board control software of the MARES AUV. In this context, and for better understanding, this section presents an overview on the architecture of the control system implemented in the vehicle.

The control layer of MARES is composed by four independent controllers: surge, heading, pitch and depth. The surge and heading controllers are responsible for the horizontal motion of the vehicle, while the pitch and depth controllers are combined to obtain the actuation in the vertical plane. This decoupling of horizontal and vertical motions of the vehicle is possible mostly due to the geometrical properties of the vehicle, and its thruster configuration, as described in Section 2.6. Each one of the four basic controllers can operate in either open or closed loop mode. A more detailed description of each of the controllers was given by Matos and Cruz (2009).

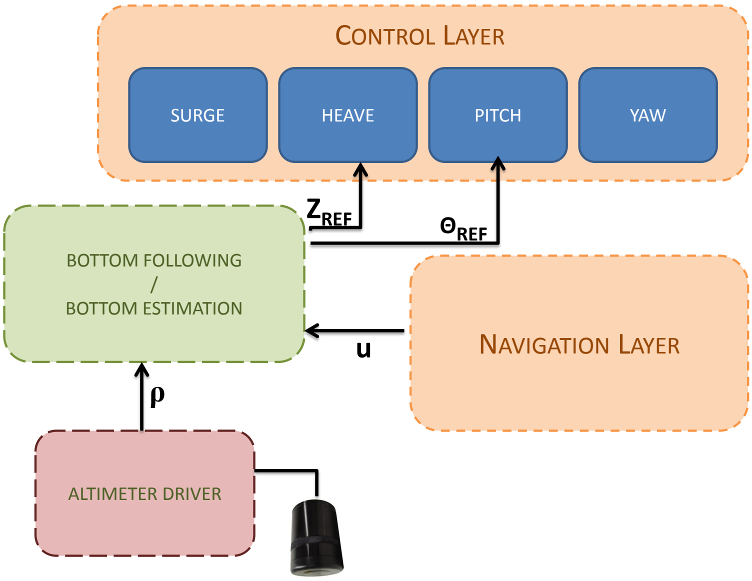


Figure 3.1: Schematic view of the MARES AUV system architecture and controller structure

The control layer of MARES also defines a manoeuvre as a set of coordinated control actions by each of the basic controllers. Four elementary manoeuvres - dive, surface, goto, hover - were defined in the core of the control system, but it is also possible to define additional manoeuvres, as each of the basic controllers can be independently actuated. This flexibility of the control layer allows to further define more entangled manoeuvres in a very efficient way. Figure 3.1 provides a schematic view of the MARES AUV system architecture and controller structure, together with the blocks for the implemented strategy.

The guidance algorithm to implement will take advantage of this flexibility, providing the desired depth and pitch references to the respective controllers. In this way, it is possible to independently control the vertical trajectory of the vehicle. A note to the fact that this implementation is entirely guidance-based and independent of the already existing four basic controllers, as opposed to some of the bottom following approaches present in the literature (Gao et al., 2008; Adhami-Mirhosseini et al., 2011).

In the bottom following manoeuvre, the vehicle should maintain a constant distance towards the seabed, while its pitch should vary in accordance to the slope of the bottom . The altitude of the vehicle off the seabed can be easily measured by using an altimeter, nowadays a very common single beam sonar sensor, that provides range distance measurements. However, measuring the slope of the bottom is something that is not so trivial and has to be estimated. Given that the slope of the bottom is closely related with variation of the relief of the bottom, its slope can be estimated from the altimeter measures. Different techniques for slope estimation are discussed in Section 3.4. The control variables, pitch and depth, are then computed from these estimates and fed the pitch and depth controllers, respectively.

The most reliable way of assessing the distance to the bottom inside the water is using sonar techniques, mostly due to the unique characteristics of sound propagation in the water. The Imagenex Model 862 sonar was integrated in the MARES hardware to obtain range measurements towards the bottom, indicating in this way distance of the vehicle to the bottom. This is a completely self-contained altimeter with a narrow conical beam of 10º, providing range measurements at rate up tp 4Hz with range resolutions of 20mm. The altimeter, mounted in the vehicle on a downward facing position, is responsible for providing range measurements of the distance towards the bottom, and the different parameters were fine-tuned up to a state on which the altimeter was providing consistent measurements throughout the time.

3.3 Filtering

The output of the altimeter, when its configuration parameters are properly set, presents range measurements that are consistent throughout time. Despite that, and as expected, these measures still present some noise, most of the times in the same order of magnitude of the quantum of the sensor, which is 2cm. Moreover, this effect is more noticeable when the sensor is sending acoustic waves while moving horizontally, for example when mounted on a vehicle which is moving with appreciable speeds. Notwithstanding, the output of the altimeter always needs to be filtered, to prevent the appearance of eventual spurious measurements or outliers, frequent when using underwater sonar.

The ranges measured by altimeter are supposed to generate proper depth references to be fed to the control of the vehicle, therefore they need to present a relatively smooth behaviour. The need for filtering the raw altimeter measurements naturally arises: on one hand outliers and spurious measurements need to be eliminated, and on the other hand, this stream of measures needs to be smoothed out. On top of that, it must be ensured that the delay introduced by the filtering process does not influence the control of the vehicle. Even though the vehicle dynamics are slow, delays higher the 0.5 seconds are already considered significant. An example of the raw output of the altimeter can be seen in Figure 3.2, where the presence of outliers is clear.

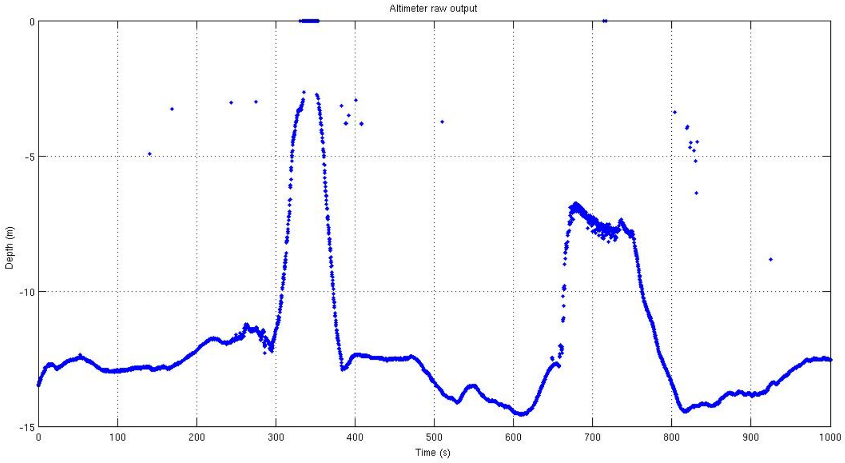


Figure 3.2: Raw output of the altimeter; outliers are clearly identified

Given the filtering requirements, a choice for a one-dimension Kalman Filter came naturally, as it provides not only efficient smoothing, but also has the possibility to discard outliers by simply evaluating the covariance of the innovation. The state model for our systems is therefore unidimensional with its state being the depth, as given by the altimeter. The process model of the system was chosen to be a first order moving average:

$$z_{k+1} = z_k + u_k \tag{3.1}$$

Equation 3.1 tries to express the fact that the depth, z, should vary only by influence of the motion of the vehicle on the vertical plane. In that sense, $u_k = u\sin\theta + v\sin\theta$, where θ is the pitch angle of the vehicle and u and k are, respectively, the surge and heave movements of the vehicle.

The Kalman Filter algorithm, now briefly described, is divided in two different phases. The time update phase, where the current state is projected ahead in time according to the system model, is described by the equations below:

$$\begin{aligned} X_{k+1} &= AX_k + Bu_k \\ P_{k+1} &= AP_kA^T + q \end{aligned} \tag{3.2}$$

In the same way, the equations describing the measurement update phase are given by (3.3). In the measurement update new measurements are incorporated in the state of the filter, with the projected estimate of the state adjusted by an actual measurement.

$$\begin{aligned} S_{k+1} &= HP_kH^T + r \\ K_{k+1} &= P_kH^TS_k^{-1} \\ X_{k+1} &= X_k + K_k(z_k - HX_k) \\ P_{k+1} &= (I - K_kH)P_k \end{aligned} \tag{3.3}$$

For the filter in question, $[A] = 1$ and $[B] = 1$. Because the distance to the bottom can be directly measured by the altimeter, z_k, then also $[H] = 1$. Due to the lack of information regarding the stochastic characterization of the altimeter, the measurement noise r was adjusted to correspond to the sensor quantum. The process noise q, on its hand, was adjusted to improve the performance of the filter in terms of delay and outlier rejection.

A very important step of the Kalman Filter is the validation of the new measures, which can be performed by evaluating the covariance of the innovation, S_k. In fact, it is possible to define a parameter, γ, that will be defined as the threshold that indicates whether a new measure, z_k, is valid and should be incorporated or rejected.

$$\|z_k - Hx_k\| S_k^{-1} \leq \gamma \tag{3.4}$$

Under some assumptions this parameter γ can be selected according to a given confidence interval for a χ^2 distribution, but it can be also found heuristically. In here, the value for γ was empirically determined as the one offering the best trade-off between rejection of noisy/spurious measurements and output of the filter. The achieved results can be seen in Figure 3.3.

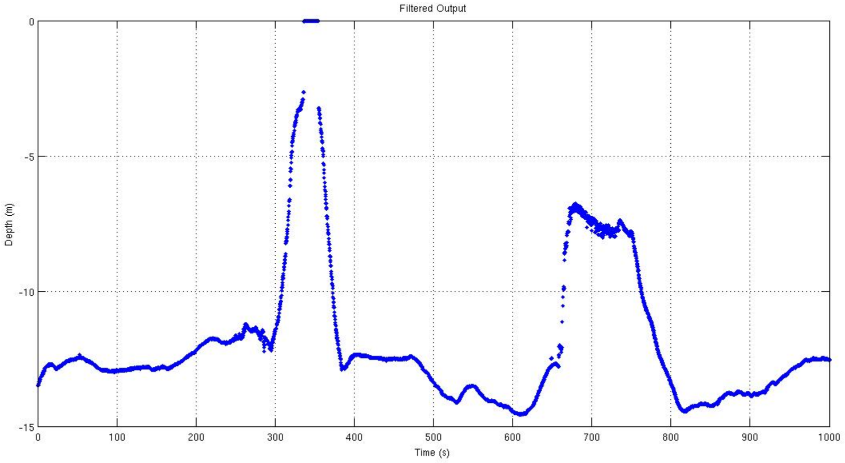


Figure 3.3: Filtered output of the altimeter; outliers were removed

3.4 Slope Estimation

The purpose of the work presented in this chapter is the development of an environment sensing based reactive behaviour, to be integrated on the on-board navigation software of the MARES AUV. This algorithm should be able to estimate the slope of the bottom, and to adjust the vehicle's pitch and depth accordingly. The estimation of the slope of the bottom is a two step process: first, the range measurements output by the altimeter need to be filtered, and with that data an estimate of the slope needs to computed. From the estimated slope, a time variable pitch and depth references are generated, always taking into account a safety distance to the bottom, preventing situations of bottom collision. Due to safety and general efficiency issues of the vehicle, the generated depth and pitch control variables are naturally bounded to limits considered reasonable.

The estimation of the slope of the seabed is an environment sensing based process. In this case, the environment sensing is made by the altimeter, which is mounted on the vehicle in a down-facing configuration, and provides range measurements to the bottom. Considering that the set of measurements has already been filtered out of outliers, they can be used directly without any major concerns. For the online estimation of the slope of the bottom, two different techniques were explored and compared: the first approach, described on Section 3.4.1, consists on the use of Linear Regression and the second approach, on Section 3.4.2, consists on extending the Kalman Filter used to filter the measurements of the altimeter, so that the slope of the bottom can also be estimated.

The slope, steepness or inclination of the bottom between any two points in space is usually defined as the rate of change of the bottom with the distance travelled in the horizontal direction. However, what is sought is to define the slope of the bottom as a function of the previous range measurements gathered throughout the time, as the altimeter provides measurements of the distance to the bottom over time. If this time series is plotted, what is obtained is a distorted map of the profile of the bottom along the direction of the vehicle's motion. However, the distortion is introduced because it accounts for the variations of the seabed profile over time, and not over travelled distance. To obtain a correct profile of the bottom, the surge velocity of the vehicle must be taken into account.

By differentiating the altimeter time series with respect to time, the rate of change of the distance to the bottom is obtained. To obtain an actual slope, this figure of merit needs to be compensated, accounting for the vehicle surge velocity, usually provided by the navigation layer of the on-board software. Nevertheless, this figure of merit, a "slope over time", by itself is enough to access the performance of the algorithms.

3.4.1 Linear Regression

The altimeter provides a continuous stream of ranges to the bottom, indicating how far from the bottom the vehicle is. As previously described, estimating the slope of the bottom can be accomplished by merely differentiating this time series with respect to time, and scale it afterwards according to the surge velocity of the vehicle.

The idea behind using a linear regression to estimate the slope of the bottom arises naturally: as differentiating a time-series can be tricky at times, being extremely sensitive to noise, the alternative is to find a curve that best fits to this time series. Having the analytical expression for this curve, it is then straightforward to calculate it's derivative, and from there inferring the "slope over time" of the bottom. From there, an estimate of the actual slope of the bottom can be obtained by just multiplying it by surge velocity of the vehicle.

The Linear Regression algorithm tries to find the polynomial, which order has to be defined in advance, that best fits a set of existing data points. Usually this fit is made in the least-squares sense. The polynomial that best fits the data can be written as in (3.5).

$$W^T X = w_0 + w_1 x + w_2 x^2 + \dots \tag{3.5}$$

It can be shown that W, the vector of coefficients of the polynomial, can be obtained by simple algebraic manipulation, as in (3.6), where X, the design matrix, is built using the input variables, and Y is the matrix of independent variables. To apply the linear regression to the specific problem of estimating the slope of the bottom, there are two main design choices to be made: the order of the approximating polynomial, and the size of data set to use.

$$W = (X^T X)^{-1} X^T Y \tag{3.6}$$

The order of polynomial is directly related to how the seabed is to be modelled. The assumptions about the bottom are that it should be smooth, without sudden variations of the profile, which should me less than one meter. Given this, the bottom could be modelled whether by a first or a second order model. Whilst a second order model might seem a good option, due to the fact that it is curvy and smoother than a linear one, it has a tendency to overfit snf adapt too closely to the set of data points, resulting in a poor performance, especially when in presence of a very noisy data set.

The second design choice, the size of the data set, has an important role on the overall performance of the fit: while increasing the size of the data set makes the fit smoother, on the other hand, it also increases the delay introduced and, therefore, the reaction time to significant changes in the slope of the bottom. The number of measurements to include in the regression must be a compromise between delay of the algorithm and performance.

3.4.2 Kalman Filter

The slope estimation described in the previous subsection, by means of a Linear Regression, is a two-step approach: it requires first a filter, to remove outliers from the measures, and then a Linear Regression estimator, that predicts the slope of the bottom. An alternative to this is to try to integrate both features onto the same algorithm, thus eliminating the extra delay introduced by the different estimators. Having in mind the good performance on removing outliers of the Kalman Filter, described in the previous section, and the bottom follower presented by Caccia et al. (1999), it was chosen to develop a Kalman Filter integrating both the outliers removal and the bottom estimation features.

A discrete dynamical systems for the bottom following and bottom estimator behaviour to implement can be represented by equations (3.7) and (3.8), respectively the process model and measurement model equations, both affected by random white noise, w_k and v_k.

$$\begin{bmatrix} d_{k+1} \\ \dot{d}_{k+1} \end{bmatrix} = \begin{bmatrix} 1 & \Delta_t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} d_k \\ \dot{d}_k \end{bmatrix} + \begin{bmatrix} 0 \\ w_k \end{bmatrix} \tag{3.7}$$

In the process model equation (3.7), the distance from the AUV to the bottom at time instant k is represented by d_k, and $\dot{d}_k$ is the derivative of this distance, the "slope over time" figure of merit.

$$z_k = \rho_k + v_k \tag{3.8}$$

At the same time, in the measurement update equation (3.8), ρ_k refers to the raw measurement of the altimeter at time instant k. The state of the system is continuously estimated by applying the usual Kalman Filter recursive prediction and update equations.

As usual, the performance of the filter can be tuned by adjusting the matrix Q and R, respectively the process noise and measurement noise. While the measurement covariance matrix was set proportional to the quantum of the altimeter, values for process noise covariance matrix were empirically set for the best performance having in consideration both the delay introduced, and the seabed tracking performance. Recalling from the Section 3.3, also here the validation of new measurements and, hence, outliers removal, can be performed by evaluating the innovation covariance.

3.5 Control Variables

The control layer of the MARES AUV is composed out of four different controllers, namely surge, heading, pitch and heave. For the bottom following behaviour, both heave and pitch need to be properly actuated. This section deals with the process of converting the estimated state variables, depth and slope of the bottom, into proper control references.

The heave controller is responsible for the controlling the depth of the vehicle and, therefore, the distance from which the vehicle is from the bottom. The depth reference for this controller, Z^*_{REF}, in (3.9), uses the estimated distance to the bottom, d_k, to generate the proper reference, but also compensates for the relative position between the location of the altimeter and the vehicle's center of gravity, x_{ALT}, and for the actual pitch of the vehicle θ. D_f is the parameter that sets bottom following distance - the distance to the bottom that the vehicle should always maintain.

$$Z^*_{REF} = -d_k + D_f + x_{ALT} sin(\theta) \tag{3.9}$$

To prevent situations of possible trap or loss of the vehicle, the reference sent to the controllers, Z_{REF} should be bounded, as in (3.10). By bounding the calculated reference, Z^*_{REF}, the vehicle is not allowed to follow sudden discontinuities in the slope, as this is not considered a safe behaviour, nor is under the assumptions according to which the bottom was modelled. Such bounding value will obviously vary according to the environment and to the missions being executed.

$$Z_{REF} = \begin{cases} Z^*_{REF} & \text{if } Z^*_{REF} < Z_{MAX} \\ Z_{MAX} & \text{otherwise} \end{cases} \tag{3.10}$$

As for the pitch controller, a suitable pitch reference must be derived from the previously estimated slope of the bottom, $\dot{d}_k$. The figure of merit "slope over time" has already been estimated and, by multiplying this value by the surge velocity, u, estimated on the navigation layer of the vehicle, the actual the slope of the bottom is obtained.

$$\theta^* = \theta + atan(\dot{d}_k u) \tag{3.11}$$

It is assumed that u accurately describes the velocity of the vehicle in surge direction, and therefore in (3.11) the effect of water currents might have on the direction of the vehicle is neglected. As $\dot{d}_k u$ is a slope, or a ratio, to get the equivalent angle it is enough to simply compute its arc tangent.

3.6 Results

In this section results for the proposed bottom estimation and bottom following will presented. These results were both simulated and experimentally validated, during several missions performed during the Summer 2012 in a location in the Douro river, close to Porto, in Portugal.

3.6.1 Simulated Results

The results presented in this section were obtained through simulation, using for that previously gathered data from the altimeter on an open-waters environment. Using this data allowed to recreate an actual profile of the bottom and compare and tune the different slope-estimation algorithms.

For this simulation tests, the main focus was on both the accuracy and the delay introduced while estimating the distance to the bottom. This estimate must be accurately estimated and evolve smoothly through time. Nevertheless, this online estimation step should not introduce a significant delay, otherwise the ability to avoid collisions with the ocean floor could be compromised. The simulations also allowed to find the best trade-off for a number of design parameters: for the Linear Regression algorithm it was possible to establish the most appropriate number of samples to use in the regression, while for the Kalman Filter the parameter γ and the covariance matrix values that best improved the results were determined.

The results obtained with the Linear Regression and with the Kalman Filter can both be seen, respectively, on Figures 3.4 and 3.5, an both of them correspond to the same altimeter data set, that simulates a given profile of the bottom. The plots contain points for the raw altimeter data, in blue, estimates of the depth of the vehicle, in red, and the estimated slope of the bottom, in green. It can be noted that the estimates for the slope of the bottom are smoother when estimated with the Kalman Filter. Also the Linear Regression estimated slope is more sensitive to small oscillations in the bottom, which in this case were caused by the normal overshoot in the heave controller. This sensitiveness, however, can lead to a behaviour which is not the desired one.

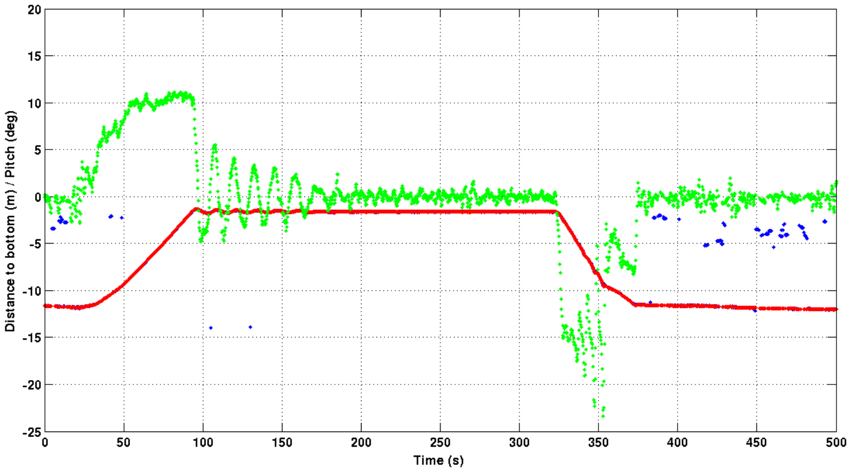


Figure 3.4: Simulation of the Linear Regression algorithm: output of the altimeter (blue), estimated distance to the bottom (red) and pitch of the vehicle (green)

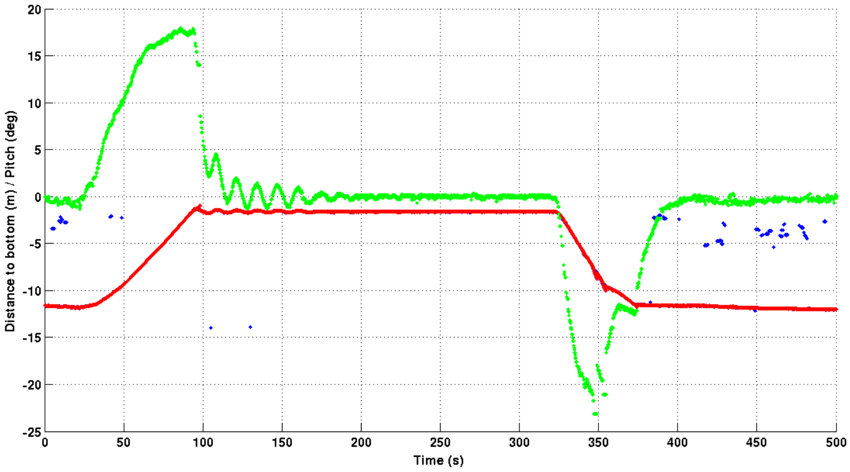


Figure 3.5: Simulation of the Kalman Filter algorithm: output of the altimeter (blue), estimated distance to the bottom (red) and pitch of the vehicle (green)

On both approaches, the points that correspond to the estimated distances to the bottom overlap almost entirely the points corresponding to altimeter measurements, with the exception of the removed outliers, which are the only points clearly visible. This demonstrates the good performance of two algorithms in terms of the introduced delay in the two estimators. Both approaches have a similar and negligible impact, introducing a very small delay. Figure 3.6 is as an example of it. There, it can be seen a detailed view of the steepest region of our simulated altimeter data produced by the Kalman Filter. Steep regions of the bottom are likely to be the ones where the delay would have a more visible effect but, as the plot demonstrates, this delay is small enough to be neglected.

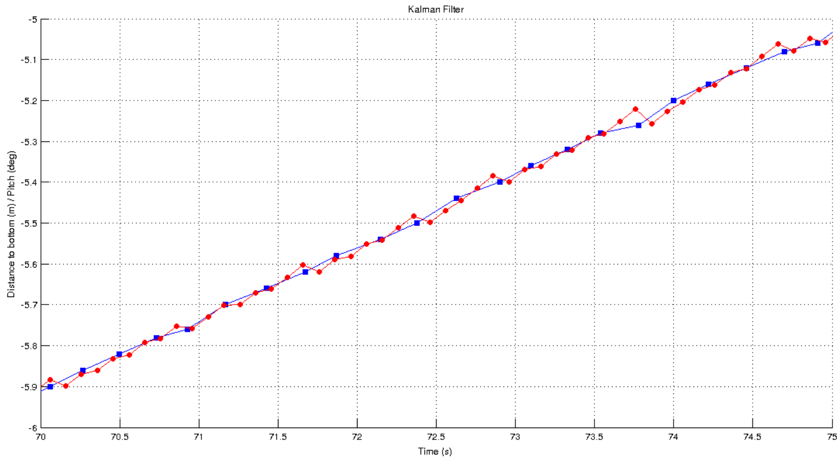


Figure 3.6: The delay introduced by both estimators is negligible: altimeter measurements (blue) and depth estimates (red) are almost overlapping in the steepest region of the simulation

These simulations results showed that the Kalman Filter has a better performance, as the Linear Regression estimator induces noisy estimates for the slope of the bottom. This is in fact a key factor for the choice of the Kalman Filter as the algorithm to be actually implemented. The final step of the simulation results consisted on integrating the Kalman Filter approach together with the vehicle simulator, in order to access the behaviour of the controllers and of the trajectory performed by the vehicle. Figures 3.7 and 3.8 depict, respectively, a simulation of the depth and pitch of the vehicle while doing a bottom following task on which the vehicle should follow the bottom with a distance of 2 meters.

It is clear from Figure 3.7 that the AUV follows the bottom at the desired 2 meters. The points in blue are relative to a simulated bottom profile, and the points in red are relative to depth data given from the navigation layer of the vehicle simulator. The distance between bottom and the vehicle is always around 2 meters, as desired, even when there are significant changes in depth of the bottom, as shown in the beginning and end of mission. The only exception to this occurs in the end of the mission, when the vehicle is only at 1.5 meters of the bottom, but already at surface level.

In Figure 3.8, there is a plot of the same profile of the bottom, in blue, against the pitch of the vehicle given from the navigation layer of the vehicle simulator. Again, the blue points are relative to the profile of the bottom, while the red ones are relative to the pitch of the vehicle. In this plot it is visible the change of pitch over time, in accordance to the profile of the bottom: when the depth of the bottom starts increasing the pitch of the AUV is negative, when the bottom is flat the pitch of the AUV is around zero, and the depth of the bottom starts decreasing the pitch of the AUV is positive.

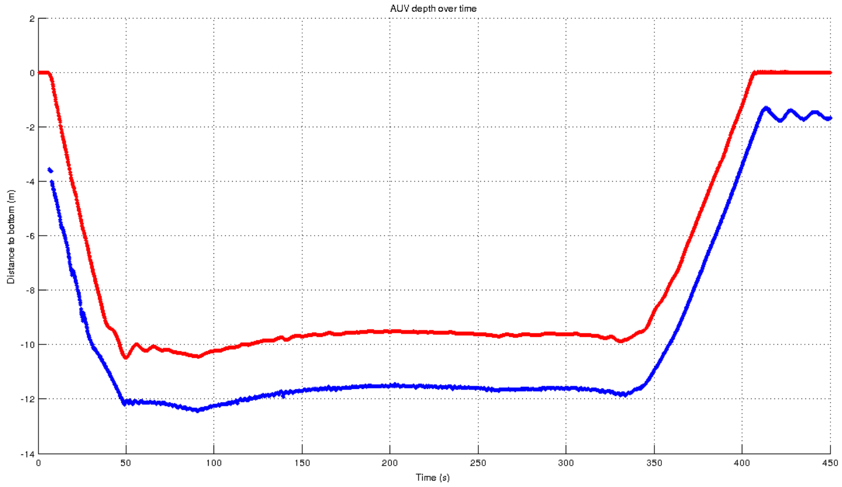


Figure 3.7: Simulation of a typical Bottom Following mission: profile of the bottom (blue) and AUV depth (red)

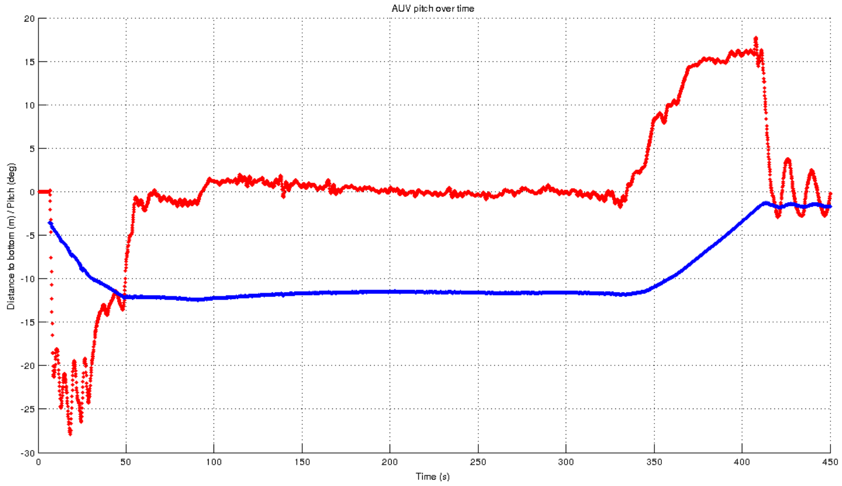


Figure 3.8: Simulation of a typical Bottom Following mission: profile of the bottom (blue) and AUV pitch (red)

3.6.2 Experimental Results

The experimental tests presented in this section are the result of a series of trials that were performed in the Summer of 2012 in the Douro river, in a location close by Porto, in Portugal. Simulations are never able to entirely model the dynamics and behaviour of the vehicle, and the complexity of an open-waters scenario. The challenge with these trials is to assess if the proposed algorithm is robust enough to be used in a real mission.

The AUV MARES was programmed to perform a number of different bottom following missions, and the plots depicting the behaviour of the vehicle during them can be seen in Figures 3.9 to 3.11. A typical mission consists on sending the AUV performing a straight line, maintaining a specified heading while controlling heave and pitch to do a bottom following manoeuvre, keeping a distance of 1.5 meters above the bottom. The selected missions for the plots here presented correspond to the data that more clearly demonstrates the performance of our approach in typical mission scenarios.

Figures 3.9 and 3.10 are complementary and correspond to the same bottom following mission. Figure 3.9 shows a plot of the depth of the vehicle, given by the navigation layer of the vehicle, and the estimated distance to the bottom. Figure 3.10 shows the same depth of the vehicle, plotted against the measured pitch of the vehicle. Initially, the vehicle is performing a hover manoeuvre, at 0.5 meters deep and then, after approximately 5 seconds, the bottom following manoeuvre is initiated. The AUV was initially on a location with very shallow waters, of less than 2 meters. As this is really close to the desired 1.5 meters, there were some oscillations on both depth and pitch. These oscillations can also be explained by the small overshoot that affects the heave controller. As the vehicle progresses, it can be seen that the measured distance to the bottom has a rough change, from around 1 meter to 2.5 meters. After this, the AUV starts behaving more smoothly, first steadily increasing it's depth for some seconds and, after second 60, decreasing again the depth. The distance to the bottom, however, clearly approaches the 1.5 meters bottom-following distance.

At the same time, the pitch of the vehicle changes accordingly to the evolution of the profile of the bottom, as can be seen on Figure 3.10. There, some oscillations on the pitch of the vehicle can be seen, while the vehicle is following some ascending or descending bottom profiles. These oscillations are quite small, usually less than 5 degrees, and they result from small changes in the topography bottom. However, if this behaviour is not desired, with the vehicle reacting to such small changes, this could likely be achieved by fine tuning the filter accordingly.

Figure 3.11 depicts a different bottom following mission, and represents the profile of the bottom and the trajectory followed by the AUV. This was done by combining data from the depth and distance to the bottom of the vehicle. The plot also contains data representing the pitch of the vehicle, which can be easily related with the steepness of the bottom. In this figure it is clear that the AUV trajectory is clearly following the bottom.

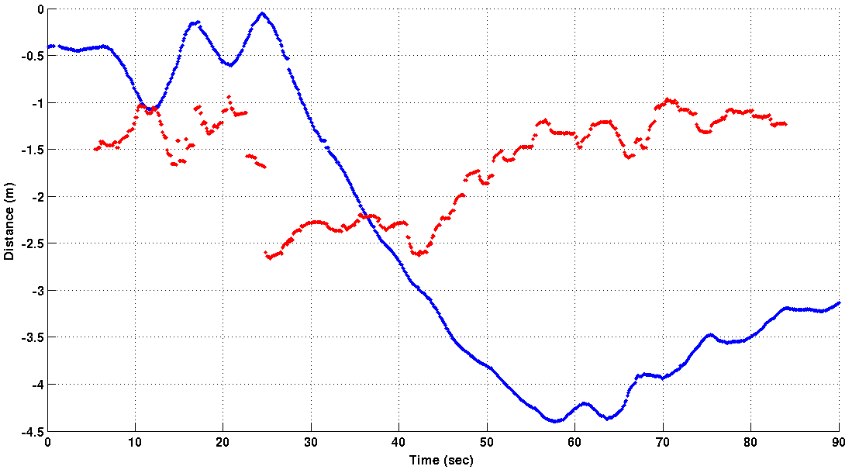


Figure 3.9: Bottom Following mission: depth of the vehicle (blue) and distance to the bottom (red) over time

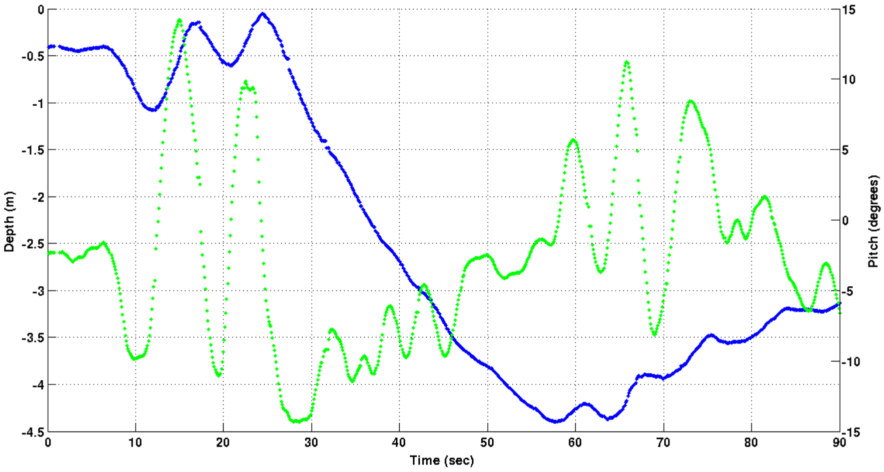


Figure 3.10: Bottom Following mission: depth (blue) and pitch of the vehicle (green) over time

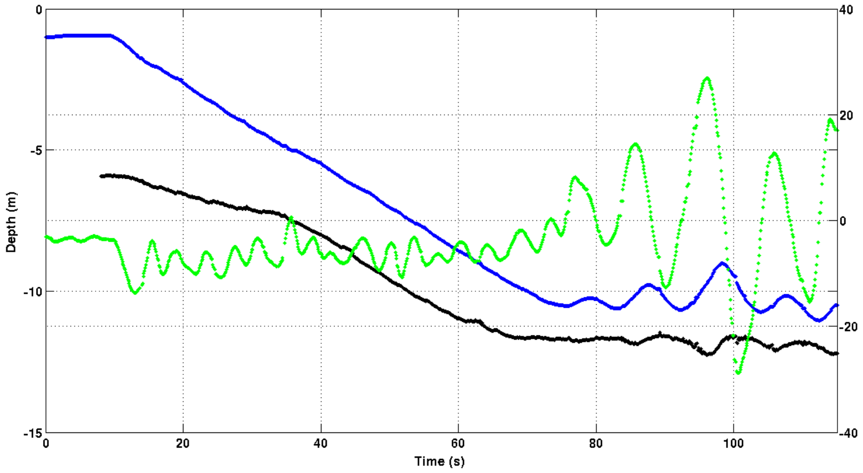


Figure 3.11: Bottom Following mission: bottom of the river (black), trajectory performed by the AUV (blue), and pitch of the vehicle(green)

3.7 Conclusion

When AUVs need to do tasks where the visual inspection of the bottom is required, they need to navigate as close to the bottom as possible, in order to maximize the quality of the final images. Moreover, they should maintain a parallel attitude towards the bottom in order to decrease the level of distortion of the images. Therefore, a bottom following behaviour, where the vehicle follows a trajectory always parallel to the bottom, is of critical importance. A Bottom Estimation and Bottom Following guidance-based approach was presented, that uses only an altimeter to continuously obtain ranges to the bottom of the seabed.

Two different estimation algorithms were initially proposed: one based on a Linear Regression, and one based on a Kalman Filter. Both the approaches consist on feeding the pitch and heave controllers of the vehicle with the desired control variables. These control variables are generated according to the real-time estimates of both the distance of the AUV to the bottom, and the slope of the bottom. After some simulation tests, it was concluded that the Kalman Filter performance to be more adequate to this problem. The subsequent integration of this Kalman Filter algorithm with the on-board software of the MARES AUV, allowed to experimentally verify the robustness of the solution in a real-world scenario. In the field tests, the AUV performed a trajectory closely resembling the profile of the bottom, something that can be very useful for missions requiring the visual inspection of the bottom. Nevertheless, in some situations small oscillations in pitch could observed. This behaviour is likely to be induced by the vertical controllers of the MARES AUV, that are not optimized to Bottom Following manoeuvres. This motivates the introduction of a dedicated controller for Bottom Following scenarios, presented in the next chapter.

Chapter 4

Bottom Following Controller

This chapter is, in a way, an extension of the previous one, as it also addresses the problem of bottom following by the MARES AUV. In the present, a suitable controller is derived for the Bottom Following problem, with particular requirements. The main requirements are the derivation of a single MIMO controller, for both depth and pitch, which should not present overshoot both in the depth and pitch outputs of the system. At the same, this controller should induce decoupling among to the remaining modes. The existence of such requirements motivates the use of Eigenstructure Assignment techniques, and a suitable controller is presented in this chapter.

4.1 Introduction

In the previous chapter a Bottom Estimation and Bottom Following guidance-based approach was developed. There, the focus was on developing a reactive bottom following behaviour based on environment sensing, which was able to provide depth and pitch references to the existing control layer of the MARES AUV. These references were derived by continuous real time estimation of the slope of the seabed using measurements provided by an altimeter. Experimental validation of the aforementioned work demonstrated the success of the approach, but lacked any guarantee in terms of overall stability of the system. Occasionally some oscillations were also observed that could degrade the performance of the approach.

On what follows those results will further be extended, by deriving a dedicated depth-pitch controller using Eigenstructure Assignment (EA) techniques. EA has been used and applied to many problems, covering a wide range of areas, from the control of aircraft (Faleiro, 1998) to unmanned underwater vehicles (White, 1998), among others. The motivation for the use of EA techniques are twofold. First, it is desired to have a simultaneous one step control of the pitch and depth degrees of freedom. Independent controllers, whereby the control of each degree of freedom is considered separately, makes no provision about the cross-coupling between AUV motion in the different planes. This might induce oscillations, like reported in the previous section. Second, it is desirable to have a controller that is compatible with the work presented in the previous chapter, but at the same time subject to strict time-domain requirements. For obvious reasons, it is desired that no overshoot or oscillations are observed on the pitch and depth transient response of the controller. Such controller should accept pitch and depth references as inputs, and be able to drive the AUV to perform trajectories that closely resemble the profile of the bottom.

This chapter is organized as follows. Section 4.2 introduces the general vehicle dynamics for AUVs, and particularizes the linearised diving equations for the MARES AUV. After that, section 4.3 presents Eigenstructure Assignment techniques, and Section 4.4 deals with the derivation of a suitable control law using full state feedback eigenstructure assignment. Simulation results that support the derived controller are presented in Section 4.5 and finally, in Section 4.6 provides some concluding remarks are presented.

4.2 Vehicle Dynamics

In this section the general dynamical model for AUVs is presented, and further detailed for the MARES AUV, which will be the object of the present chapter. The MARES AUV is a modular small sized AUV, which was already introduced in Section 2.6. Even though there are different configurations are possible, by adding or removing additional payload, the focus will be on the standard configuration of MARES as modelled by Ferreira et al. (2010b).

The notation and structure of the vehicle model used closely follows what was presented by Fossen (1994). Therefore, the state vector $\mu = [u, v, w, p, q, r]^T$ refers to the body-fixed vector of linear and angular velocities while $\eta = [x, y, z, \phi, \theta, \psi]^T$ is the absolute position vector. With that in mind, the general 6 DOF non-linear vectorial equation of motion of an underwater vehicle in a body-fixed reference frame can be written as:

$$M\dot{\nu} + C(\nu)\nu + D(\nu)\nu + g(\eta) = \tau \tag{4.1}$$

In (4.1), the matrix M refers to the inertia, $C(\nu)$ is the matrix with the Coriollis and centripetal terms, $D(\mu)$ contains the Hydrodynamical damping terms and $g(\tau)$ refers to the vector of restoring forces and moments.

$$M = M_{RB} + M_A \qquad C(\nu) = C_{RB}(\nu) + C_A(\nu) \tag{4.2}$$

The matrix M commonly encompasses both the rigid body dynamics and added mass terms, the same happening with $C(\nu)$, as indicated by (4.2). Finally, $g(\eta)$ refers to the restoring forces and moments acting on the vehicle, while τ describes the external forces and moments applied to the vehicle. Following the standard approach, no hydrodynamic interactions between the seabed and the vehicle are considered. The interested reader should refer Fossen (1994) for a more explicit and detailed derivation of this 6 DOF equation of motion.

Even though the model derivation for generic AUVs can be somehow intricate, some geometrical properties of MARES, depicted on Figure 2.2, alleviate that effort. Examples of this are the existence of different planes of symmetry and an appropriate choice of origin of the body vector frame. Another particularity of MARES is the use of 4 thrusters, whose forces and moments produced will compose τ. This thruster configuration provides peculiar ability to move with arbitrary low velocities, and even stop and hover in the water column, a characteristic that is of enormous interest when performing inspections operations. The complete analysis and modelling of the MARES AUV was covered by Ferreira et al. (2010b) and, for the sake of brevity, will not be further pursued.

4.2.1 Linearized Model

The general equations of motion for underwater vehicles, described in this section, are a set of differential non-linear equations (4.1). From those, a set of linear equations can be obtained by applying a Taylor series approximation around the equilibrium point $(\nu_0(t), \eta_0(t))$, and neglecting the terms with order two or higher. The following notation can then be introduced:

$$\Delta\nu(t) = \nu(t) - \nu_0(t) \qquad \Delta\eta(t) = \eta(t) - \eta_0(t) \tag{4.3}$$

The linearised model obtained by applying (4.3) to (4.1) assumes that the vehicle is moving on the longitudinal plane with non-zero surge and heave velocities, u_0 and w_0 respectively. Also, if the equilibrium point is characterized by roll and pitch angles equal to zero, and that the steady-state linear and angular velocities v_0, p_0, q_0 and r_0 are also equal to zero, it can be shown that a linear time-invariant equations of motion for an underwater vehicle can be described as (see Fossen (1994)):

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} -M^{-1}[C+D] & -M^{-1}G \\ J & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} M^{-1} \\ 0 \end{bmatrix} \tau \tag{4.4}$$

where $x_1 = \Delta\nu$, $x_2 = \Delta\eta$ and the matrices C, D, and G arise from the linearisation of $C(\nu)\nu$, $D(\nu)\nu$ and $G(\eta)$, respectively, around the equilibrium point. Further detailing (4.4), the vector of forces applied by the thrusters if given by

$$\tau = Pf_p = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 \\ z_{p1} & z_{p2} & -x_{p3} & -x_{p4} \\ -y_{p1} & -y_{p2} & 0 & 0 \end{bmatrix} \begin{bmatrix} F_{p1} \\ F_{p2} \\ F_{p3} \\ F_{p4} \end{bmatrix} \tag{4.5}$$

$p1$ and $p2$ refer to the horizontal left and right thrusters, respectively, while $p3$ and $p4$ refer to the vertical back and front thrusters. In (4.5) the matrix P is the propulsion matrix, assigning the contribution of each of the thrusters to the different state variables, and x_{p1}, x_{p2}, y_{p1}, y_{p2}, z_{p1} and z_{p2} refer to geometric properties of the location of the thrusters.

Note that (4.4) can now be expressed in the standard state-space form (4.6), which will be of particular usefulness for developments presented ahead in the paper.

$$\dot{\mathbf{x}} = A\mathbf{x} + B\tau. \tag{4.6}$$

4.2.2 Decoupled Pitch and Depth Control

The nature of the combined pitch and depth control being considered, together with the particular characteristics of the MARES AUV, allow for additional simplifications of the vehicle model to be used, namely by considering only state variables relevant for the motion on the longitudinal plane. That means that it will only be considered the state variables corresponding to both surge velocity u, heave velocity w, pitch rate q and pitch angle θ, and also the depth z. Thus, the following model can be used, which considers only the diving equations of motion:

$$\begin{bmatrix} \Delta\dot{u} \\ \Delta\dot{w} \\ \Delta\dot{q} \\ \Delta\dot{\theta} \\ \Delta\dot{z} \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} & 0 & 0 \\ a_{21} & a_{22} & a_{23} & a_{24} & 0 \\ a_{31} & a_{32} & a_{33} & a_{34} & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \Delta u \\ \Delta w \\ \Delta q \\ \Delta\theta \\ \Delta z \end{bmatrix} + \begin{bmatrix} \beta_1 \\ \beta_2 \\ \beta_3 \\ 0 \\ 0 \end{bmatrix} \tau \tag{4.7}$$

The diving motion model (4.7), introduced by Fossen (1994), was obtained by linearization around the point $[u_0, w_0, q_0, \theta_0, z_0]^T = [1,0,0,0,0]^T$. The term τ naturally referes to the forces produced by the thrusters. The terms of the matrix A, a_{ij}, are obtained from the corresponding terms of (4.4). For the sake of simplicity of notation, in what follows the Δ will be dropped when referring to any of the linearized state vectors.

4.3 Eigenstructure Assignment

Among the main advantages of EA techniques are the ability to deal with MIMO systems in a natural fashion, while at the same time using the available extra degrees of freedom to assign the eigenvectors in a way that mode decoupling can be achieved. Successful applications of EA have been mostly used to the control of aerial vehicles like (see Andry et al. (1983) for example), but little work as been done concerning autonomous underwater vehicles. This section introduces the general EA algorithm for output feedback control.

Consider the following multivariable linear system:

$$\begin{cases} \dot{x} = Ax + B\tau \\ y = Cx \end{cases} \tag{4.8}$$

where $x \in \mathbb{R}^n$, is the state vector, $\tau \in \mathbb{R}^r$ is the input vector, $y \in \mathbb{R}^m$ output vector and A, B and C are matrices of appropriate dimensions. It is possible to demonstrate that the system time response to an initial condition x_0 of the state vector, is given by

$$x(t) = Me^{\Lambda_i t}M^{-1}x_0 \tag{4.9}$$

where M is the matrix of eigenvectors of the system, $\Lambda = diag(e^{\lambda_1 t}, e^{\lambda_2 t}, \dots e^{\lambda_m})$ with λ_i the eigenvalues of the system, and x_0 the initial conditions. This is valid if and only if A is a diagonalizable matrix, for more details refer to Andry et al. (1983). Equation 4.9 demonstrates that the transient response of a system is dependent, apart from the initial conditions, on its eigenvalues but also on its eigenvectors. From (4.9) it is also possible to see that while the eigenvalues λ_i are mostly responsible for determining the decay rate of the response, the eigenvectors are responsible for the shape of the solution. It is now obvious that the transient response of a system depends most critically on its eigenstructure - set of eigenvalues and eigenvectors of the system. Thus, any results concerning changing the shape of the transient response of a system must change its eigenstructure.

Consider now applying a output feedback control law to the system(4.8) such that $\tau = Ky$. Under the influence of such control, the closed loop system becomes

$$x(\dot{t}) = (A + BKC)x(t). \tag{4.10}$$

For such system, EA is then reduced to the problem of finding the matrix K such that the eigenstructure of $[A + BKC]$ is as desired. Note that (4.10) doesn't make any restrictions on the number of inputs or outputs of the systems, thus making EA a technique suitable for the control of Multi-Input Multi-Output (MIMO) systems.

The eigenvalues of such system represent its poles and, therefore, are important for the stability and speed of the system. The eigenvectors, on the other hand, are linked with the shape of the system and are used to induce decoupling among the different modes. Recalling the definition of eigenvalues and eigenvectors, from (4.10) it is possible to write

$$(A + BKC)v_i = \lambda_i v_i. \tag{4.11}$$

or equivalently,

$$v_i = (\lambda_i I - A)^{-1}BKCv_i. \tag{4.12}$$

Equation 4.12 provides the only restriction to the nature of the choice of eigenvectors. That means that they should be chosen such that they lie in the subspace spanned by the columns of $L_i = (\lambda_i I - A)^{-1}B$. Fully specifying all the elements of all the eigenvectors is usually not possible due to the mathematical properties of the problem. According to Faleiro (1998), the number of specified elements in each eigenvector that can be decoupled is equal to $m-1$, where m is the number of inputs of the systems. Because of that, it is common to only specify the elements that are likely to induce the decoupling of the modes, thus leading to the construction of the vector v_i^d of the desired eigenvectors.

$$v_i^d = \begin{bmatrix} v_{i1} \; X \; X \; \ldots \; v_{ij} \; v_{in} \end{bmatrix}^T \tag{4.13}$$

In (4.13), v_{ij} are the specified components of the desired eigenvector, while X are the components of no interest. A common approach to induce decoupling is to set to zero the respective elements of v_i^d. In general a desired eigenvector, v_i^d doesn't lie on the subspace L_i. Therefore, an achievable eigenvector is obtained by projecting the specified elements of v_i^d onto an appropriate achievable subspace. The usual way to do this, as described by Andry et al. (1983), is by first applying a reorder operator $\{.\}^{R_i}$ such that

$$\{v_i^d\}^{R_i} = \begin{bmatrix} l_i \\ d_i \end{bmatrix} \tag{4.14}$$

where l_i are the specified elements of v_i^d and d_i the unspecified ones. The same operator should be applied to the subspace L_i so that

$$\{L_i\}^{R_i} = \begin{bmatrix} \tilde{L}_i \\ D_i \end{bmatrix}. \tag{4.15}$$

An achievable eigenvector, which on the general case differs from the desired one, can then be obtained as:

$$v_i^a = L_i z_i = L_i \tilde{L}_i^{\dagger} l_i \tag{4.16}$$

Obtaining all the desired achievable eigenvectors involves using (4.16) several times, on which the notation $(.)^{\dagger}$ refers to the Moore–Penrose pseudoinverse. Due to the involved matrix inversion operation, the numerical integrity of such operations can be at risk. Because of that, an alternative to (4.14 - 4.16) can be used using a constrained least square formulation.

4.3.1 Constrained Least Squares Formulation

By considering the case of a full output feedback where $Kv_i = u_i$, it is possible to rewrite (4.11) in a matricial form as

$$\begin{bmatrix} (\lambda_i I - A) | B \end{bmatrix} \begin{bmatrix} v_i \\ u_i \end{bmatrix} = 0 \tag{4.17}$$

What is sought is to find an achievable vector v_i^a such that (4.17) holds, while at the same time being as close as possible to the desired eigenvector v_i^d. In specific, the corresponding elements of

v_i^a should be as close as possible to the specified elements of v_i^d. Introducing the selection matrix S, this can be translated into the following:

$$\underset{x}{\text{minimize}} \; \left\| S v_i^a - l_i \right\|_2 \tag{4.18}$$

From (4.17 - 4.18), it becomes obvious that the EA problem can be transformed into a standard least square problem subject to equality constrains, in the form

$$\begin{aligned} \underset{x}{\text{minimize}} \quad & \|C\mathbf{x} - \mathbf{d}\|_2 \\ \text{subject to} \quad & L\mathbf{x} = \mathbf{j} \end{aligned} \tag{4.19}$$

that can be solved using any of the widely available optimization engines.

After solving (4.19), the gain matrix K from (4.10)can then be calculated straightforwardly as

$$K = UV^{-1}, \tag{4.20}$$

where $U = [u_1 \; \dots \; u_n]$ and $V = [v_1 \; \dots \; v_n]$.

4.4 Controller Design

This section will present the design of the desired depth-pitch controller for the MARES AUV. The derived control law should be able to follow depth and pitch references that are fed to the system.

Recall from Section 4.2 the expressions for the simplified 5 DOF diving equations of the MARES AUV. Gathering all the appropriate terms, the matrix A in (4.7) is:

$$A = \begin{bmatrix} -0.12 & 0 & 0.0006 & 0 & 0 \\ 0 & -1.5483 & 0 & -0.9782 & 0 \\ 0.0032 & 0 & -1.5858 & 0 & 0 \\ 0 & 0 & 1.0000 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \end{bmatrix} \tag{4.21}$$

By analysing (4.21), it is possible to verify that it presents five different modes, whose values are listed in Table 4.1. From there it can be seen that all the open-loop eigenvalues are real, but two of them are zero, and one is very small. In order to improve overall stability of the system, those eigenvalues should be brought further inside the left s-plane, but without demanding excessive control power. In what follows, it will be assumed that all state variables can be directly measured. These are in fact relatively mild assumptions for nowadays modern AUVs. This means that the output matrix C of the linearised system is simply an identity matrix.

In order to implement a tracking system, the standard approach is followed. This consists simply on introducing additional states in the state vector, one for each signal to be tracked. The

Table 4.1: Open loop eigenvalues

	λ_1	λ_2	λ_3	λ_4	λ_5
Value	-1.54	-1.59	-0.12	0	0

system state is then augmented with $z = [z_1\ z_2]^T$, so that

$$\dot{z} = r - Ex. \tag{4.22}$$

with $r = [r_1\ r_2]$ being the vector of references, and E being the selection matrix assigning the outputs which are required to follow the input vector r. The augmented system is now as in (4.23).

$$\begin{bmatrix} \dot{x} \\ \dot{z} \end{bmatrix} = \begin{bmatrix} A & 0 \\ -E & 0 \end{bmatrix} \begin{bmatrix} x \\ z \end{bmatrix} + \begin{bmatrix} B \\ 0 \end{bmatrix} u \tag{4.23}$$

Similarly to (4.10), the closed-loop system that results after applying a output feedback law $u = Ky$ is

$$\begin{bmatrix} \dot{x} \\ \dot{z} \end{bmatrix} = \begin{bmatrix} A + BK_1 & BK_2 \\ -E & 0 \end{bmatrix} \begin{bmatrix} x \\ z \end{bmatrix} + \begin{bmatrix} 0 \\ I \end{bmatrix} r. \tag{4.24}$$

It is interesting to note that by using EA, all the gains involved are calculated in a single step. For practical reasons, the gain $K_3 = K_1$ is also commonly used as anti windup strategy. A schematics depicting the structure of the controller derived can be seen in Figure 4.1.

The eigenvalues of the tracking system were specified using mostly empirical knowledge about the system. The desired and obtained eigenvalues can be compared in Table 4.2. As for the eigenvectors used for decoupling they were specified as indicated in Table 4.3.

Table 4.2: Closed-loop eigenvalues

	λ_1	λ_2	λ_3	λ_4	λ_5	λ_6	λ_7
Desired	-2	-5	-3	-2.5	-3.5	-1.5	-1.7
Obtained	-2.00	-4.58	-3.00	-2.50	-3.50	-1.50	-1.71

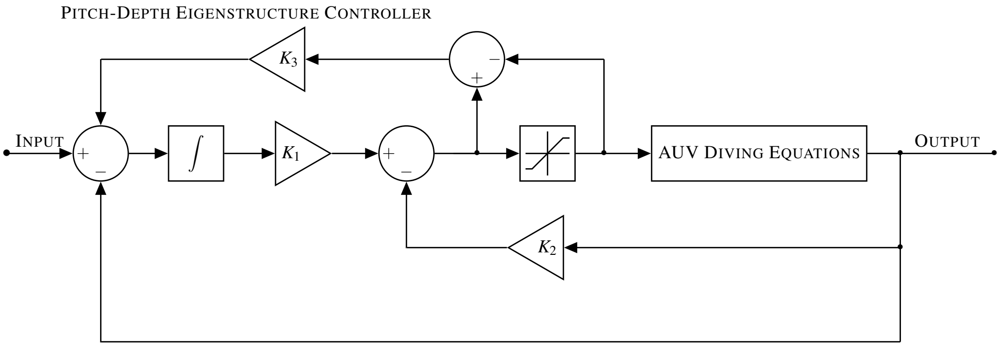


Figure 4.1: Structure of the implemented controller

	v_1	v_2	v_3	v_4	v_5	v_6	v_7
u	X	1	0	X	1	0	0
w	X	X	X	0	X	X	X
q	X	X	X	X	X	X	X
θ	0	X	X	1	X	0	X
z	X	0	X	X	X	X	0
r_θ	1	X	1	X	X	1	X
r_z	X	1	1	X	X	0	1

Table 4.3: Desired eigenvectors

4.5 Simulation Results

The overall performance of the controller derived in the previous sections was assessed in simulation environment. First, the system was tested when subject to a step reference input on depth only. The purpose of this test is to assess the coupling between states. The result of this can be seen in Fig. 4.2. Naturally, the reference in depth induces a variation not only in z but also in w, as expected. As for the other states, there are minimal transient variations, small enough to be neglected, suggesting that the decoupling was successful.

Next, it was simulated a constant slope bottom with an inclination of about 15 degrees. This was achieved with a negative ramp reference signal for depth, and a corresponding step reference signal for pitch. The results of it are depicted on Fig. 4.3 and Fig. 4.4. On the first one can see the evolution of all the relevant states along time, and it is possible to see that there is a change in w compatible with the variations in depth z, while at the same time variations in q match the changes in pitch, θ. Therefore, the results are very satisfactory. Most importantly, there is no overshoot neither in depth or pitch.

Figure 4.4 presents a detailed view on the error signal, that is the difference between reference and output of the system along the time. It is possible to see that while pitch is correctly tracked along time with zero error, there is a small steady state error in depth, of less than 0.25 meters. This is because the reference signal from time step $t = 5s$ till $t = 20$ is a ramp, and only one integrator

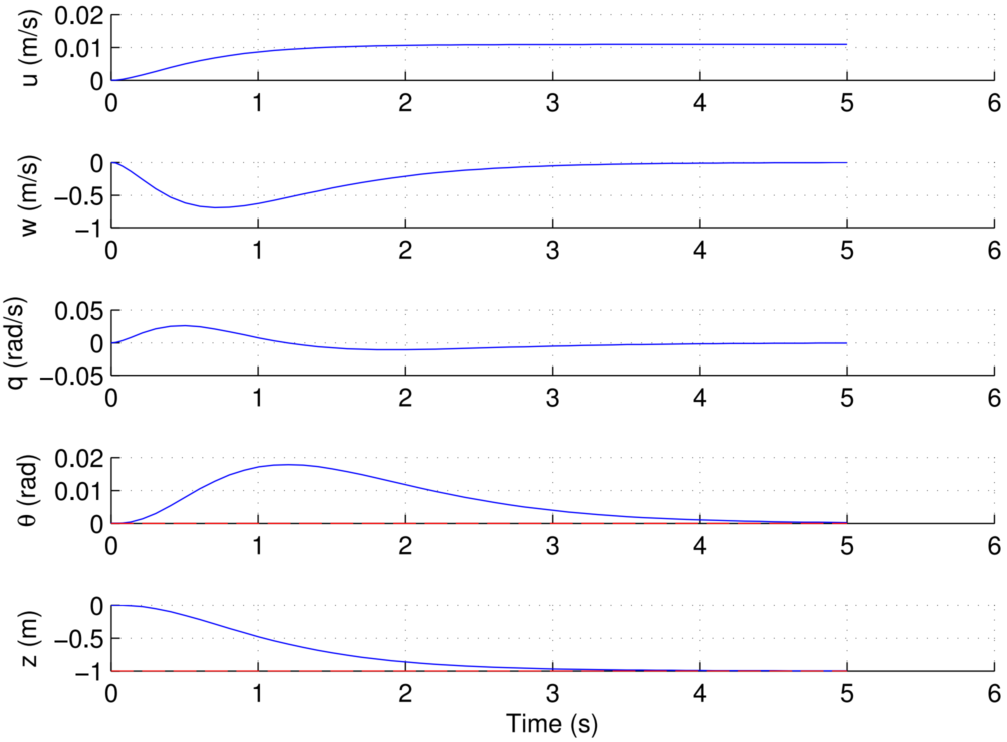


Figure 4.2: Evolution of the different state variables when the system is following a step depth reference signal

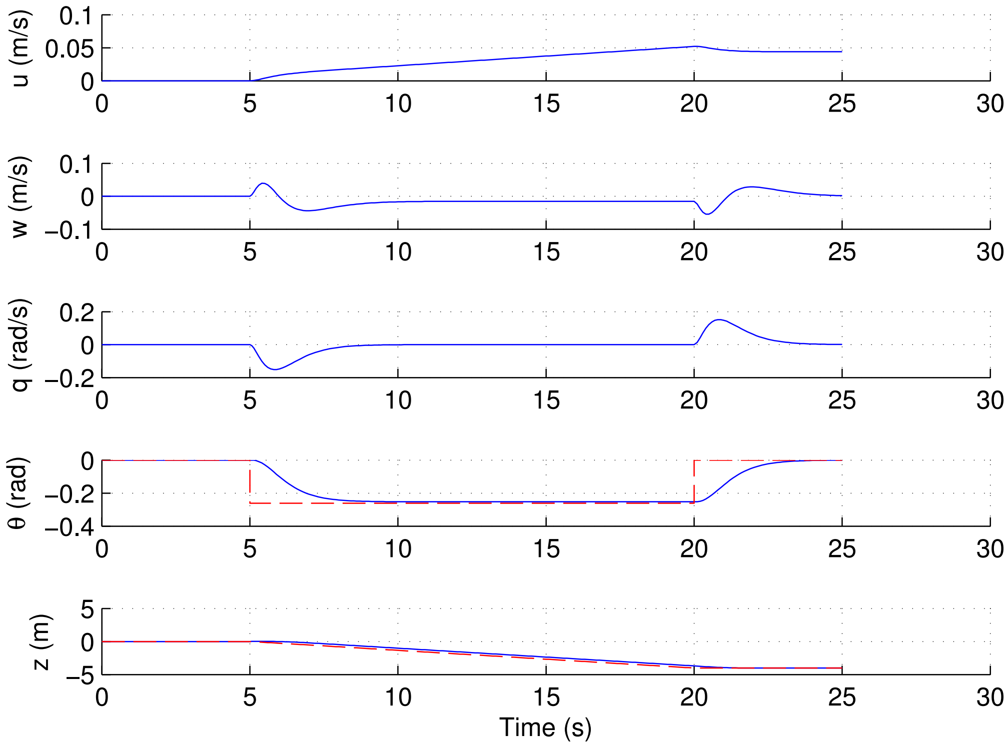


Figure 4.3: Evolution of the different state variables to dept-pitch reference. Solid line correspond to the output of the system, dashed lines correspond to the reference signals.

is included in the controller. Should this error be required to be zero, another integrator is needed.

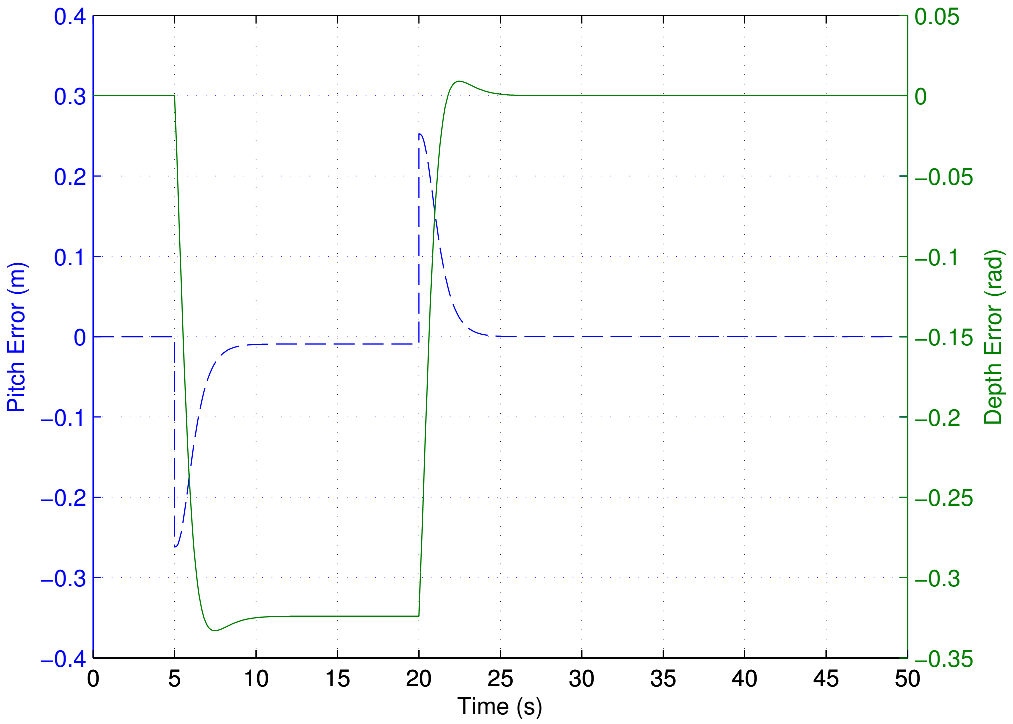


Figure 4.4: Transient error response of both depth and pitch following a reference signal

4.6 Conclusion and Future Work

This paper addressed the problem of designing a bottom following depth-pitch controller using Eigenstructure Assignment. Among the main advantages of this technique are the ability to deal with MIMO systems in a natural fashion, and the calculation of all the gains involved in a single step, without worrying on the effects cascading implementations may suffer. Moreover, EA allows to take into account restrictions related with the transient response of the system and, in specific, with the overshoot in the response to a desired reference signal. This of particular importance for the problem here addressed.

The simulation results clearly indicate that the controller derived has a decent performance for bottom following scenarios. Stability and zero tracking error of the closed-loop system were achieved in response to step inputs in both depth and pitch, and in both cases there is no overshoot observed, which was the main objective. Naturally, there exists a small steady-state error in response to ramps, but this can be tackled by introducing an additional integrator in the system, though care must be taken to ensure the overall stability of the system. Future research will address such issue, as well as implementing the controller on the on-board computer of the MARES AUV in order to assess the performance real-world scenarios.

Chapter 5

Terrain Based Navigation for sensor-limited AUVs

In this chapter the problem of Terrain Based Navigation for sensor-limited AUVs will be discussed, and in particular the Particle Filter based algorithms to address this problem. First, the TBN problem will be reviewed, together as an adequate problem formulation. Then, the Particle Filter framework is introduced and its application to TBN is discussed. Finally, a Data-Driven Particle Filter is derived, and results are presented demonstrating it can outperform other approaches proposed in the literature.

5.1 Introduction

Terrain Based Navigation (TBN), Terrain Relative Navigation (TRN) or Terrain Aided Navigation (TAN) are all interchangeable terms used to describe a family of position fixing systems based on the profile of the terrain. Generally speaking, the goal of Terrain Navigation techniques is to obtain a position estimate that best fits a set of acquired terrain measurements. This is done by comparing the obtained terrain profile with a prior map database.

The pioneering techniques for Terrain Based Navigation were initially derived to be used on missiles and aerial military vehicles. In an era when the ubiquity of GPS was still not a reality, TBN presented some unique characteristics that lead to its widespread acceptance. The autonomy and robustness against interference or jamming, together with the fact that such systems could be used under all weather conditions, during day or night, where among the main advantages of such algorithms. The ever increasing availability of high-resolution digital terrain maps, delivered with the aid of satellites, played a great contribution to the wide acceptance of this technique. Initial techniques eventually evolved into commercial products and nowadays several NATO aircraft are supposedly equipped with such systems. For a recent review on the different Terrain Based Navigation solutions for aerial vehicles, refer to works by KARABORK (2010), Vaman (2012) and the references therein. Besides providing accurate navigation solutions, modern terrain navigation systems have been extended for both safety and tactical benefits of the aircraft, offering features like predictive ground collision avoidance or obstacle warning, making it particularly suited to military vehicles. A recent description of a commercial TBN system for aircraft has been done by Cowie et al. (2008). More recently, Terrain Relative Navigation techniques have also been proposed to be used on several space exploration missions, namely for planetary entry, descent and landing, as for example by Alexander et al. (2012) and Johnson and Montgomery (2008).

Terrain Based Navigation has the potential to improve the autonomy of AUVs, allowing for long-term and long-range navigation of AUVs, given that appropriate terrain maps of the areas to navigate are available. Similarly to the use of GPS or Acoustic Navigation, TBN uses information of the variations of the terrain to bound the errors of inertial navigation, thus increasing the long term position estimation accuracy of a vehicle. Moreover, TBN is a self-contained technique, in the sense that no external aiding signals or devices are needed, which is in fact a great advantage.

5.2 Terrain Based Navigation for Underwater Vehicles

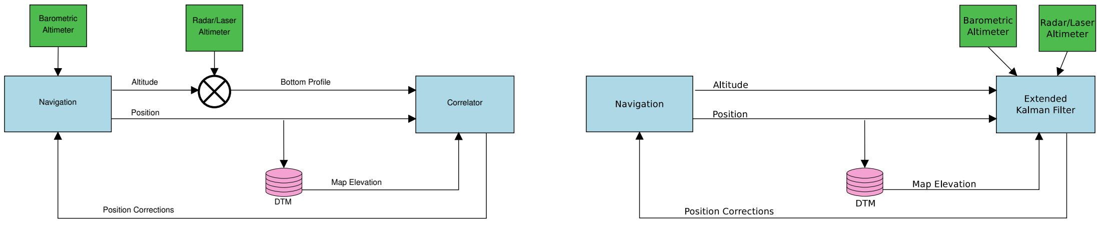
Terrain Based Navigation for underwater environments is fairly recent, at least when compared to aerial techniques. In this section a brief overview of Terrain Based Navigation for underwater vehicles will be provided, with a focus on the most prominent approaches.

Early classifications divided the existing TBN methods on two different broad categories: Batch Methods and Sequential Methods. The designation arose due to the nature of TERCOM and SITAN methods, arguably the most influential terrain navigation methods. Figure 5.1 schematically depicts the main differences between these two algorithms.

The Terrain Contour-Matching (TERCOM), the first Terrain Based Navigation method ever developed, dates back to the 1950's, when the concept was initially proposed. On its original formulation,TERCOM was designed to provide positional fixes that would update the INS of the vehicle. This positional fix was obtained by taking the best match from a correlation-like function that assesses how well the obtained terrain profile would match an existing map of the terrain. These profiles of the terrain were acquired by collecting a set of readings from the altimeter sensors, which are then processed in a batch, hence, the classification.

The Sandia Inertial Terrain Aided Navigation (SITAN), best known as SITAN, is the other fundamental terrain navigation method developed at the Sandia National Labs and first proposed by Hostetler (1978). The sequential nature of SITAN comes from the fact that every new terrain measurement is processed independently, and used as an input to the navigation filter. SITAN uses the obtained terrain measurements directly as an update of the navigation filter.

Regardless of that, Ånonsen (2010) introduced an alternative classification of underwater terrain navigation methods, dividing them between Search Area methods or Gradient-Based methods. This new designations are related with how terrain measurements are incorporated on the navigation filter of the vehicle. Loosely speaking, TERCOM tries to correlate the entire swath of measurements of the terrain with profiles of the map, using the best result of such match as a system measurement. Hence, in TERCOM a search is made on the entire map, or part of it, to find the position that best matches the entire terrain measurements swath. On the other hand, in



(a) (b)

Figure 5.1: Schematic view of the two instrumental Terrain Based Navigation methods: TERCOM (5.1a) and SITAN (5.1b)

SITAN like methods the multiple depth measurements provided by the sonar are used to compute local gradients and propagate the estimated position in the direction indicated by such gradients. Having this in mind, TERCOM could be classified as a Search Area method while SITAN would be a Gradient-Based methods. Indirectly, these new classifications are also a consequence of the ability to process large swaths of terrain in a single step, proportioned by the always increasing availability of computational power.

On an alternate perspective, terrain navigation algorithms can also be classified according to the degree of integration with the INS, namely between loosely and tightly coupled integrations. According to Hagen et al. (2011), in the tightly coupled approach the bathymetric measurements are used directly within the filter of the INS, along with all the other sensor measurements. On the other hand, in the loosely coupled approach, the bathymetric measurements are processed in a parallel filter until convergence, and the position estimate from terrain navigation is then fed back to the INS as a regular position measurement.

5.2.1 TBN for sensor-limited systems

Since the early days, the use of TBN for underwater vehicles has been highly focused towards the use of powerful multiple beam sonar, able to provide a high resolution perception of the environment. The experimental validation of such approaches was also consistently coupled with the use of high-grade INS. However, nowadays research efforts are being directed to the application of TBN for sensor limited systems.

As previously mentioned, sensor-limited systems refers to vehicles equipped with a combination of low accuracy inertial sensors, and simpler, low-information sonar like DVLs or altimeters. The use of DVLs or single beam sonars for TBN purposes has been reported by several authors, for example Morice et al. (2009); Donovan (2011). Since the larger group of AUVs is, by far, equipped with such low accuracy sensors, the rational of such approach is evident. The use of low accuracy inertial sensors motivates the use of a tightly-coupled integration between all the sensors, which allows the online estimation of critical sensor errors (Meduna et al., 2008).

Terrain Navigation is inherently a strong non-linear problem, mostly because of the non-linear nature of the terrain measurement function. As such, the interest on using non-parametric nonlinear Bayesian methods to address the problem of underwater terrain navigation is obvious. Following the initial theoretical work by Bergman (1999), the current trend seems to be on addressing the problem of underwater TBN using non-linear Bayesian estimators (Anonsen and Hallingstad, 2006; Anonsen and Hagen, 2011; Meduna, 2011; Teixeira et al., 2012). Even though TBN techniques have been demonstrated to perform well, in some situations the use of such non-linear filters can converge to incorrect solutions, especially if the vehicle is operating in particularly uninformative terrain. Filter divergence monitoring and recovering has also been under investigation (Murangira et al., 2011; Houts et al., 2012).

5.3 Problem Formulation

From the generic state-space representation of a dynamical system (2.5), and considering both process and measurement noise of additive nature, a basic discretized state-space model for an INS-based AUV system can be expressed by the following difference equations:

$$\mathbf{x}_{k+1} = f(\mathbf{x}_k, \mathbf{u}_k) + w_k \tag{5.1a}$$

$$\mathbf{z}_k = h(\mathbf{x}_k, \mathbf{u}_k) + v_k \tag{5.1b}$$

The difference equations (5.1) try to capture the fundamental idea of the general TBN approaches, on which vehicle terrain elevation data is used to correct the existing drift of Inertial Navigation Systems. The characteristics of such problem are described by both the state transition equation (5.1a) and measurement model equation (5.1b), that will be further described the following subsections.

In most of the early approaches, only motion on the North-East plane was assumed. This means that no uncertainty in any other directions can be observed and, therefore, there is no need to estimate them. While such assumptions are rather strict and difficult to observe, especially when in presence of sensor limited systems, they were introduced, up to a great extent, due to the limited available computational resources.

$$\mathbf{x}_k = \begin{bmatrix} x_{N,k} & x_{E,k} \end{bmatrix}^T \tag{5.2a}$$

$$\mathbf{u}_k = \begin{bmatrix} \Delta x_{N,k} & \Delta x_{E,k} \end{bmatrix}^T \tag{5.2b}$$

A problem with the state space in (5.2) is that uncertainty depth measurements might arise due to tidal levels variations. This can negatively affect the results obtained by TBN algorithms, particularly on missions with several hours of duration. To counter this, the time varying tidal level needs to be estimated. Donovan (2011) added to the state vector a depth bias term, related to tidal errors, in order to address the non-negligible effects of tides when determining the altitude of an AUV with respect to the sea bottom. A similar approach is also suggested by Anonsen and Hallingstad (2006). Alternatively, it was also suggested to work with relative profiles, but this option must be carefully considered as it is known to perform poorly in self-similar terrains.

With the increase of available computational power on-board the vehicles, different authors have proposed to increase the state space model to cope with different characteristics of the TBN problem. This has the purpose of trying to incorporate prior knowledge and take advantage of specific vehicle configurations. Another reason for the use of an augmented state vector is the use sensors with limited levels of accuracy, thus requiring the respective state variables to be estimated, as concluded by Anonsen et al. (2007). Meduna et al. (2010) introduced significant changes to traditional underwater TBN systems: the state vector is augmented to a dimension of eight, being composed not only by the vehicle position, but also by the vehicle attitude and angular rates. The introduction of these state variables is in part motivated by the tightly integration of multiple bathymetric measurements, and by the use of inertial sensors with poor accuracy.

5.3.1 Motion Model

Equation (5.1a) represents the state transition equation. The state vector, $\mathbf{x}_k$, is assumed to be Markovian, and the state variables usually represent the vehicle's two-dimensional horizontal position, at time instant k. These positions are frequently referenced to a north-east-down earth-fixed frame. $\mathbf{u}_k$ contains the position updates as calculated from the INS, and v_k represents the noise associated to the updates. In the literature, there is a preference for having the state space model, described by (5.1a), to follow simple linear models. This is what happens for example in Anonsen et al. (2007); Donovan (2012). Nevertheless, depending on the available on-board sensors and on-board computational power, other models can also be used, for example by performing a tight integration with the INS.

According to Donovan (2011), the stochastic motion model errors are dominated by the INS drift error, commonly characterized as an independent white Gaussian sequence. Some references also mentioned the use of the Singer model, a zero-mean stationary first-order Markov process. This is justified when it is necessary to capture the strongly correlated propagation of the inertial navigation system error (see Bergem (1993), Anonsen et al. (2007)).

5.3.2 Measurement Model

The measurement model equation (5.1b) compares the measurements from observation vector $\mathbf{z}_k$, to a non-linear function, $g(\mathbf{x}_k)$. For terrain navigation, it is considered that the measurements are obtained by bathymetric sonar sensors, therefore come in the form of ranges. At the same time, these measurements are compared with a bathymetric map of the area of operations. The measurement model equation can be further detailed as:

$$\mathbf{z}_k = \mathbb{M}(\mathbf{x}_k) - d_k + v_k. \tag{5.3}$$

In the equation above, $\mathbb{M}(\mathbf{x}_k)$ refers to the bathymetric map function and d_k to the vehicle's current depth. Because the vehicle is not navigating at the surface when acquiring the range measurements the depth of the vehicle must also be taken into account. The depth of the vehicle is customarily measured with relatively high precision pressure sensors. All these values are usually considered to be relative to the mean sea level (MSL). Analogously to the motion model, stochastic measurement model errors are described by independent white Gaussian sequence, v_k. However, the combination of different sensors can motivate the introduction of more detailed models, capturing the specific sources of error, as detailed by Hagen et al. (2011), namely AUV depth errors v_z, attitude and alignment errors v_a, sound speed errors v_c, map errors v_h or even sensor processing errors v_s.

On it's most commong form, the map function, $\mathbb{M}(\mathbf{x}_k)$, is usually implemented as a Digital Terrain Model (DTM). DTMs consist of gridded nodes, where each node corresponds to a depth at a specific location. The grid is usually equally spaced, and depth at inter-node locations are obtained through bilinear interpolation. Nygren (2005) claimed that in relatively flat areas using linear interpolation can be considered enough, but in more rough hilly terrains a more accurate method must be used. However, in a more recent work, Meduna (2011) stated that higher-order interpolation methods give small performance gains at the expense of large computational effort and that such higher order methods are more susceptible to overfit. Such situations can lead to an over-estimation of terrain variability and consequently degrade the performance of the system. Related to that, Ånonsen (2010) provided insight on problems that can arise due to map representation issues.

Going back to (5.3), the vector of observations consists on range measurements, as obtained by the bathymetric sensor. For single beam sensors, the observation vector consists of only a single range measurement, but for multiple beam sensors the observation vector will consist on the different r_i range measurements stacked in a vector, as in:

$$\mathbf{z}_k = \begin{bmatrix} r_1(\mathbf{x}_k) & \dots & r_i & \dots r_M(\mathbf{x}_k) \end{bmatrix}^T \tag{5.4}$$

When the vehicle has non-negligible roll and pitch angles, the range measurements provided by the sonar need to be compensated. This is of particular importance when the vehicle is navigating at high altitudes. The same happens when sonar sensors have multiple beams with a large swath. Most implementations of TBN use a projection-based scheme on which the measured range, r_i, is projected in the three-dimensional space. By doing so it is possible to compute the location p_{r_i} where a specific beam hit the bottom of the sea:

$$\begin{bmatrix} p_{N,r_i} \\ p_{E,r_i} \\ p_{D,r_i} \end{bmatrix} = R(q)a(r_i)r_i \tag{5.5}$$

In (5.5) $R(q)$ refers to the rotation matrix for the specific vehicle orientation q, and $a(r_i)$ refers to the unit direction vector of the beam corresponding to r_i. Combining (5.3-5.5), the following is obtained:

$$p_{D,r_i} = \mathbb{M}\left(\begin{bmatrix} x_{N,k} \\ x_{E,k} \end{bmatrix} + \begin{bmatrix} p_{N,r_i} \\ p_{E,r_i} \end{bmatrix} \right) - d_k + v_k \tag{5.6}$$

The measurement update equation compares each of the elements of measurement vector $\mathbf{z}_k$ with the bathymetric map of the area $\mathbb{M}(\mathbf{x}_k)$. This is done by evaluating the vertical distance between depth given by the map at the current location, and the depth obtained by the projection of the range obtained for a given beam, as indicated by (5.6). Figure 5.2 illustrates the projection-based measurement model. While the notation used here is adapted from Meduna (2011), similar approaches can be found elsewhere. According to the same author, range-based schemes which compare range measurements with predicted ranges could also be used. However, these are based on ray-tracing techniques, known to be computationally demanding, and thus incompatible with the limitied processing power available on-board the vehicles.

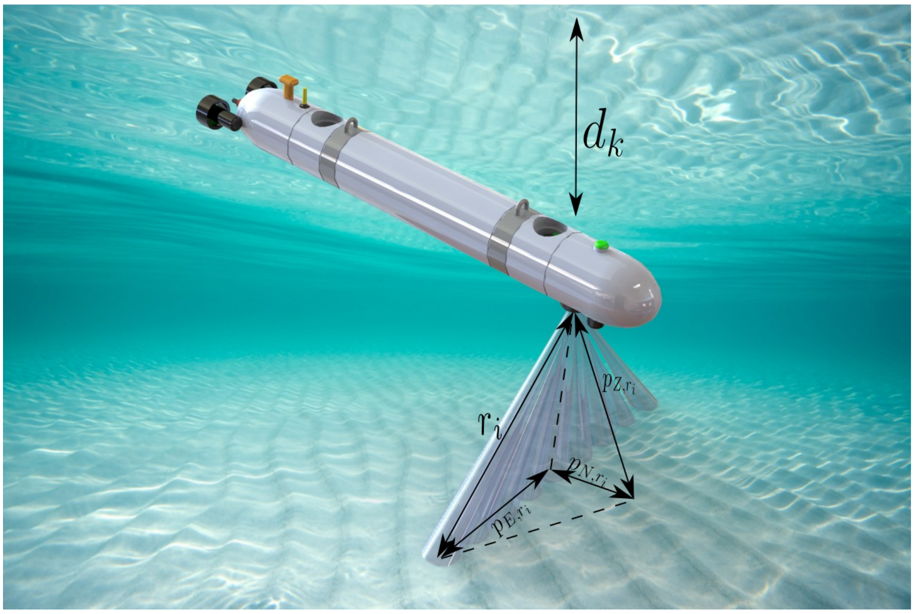


Figure 5.2: Schematic view for the terrain measurements for the multibem sonar case

For the sake of simplicity, lets rename the vertical the projection of the range from beam i, p_{D,r_i} as y_i, while h_i is the horizontal projection of the same beam, compensated by the vehicle's depth. Then, for a measurement vector of M total beams, affected by independent Gaussian noise, with covariance σ_e, the likelihood of a given measurement vector can be computed as:

$$p(\mathbf{z}_k|\mathbf{x}_k) = \alpha exp\left(-\frac{1}{\sigma_e^2}\sum_{i=1}^{M}\beta_i(y_i-h_i)^2\right). \tag{5.7}$$

The term β_i is sometimes used to affect each of the beams considered with different weights. This has can be useful particularly when the sonar in use has multiple beams. In such situations, it can be advantageous to weight the central beams more than outermost beams.

5.4 Particle Filters

Non-linear realizations of the Bayes Filters are now the primary choice for the design of TBN filters. Not only the bayesian approach allows for the fusion of information from multiple sensors, but it also provides a statistical measure of the uncertainty in the produced position. Bayesian parametric approaches, like the Kalman Filter, have been used in the early TBN techniques, for example with SITAN, or when the position output of correlation methods is used directly as a position update to the filter. Despite that, when the measurements are to be integrated with the main navigation algorithm, thus introducing non-linearities to the system, non-parametric approaches are more appropriate.

The strong non-linearities present in the measurement model of the system (5.1b), and introduced by the map-function $\mathbb{M}(\mathbf{x}_k)$, motivate the use of non-parametric filters like the Particle Filter or the Point-Mass Filter. This is mostly due to their ability to represent strong non-linearities more accurately. Moreover, these filters do not require the measurement model to be Gaussian, making them more appropriate in situations of multi-modality, or when only a few terrain height measurements are available. While both PF or PMF have been successfully demonstrated to handle the problem of terrain navigation, there has been a strong preference towards Particle Filters. In this section, some insight is provided on the theoretical background of the Particle Filters used, but a thorough derivation can be found for example in Gordon et al. (1993); Doucet et al. (2000); Arulampalam et al. (2002).

5.4.1 Monte Carlo methods

While closed-form solutions of the Bayes Filter, like the Kalman Filter, work well whenever linear and Gaussian assumptions apply, they are not appropriate for example when the distributions are non-linear, non-Gaussian or even multi-modal. In those cases, Monte-Carlo based methods can be a better alternative are lead to a superior estimation performance. Monte Carlo methods are a broad range of techniques that provide a numerical solutions for problems who don't have an easy analytical or closed form solution. They have been used in different domains, from physics to computational biology or even finance, most commonly as numerical integration method, for problems that otherwise would be intractable. When applied to the Bayes Filter, Monte Carlo methods can be used to compute the posterior distribution of the Bayes Filter (2.2). Particle Filter is the name generically given to a family of algorithms that uses Monte Carlo numerical methods to approximate the posterior distribution of the Bayes Filter. For that reason, Particle Filters are also known as Sequential Monte Carlo Methods.

Monte Carlo methods can approximate a probability distribution by sampling a large number of samples, that will be distributed according to the target density. By averaging the N samples, $\{x^{(1)}, \dots, x^{(N)}\}$, the quantity of interest can be obtained according to

$$p(x) = \frac{1}{N} \sum_{i=1}^{N} \delta(x - x^{i}). \tag{5.8}$$

with δ being the Dirac delta function.

5.4.2 Importance Sampling

Often it is not possible to sample from a given distribution of interest. When this is case, a technique called importance sampling can be used. The main idea behind importance sampling is to estimate the properties of a particular distribution, $p(x)$, by using samples from a similar distribution, $q(x)$, known as importance distribution or proposal distribution. The proposal distribution should be in some sense approximate to the original distribution, and its support should include the support of $p(x)$. Naturally, it should easier to sample from the proposal distribution.

The samples then need to be weighted in order to accurately represent the underlying distribution $p(x)$. Defining the importance weights $w(x)$ as

$$w(x) \propto \frac{p(x)}{q(x)} \tag{5.9}$$

then $p(x)$ can be approximated by N weighted samples as

$$p(x) \approx \sum_{i=1}^{N} w^{(i)} \delta(x - x^{(i)}) \tag{5.10}$$

The family of algorithms know as Particle Filters apply Monte Carlo methods and importance sampling to obtain estimates of the posterior distribution, $p(\mathbf{x}_k|\mathbf{z}_k)$. Following the equations of the recursive Bayes Filter (2.2-2.4), the idea is to use a sequential version of importance sampling, Sequential Importance Sampling (SIS) to obtain estimates of the posterior, $p(\mathbf{x}_k|\mathbf{z}_k)$. Because it is generally not possible to sample from the posterior, a proposal density will be used to sample the particles. The weights associated with each of the particles can be obtained recursively and using both the process and measurement model. Applying the Bayes rule to the posterior, $p(\mathbf{x}_k|\mathbf{z}_k)$, the following expression is obtained:

$$w_k^{(i)} = w_{k-1}^{(i)} \frac{p(\mathbf{x}_k^{(i)}|\mathbf{x}_{k-1}^{(i)})p(\mathbf{z}_k|\mathbf{x}_k^{(i)})}{q(\mathbf{x}_k^{(i)}|\mathbf{x}_{k-1}^{(i)})}. \tag{5.11}$$

Naturally, the choice of an appropriate importance distribution will determine the performance of the algorithm. An estimate of the posterior can then be obtained by averaging all the particles.

$$p(\mathbf{x}_k|\mathbf{z}_k) \approx \sum_{i=1}^{N} w_k^{(i)} \delta(\mathbf{x}_k - \mathbf{x}_k^{(i)}) \tag{5.12}$$

5.4.3 Resampling

The sequential update of the importance weights can lead to a situation where the majority of the particles have very small weights, while only a few dominate. Consequently, a lot of computational effort will be needed to propagate the low-weight particles, while their contribution is minimal. This phenomenon, known as particle degeneracy, is undesirable and can be minimised with a good choice of importance function, but also by introducing a resampling procedure on the particles.

The resampling procedure can be interpreted as a probabilistic implementation of the Darwinian idea of the survival of the fittest (Thrun et al., 2005). In the resampling procedure, from the original set of particles, $\mathbf{X_k}$, and weights, $\mathbf{W}_k$, a new set is generated which better approximates of the posterior. This is done by replacing particles with the smallest weights with a copy of the particles with the highest weights, thus increasing the concentration of particles in higher likelihood regions of the state-space.

Different resampling strategies have been proposed, like Multinomial resampling, Stratified resampling, Systematic resampling among others. While in most resampling strategies the number of samples is maintained constant, this does not necessarily always hold. In what follows the Minimum Variance Sample, or Systematic Resampling will be used. This algorithm presents a very low complexity, when compared to others. Additionally, from a uniform distribution perspective systematic resampling is theoretically superior (Hol et al., 2006). This algorithm uses a single random number to sample from and has a minimal complexity, making it the preferential resampling algorithm in numerous applications. The Systematic Resampling is detailed on Algorithm 5.1.

Algorithm 5.1 Systematic Resampling Algorithm

```
function SYSTEMATIC RESAMPLE($\mathbf{X}_k$, $\mathbf{W}_k$)
    $\overline{\mathbf{X}_k} = \emptyset$; $\overline{\mathbf{W}_k} = \emptyset$
    $r = rand(0; N^{-1})$
    $c = w_k^{(1)}$
    $i = 1$
    for $m = 1$ to $N$ do
        $U = r + (m-1)N^{-1}$
        while $U > c$ do
            $i = i + 1$
            $c = c + w_k^{(i)}$
        end while
        add $\mathbf{x}_k^{(i)}$ to $\overline{\mathbf{X}_k}$
        add $w_k^{(i)} = \frac{1}{N}$ to $\overline{\mathbf{W}_k}$
    end for
    return $\overline{\mathbf{X}_k}$, $\overline{\mathbf{W}_k}$
end function
```

Resampling is a very effective method to avoid the degeneracy of the particle filter set, or sample depletion, which happens whenever all but one of the importance weights are close to zero. However, resampling is also known to introduce the problem of loss of diversity among the particles, which in some cases can lead to "particle collapse", when all particles are just a copy of each other. This phenomenon, known as sample impoverishment, is particularly present when the noise of the dynamical system is low.

One obvious solution for stopping sample impoverishment is to add some random-noise to each particle before propagating it onto the next time step. This simple strategy, proposed by Fearnhead (1998), is known as jittering. However, and because the variance of the random noise introduced is freely determined by the user, this becomes a very *ad-hoc* solution. A similar, but slightly more elaborate technique was introduced by Musso et al. (2001), by proposing the Regularized Particle Filter (RPF). The problem of loss of diversity arises due to the fact that in the resampling stage, samples are drawn from a discrete distribution rather than a continuous one (Arulampalam et al., 2002). By sampling new particles from a continuous approximation of the posterior density $p(\mathbf{x}_k|\mathbf{z}_k)$, the RPF introduces the necessary diversity of the particle set. The continuous approximation of $p(\mathbf{x}_k|\mathbf{z}_k)$ is obtained according to (5.13), where $K(.)$ is a multivariate kernel estimator.

$$p(\mathbf{x}_k|\mathbf{z}_k) \approx \sum_{i=1}^{N} w_k^i K(\mathbf{x}_k - \mathbf{x}_k^i) \tag{5.13}$$

Using a kernel estimator to approximate the posterior requires minimal changes to the Particle Filter algorithm. According to what was stated above, the RPF only needs an additional step so that after resampling, some extra noise is added to the samples. Under certain conditions, when all the particles are equally weighted, it can be shown that the Epanechnikov kernel, in (5.14), is the optimal choice with respect to the mean integrated square error of the density estimation (Musso et al., 2001).

$$K(x) = \begin{cases} (2c_d)^{-1}(d+2)(1-x'x), & \text{if } x'x < 0. \\ 0, & \text{otherwise.} \end{cases} \tag{5.14}$$

Generating from the Epanechnikov Kernel consists on sampling new particles from a beta distribution uniformly distributed over a unit sphere of $\mathbb{R}^d$ (Devroye and Györfi, 1985). Thus, in (5.14) c_d refers to the volume of the unit sphere with dimension d. When using this kernel, after the resampling step new particles are then generated according to

$$\mathbf{x}_{k+1}^i = \mathbf{x}_{k+1}^i + HD_k\varepsilon \tag{5.15}$$

where H is the smooth parameter, or bandwidth, computed according a generalization of the Scott's rule of thumb for general multivariate distributions $H = n^{\frac{-1}{d+4}}$ (Ahamada et al., 2010). Additionally, D_k is the Cholesky factorization of the covariance matrix of the particles before resampling, and ε is generated from the Epanechnikov Kernel.

5.4.4 Generic Particle Filter

The Particle Filter is a numerical approximation to the recursive Bayes Filters that uses a weighted set of particle to approximate the posterior density, $p(\mathbf{x}_k|\mathbf{z}_k)$. Particle Filters borrow ideas from Monte Carlo methods, Importance sampling, and Resampling, in order to perform

Algorithm 5.2 Generic Particle Filter Algorithm

```
 1: Initialization
 2: for all $i$ do
 3:     $x_0^{(i)} \sim p(x_0)$
 4: end for
 5: for all $k$ do
 6:     for all i do
 7:         $\mathbf{x}_k^{(i)} \sim q(\mathbf{x}_k^{(i)}|\mathbf{x}_{k-1}^{(i)})$
 8:     end for
 9:     for all i do
10:         $w_k^{(i)} = w_{k-1}^{(i)} \frac{p(\mathbf{z}_k|\mathbf{x}_k^{(i)})p(\mathbf{x}^{(i)}|\mathbf{x}_{k-1}^{(i)})}{q(\mathbf{x}_k^{(i)}|\mathbf{x}_{k-1}^{(i)})}$
11:     end for
12:     for all i do
13:         $\tilde{w}_k^{(i)} = \frac{w_k^{(i)}}{\sum_{j=1}^{N} w_k^{(j)}}$
14:     end for
15:     RESAMPLE($\mathbf{X}_k$, $\mathbf{W}_k$)
16: end for
```

this approximation. The idea is to use a large number of particles sampled from an appropriate proposal distribution, whose weighted average will approximate the desired posterior.

While this representation is only approximate, it is guaranteed that such approximation will converge to the true solution as the number of sampled particles N increases. Moreover, this approximation can represent a much broader space of distributions than, for example, the Kalman Filter, that is restricted to Gaussian distributions only (Thrun et al., 2005). Notwithstanding, this realization of the Bayes Filter suffers for the curse of dimensionality, as its complexity increases exponentially with the dimension of the problem.

As a summary of this section, the pseudo-code for a generic Particle Filter is provided in Algorithm 5.2. This recursive algorithm can be informally described by the different stages, process update, measurement update and resampling. In the initialization, particles are initialized with according to some prior distribution, reflecting prior knowledge of the system. Then, lines 6 to 8 correspond to the prediction step, where new particles are sampled form the proposal distribution $q(:)$, and lines 9 to 11 correspond the update step, where the weights $w_k^{(i)}$ are computed. These weights and the sampled particles can be used to compute the prior. After the weight normalization, in lines 11 to 13, the particles are resampled, according to any of the resampling algorithms mentioned in the previous subsection.

5.5 SIR Particle Filter for TBN

The starting-point of this section is the Sequential Importance Resampling (SIR) Partile Filter, sometimes also referred to as Bayesian Bootstrap Filter. This SIR-PF is arguably one of the most widely used implementations of the PF, mostly due to it's simplicity and its performance, deemed adequate for most applications. This is also the case for underwater TBN, with a majority of the authors using it, even in applications to sensor-limited systems (Meduna et al., 2010; Ånonsen, 2010; Donovan, 2012; Hagen et al., 2011). In this section the SIR Particle Filter will be introduced, and its application to underwater TBN will be studied. The purpose of the simulation results presented in this section is to better understand the SIR-PF sensitivity to some of its parameters.

5.5.1 SIR-PF

The SIR-PF is perhaps the most common variation of Particle Filters. There are two main aspects that distinguish the SIR-PF from other variations, its choice of proposal distribution, and its adaptive strategy for resampling.

While it is known that $p(\mathbf{x}_k|\mathbf{x}_{k-1},\mathbf{z}_k)$ is the optimal proposal in terms of variance of the weights, this distribution is unknown and therefore needs to be approximated. In the SIR-PF, the dynamical model $p(\mathbf{x}_k|\mathbf{x}_{k-1})$ is used as the proposal distribution. Thus, there is no need to find an appropriate proposal distribution. This makes the implementation of the filter very simple, and sampling the particles is just sampling from the process model, $p(\mathbf{x}_k^{(i)}|\mathbf{x}_{k-1}^{(i)})$. It also simplifies the computation of the weights. Recalling (5.11), computing the weights now only requires the observation model:

$$w_k^{(i)} = w_{k-1}^{(i)} p(\mathbf{z}_k|\mathbf{x}_k^{(i)}). \tag{5.16}$$

The other aspect that differentiates the SIR-PF is the resampling step. While other variations of the Particle Filter might resample in every time step, SIR-PF adopts and adaptive resampling strategy, on which the effective number of particles, N_eff, is used for monitoring the depletion of the particle set, and the need for resampling:

$$N_{eff} = \frac{1}{\sum_{i=1}^{N}(\tilde{w}_k^{(i)})^2}, \tag{5.17}$$

where $\tilde{w}_k^{(i)}$ refers to the normalized weights of the particles.

For every time step, the SIR-PF starts with a prediction or time update step, where new particles are generated according to the proposal distribution, which in this case is the process model. Then there is a measurement update step, on which for every particle new weights are evaluated according to the observation model. After the weights have been normalized, the resample step takes place. For the SIR filter, resampling is only done whenever the effective sample size, N_{eff}, which is an indicator of particle set degree of depletion, is below a given threshold, N_{thr}, usually between $0.5N$ and $0.8N$. As there is usually no prior knowledge on the distribution of the particles once the algorithm is initiated, $p(x_0)$, it is a common practice to sample new particles from an uniform distribution covering the whole space. In Algorithm 5.3 the SIR Particle Filter is detailed.

5.5.2 Parameters Tuning

The main objective of navigation algorithms for sensor-limited AUVs is to bound the drift in the position estimates. Notwithstanding, heading estimates will also be of importance, mostly due

Algorithm 5.3 SIR Particle Filter Algorithm

```
Initialization
for all $i$ do
    $x_0^{(i)} \sim p(x_0)$
end for
for all $k$ do
    for all i do
        $\mathbf{x}_k^{(i)} \sim p(\mathbf{x}_k^{(i)}|\mathbf{x}_{k-1}^{(i)})$
    end for
    for all i do
        $w_k^{(i)} = w_{k-1}^{(i)} p(\mathbf{z}_k|\mathbf{x}_k^{(i)})$
    end for
    for all i do
        $\tilde{w}_k^{(i)} = \frac{w_k^{(i)}}{\sum_{j=1}^{N} w_k^{(j)}}$
    end for
    $N_{eff} = \frac{1}{\sum_{i=1}^{N} (w_k^{(i)})^2}$
    if $N_{eff} < N_{thr}$ then
        RESAMPLE($\mathbf{X}_k$, $\mathbf{W}_k$)
    end if
end for
```

to their role in propagating the positions throughout time. In the remainder of this section a number of simulations will be provided, aiming to study if a SIR-PF can track, besides the position, the correct heading of the vehicle. At the same time, it is also a goal of this section to study how the number of particles, the process noise and the sensor noise can affect the performance of the SIR-PF. Additionally, the influence of sampling from a continuous approximation of the posterior density, by using a Regularized Particle Filter, is also going to be studied.

The simulations will be based on a six-degrees of freedom model of an AUV, with measurements from both the IMU and the DVL artificially generated. By doing so it is expected to simulate a sensor-limited AUV. Using the orientation estimates of the IMU, the DVL output values are then transformed from the sensor frame of reference to the vehicle frame of reference; Finally the position of the vehicle is obtained by suitable integration of the velocity of the vehicle. The trajectory obtained in this way is going to be considered the ground truth for all the further calculations. However, in the real world it is obvious that noise always affects sensor readings in different ways. In this way, additive noise will also be introduced to the simulated sensor measurements. In specific, the noise of DVL velocities v_x^{DVL} and v_y^{DVL}, and the heading ψ are going to be affected by a first-order Markov Processes, while to the remaining sensor measurements were affected by Gaussian white noise.

The performance of the algorithms in terms of processing time is also of uttermost importance, as such filters should be running online, in the on-board computer of a sensor-limited AUV. Therefore, the first limitation imposed to our system will be on the time it takes for every iteration of the filter to process. This constitutes an indirect upper bound on the computational complexity involved. For this reason, simpler versions of the Particle Filter like the previously described SIR-PF, or the Regularized PF are more interesting.

5.5.2.1 Number of Particles

The number of particles of a Particle Filter has always been a design choice critical to the overall performance and convergence of the filter. It is empirically known that a large number of particles usually means better accuracy of the filter, and this is due to a more efficient covering of the whole sample space. This is particularly true when there is no prior information of the position of the vehicle and global localization is needed. However, when the filter is only tracking the vehicle, as in the case here considered, the number of particles needs decreases dramatically. By using the Kullback-Leibler metric, it was proved that filters with a small number of samples track a vehicle in a very satisfactory way (Fox, 2001).

In the simulations performed the number of particles was incremented from 250 to up to 6000 and the errors in position were compared. The upper bound on the number of particles, 6000, corresponds the the maximum number of particles able to be processed between the arrival of two consecutive measurements of the DVL, that in here was considered to occur every 5 seconds. The lower bound, on the other hand, corresponds to the minimum number of particles needed to have a consistent convergence of the filter to the true position.

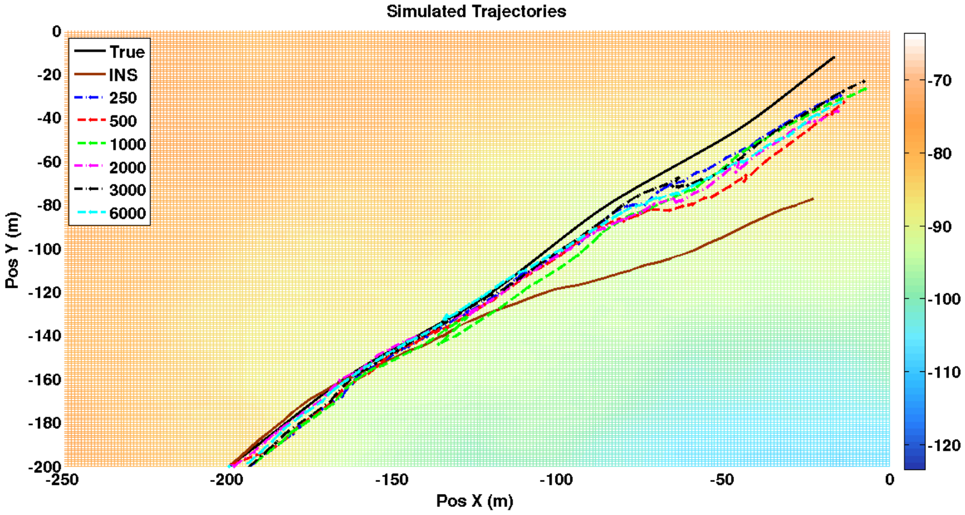


Figure 5.3: Simulated Trajectories with different numbers of particles. The filter clearly improves position estimates even when the number of particles is small.

In Figure 5.3 the trajectories for several simulations of a SIR PF are depicted, where only the number of particles was varied. It is clear that the filter provides better position estimates that traditional INS-based solutions. However, the filter is unable to closely track the true position of the vehicle.

The results obtained in terms RMS of distance to the true trajectory are depicted on Figure 5.4. As expected, the more accurate tracking of the true position happens when the filter is using an higher number of particles. However, comparing the final position of all the simulations, and the average RMS distance to the true trajectory, it is possible to verify that increasing the number of particles is not necessarily reflected on the performance of the filter. By comparing the average RMS of the error position between the filter using 3000 particles, and the filter using 6000 particles, the results are fairly similar. Considering that by choosing 3000 particles half of the processing power is being save, this is likely to present the best trade-off. In this specific situation, trading processing power for an higher number of particles is not a particularly attractive, as the increase in accuracy does not pay off. Consequently, and for the subsequent simulations, the number of particles was set to 3000.

As it can be seen on Figure 5.5, the states concerning the position of the vehicle, x^N, x^E and x^D can approximate fairly well the true states. It can be seen the filter is able to overcome the effect of noisy control inputs. However, the same doesn't happen for the roll, pitch and yaw of the vehicle, and the estimated values for these states present a non negligible deviation from the ground truth. This is especially noticeable for the heading of the vehicle, and this is likely to be the cause for growing position errors visible at the final part of the simulation.

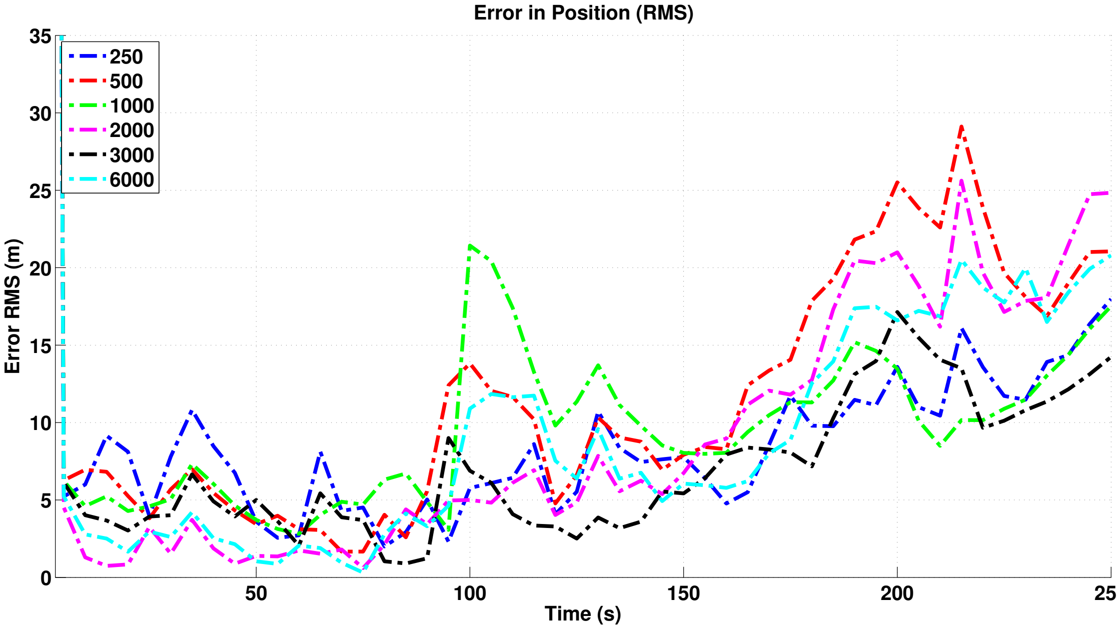


Figure 5.4: RMS Error in Position when comparing with the true position of the vehicle, for simulations with different number of particles.

5.5.2.2 Process Noise and Measurement Noise

The process noise and measurement noise are of paramount importance in the particle filter. In fact, if the process noise is small, or in extreme cases zero, the particle set will rapidly collapse (Salmond and Gordon, 2005). On the other hand, it is also known that increasing the noise levels is a common trick to improve the robustness of the filter. Increasing the noise level in the process model is supposed to increases the support of the sampled particles, while increasing the noise

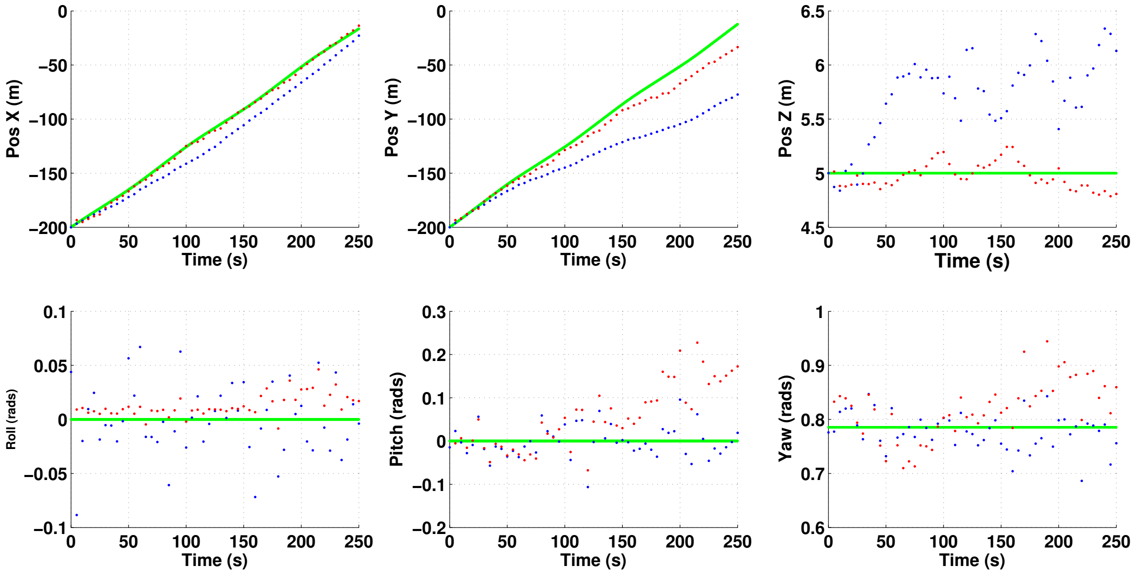


Figure 5.5: Evolution of the different state variables along the time. The true simulated state is depicted on green, traditional INS estimates are on blue, and the particle filter estimates are in red.

level in the observation model implies that the particles whose weights are smaller are more likely to be resampled (Gustafsson, 2010). To study the influence of both process and measurement noise on the performance, a different set of simulations were performed, while only changing this two sources of settings.

Increasing the measurement noise causes the particles with low weights to take more time to be replaced with newer particles. What this means is that when tracking a vehicle, the filter will have a slower response when trying to compensate for disturbances. By varying the measurement noise variations of the estimated variables were observed. However, it is likely this behaviour to be caused by the inherent stochastic nature of the filter. Furthermore, it is advisable to keep the measurement noise to levels similar to the ones introduced by the actual sensor.

Simulations confirmed that the level of process noise needs to be high so the tracking is successful, and this is depicted in Figure 5.6. This is particularly important if the noise that corrupts the sensor readings is significant. Naturally, simulations revealed that the filter is only able to asymptotically converge to the true position if the disturbances present were small. However, when the measurement noise is relatively high, then the process noise needs to be at higher levels too. However, an high level of process noise causes the cloud of sampled particles to spread, which can have negative effects. If the particles are spread over a too wide area in the space, any outliers could lead the filter to diverge from the true solution. At the same time, when the process noise is high, the estimated trajectory tends to be much less smooth as the estimated position tends to bounce around the true position.

At this point a closer look to the state variables concerning the attitude of the vehicle, namely roll, pitch and yaw is needed. While roll and pitch of the vehicle have little influence in the filter, as long as their value remains close to zero, the same doesn't happen with the heading estimate.

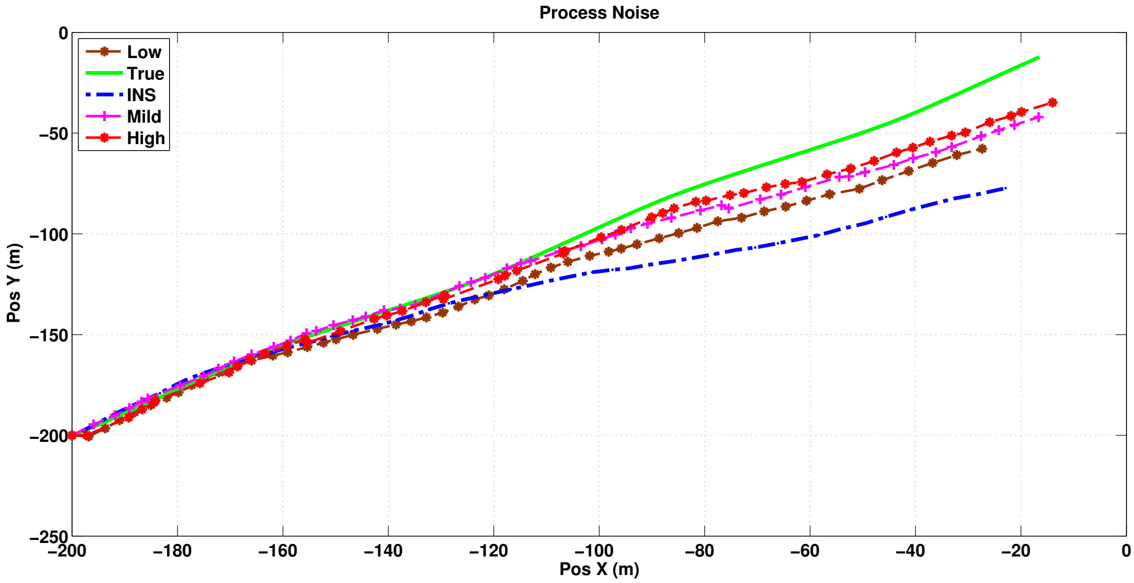


Figure 5.6: Effect of diferent levels of process noise in the filter, comparing with true trajectory and traditional INS-estimated trajectory

In fact, yaw is more difficult to measure and it seems to be more influenced by noise than the other two. This inability to accurately track the heading of the vehicle occurred throughout the simulations, even though a higher process noise seems to improve the results. If one is only interested in position corrections, the Particle Filter approach here proposed can be considered satisfying. However, if there is a need for an accurate estimate of the heading of the vehicle, other methods must be sought.

5.5.2.3 Regularized Particle Filter

As stated before, high levels of process noise are desirable, but if the process noise is too high this can cause some sort of degeneracy. To overcome this issue, the Regularized Particle Filter was tested. Ideally, the use of regularization would allow to reduce the process noise while maintaining a satisfactory estimation results, and therefore improve the heading estimation accuracy.

Despite the popularity of this kind of Particle Filter, the only improvements verified were obtained only when the number of particles was relatively small. Nevertheless, it was not possible to determine if this was caused by the regularization factor, or if it was in fact a consequence of the stochastic nature of the filter. By keeping the number of particles high, thus providing good estimation accuracy, the benefits of using such filter were negligible. This is presumably caused by the levels of noise present in the system, required to be high as previously mentioned.

5.6 Data-Driven Particle Filter

The previous sections presented the technique of Terrain Based Navigation and its application to underwater sensor-limited systems. Moreover, the theoretical grounds for the Particle Filter were also presented, together with the derivation of the SIR-PF, the most widely used implementation of the Particle Filter framework. Furthermore simulations provided an insight on the sensitivity of the SIR-PF to the process and measurement model, but also to the number of particles. In this section a new data-driven formulation of the Particle Filter will be proposed, that is more robust than the SIR-PF for TBN problems.

Recalling from the previous section, in the SIR-PF the proposal distribution is the process model. This means that the sampling density is independent of the most current measurement, therefore no knowledge from the most current observations is used when sampling for the particles. At the same time, if the system process model is in some sense distant from the true proposal distribution, the filter can become sensitive to outliers. If the process model does not accurately describe the system, and it is used as proposal density, this will typically influence the performance of the filter.

Sensor limited systems are, by definition, equipped with only a limited set of sensors, which characterized by their low accuracy. These sensors are commonly affected by different sources of noise, that negatively affect their measurements. Accurately modelling such system requires a thorough understanding of the sources of noise affecting it, which is not always easy or even possible. At the same time, due to the reduced amount of sensors being used, un-modelled disturbances can affect the system, without the possibility of measuring its effects. For example, if water currents are present, but they are not being modelled nor measured, they can have a influence in the position of the vehicles, though they are not taken into account.

In such situations, using the dynamical model of the system as proposal density leads to poor performance of the particle filter, and sometimes even to divergence of the filter. While increasing the process noise variance could somehow mitigate such effects, this is only effective if the un-modelled disturbances are mild. Nevertheless, this leads to a situation of excessive resampling, which is very inconvenient, as the resampling step adds up to the complexity of the filter. Even though resampling promotes the movement of particles from the prior areas to the high likelihood regions, performing it consecutively might also have some negative effects. If no jittering is performed, then it can cause a rapid loss of diversity in particles. At the same time, excessive resampling also causes the filter to be more sensitive to outliers. Recalling from general particle filter, after any resampling stage all the weights are set equal. Therefore, on the following time step the SIR-PF weight equation (5.16) can be simplified, with new weights being computed according to

$$w_k^{(i)} = p(\mathbf{z}_k|\mathbf{x}_k^{(i)}). \tag{5.18}$$

Therefore, after resampling the filter becomes insensitive to any prior knowledge of each of the particles historic. This is particularly serious if the likelihood is relatively narrow. From the explanation above, it is clear that obtaining a more accurate proposal density would be advantageous.

5.6.1 Related Work

It is known that in some situations the SIR-PF can fail, for example if new measurements appear at the tail of the prior of if the likelihood is relatively too peaked, when comparing to the prior. This could be solved by sampling from a better proposal distribution, one that would better match the true one. However, finding such distribution is no easy task.

In fact, the topic of generating better proposal densities has received some attention by researchers. First the HySIR (de Freitas et al., 2000) and later the Unscented Particle Filter (UPF) (van der Merwe et al., 2000) are two algorithms that have been designed to generate better proposal densities, thus addressing some of the shortcomings of the SIR-PF. The foundations of both these algorithms are similar, and are based on generating a proposal distribution by using, respectively, an Extended Kalman Filter and an Unscented Kalman Filter. In these algorithms the EKF and UKF equations are used in each particle, to sample and propagate individual approximations to the proposal distribution. Given that for each new particle a non-negligible number of matrix operations need to be performed, the additional computation requirements might not be negligible. Despite that, it has been theoretically demonstrated that particle filters with a proposal distribution obtained using the UKF outperform other existing filters.

An alternative for generating better proposal distributions, but with more modest computational requirements, has been proposed by Ju et al. (2010). Such approach uses a Kalman linear smoothing estimator for generating the proposal distribution. The main difference to the previous algorithms is that only a single proposal is generated for all the particles, being in that sense a more efficient approach.

In the context of Terrain Based Navigation, the effect of the proposal distribution in Particle Filters algorithms has also been studied. Teixeira et al. (2012) proposed the Mixture Particle Filter (MPF), and the Prior Particle Filter (PPF), both based on using a non-informative uniform distribution as proposal density. In the proposed MPF, the proposal density used is a mixture distribution, with a subset of particles being sampled from the usual process model, and another subset of the particles being drawn from a uniform distribution. Accordingly, these different proposals are also taken into consideration while computing the weights. In the PPF new particles are also sampled from the same mixture distribution. The difference to the MPF comes from how the the weights for each particle are obtained. Even though the particles are drawn from the different proposals, some of them from a uniform distribution and some from the process model, when computing the weights the proposal density is assumed to be the uniform distribution for all the particles. Simulation results demonstrated the superiority of the PPF in terms of the asymptotic convergence of the filters. In a later work (Teixeira et al., 2016), a suitable compact support for the uniform distributions is derived, using the Fisher information matrix of the terrain, a commonly used measure of the available terrain information. By using the aforementioned strategy, the authors noted that an adaptive number of uniform samples drawn at each iteration of the filter might be required, as the areas of the supports of the uniform distributions will be varying according to the terrain information.

5.6.2 Learning the Proposal Density

From what has been reported above, it is clear that finding a suitable proposal distribution, that is somehow better than the process model, is of paramount importance. This is particularly relevant for problems where the observation function is multimodal, like in underwater terrain navigation settings. However, obtaining a suitable proposal distribution is no easy task. Previous authors have used a non-informative, uniform distribution, with support selected according to the terrain information. Here, a new data-driven Particle Filter formulation is presented, that learns the proposal density from previous data. In what follows, only the horizontal position is going to be considered.

Up to the authors knowledge, only one previous data-driven particle filter has been reported. Wang and Chaib-draa (2012) proposed an adaptive nonparametric Particle Filter to solve the inverse kinematic problem of a two-link robot arm. The approach used Gaussian Processes to learn a suitable proposal density, that would later be used in a PF. While Gaussian Processes have been proved a valuable unsupervised machine learning tool, they require an offline learning process, which naturally requires a training data set. Thus, a similar approach would be difficult to implement in the context of underwater navigation.

The Data-Driven Particle Filter (DD-PF) here proposed borrows from other previous works that have been mentioned. While the goal is to learn a suitable proposal density, to be used in a particle-filter based terrain navigation algorithm, it is desirable that the computational requirements are kept relatively low. Thus, it would be interesting to learn a single proposal distribution, from which all the particles would be sampled from. The DD-PF can be implemented following the general particle filter algorithm, as listed in Algorithm 5.2, with the only difference that prior to any time step, the prior density needs to be learnt from previous data. Naturally, in the initial time steps the learning procedure can't be performed, and in this case the DD-PF falls back to the SIR-PF.

The main idea behind the proposed DD-PF algorithm is to approximate the learn proposal density by a Gaussian distribution. By using previous data, the historic of the previous positions of the vehicle, it is possible to predict a likely predicted position of the vehicle. This information can be valuable in generating a suitable proposal density. The proposal density, $q(\mathbf{x}_k|\mathbf{x}_{k-1},\mathbf{z}_k)$, will then be approximated as

$$q(\mathbf{x}_k|\mathbf{x}_{k-1},\mathbf{z}_k) \approx \mathcal{N}(\mu_D,\Sigma_D) \tag{5.19}$$

with $\mathcal{N}(\mu_D,\Sigma_D)$ being the Gaussian distribution with the mean and covariance matrix, respectively μ_D and Σ_D, extrapolated from the data.

Underwater vehicles, and in particular AUVs, have relatively slow dynamics, characterized by smooth motions, without any sudden changes of velocity. For that reason, they are often modelled with a constant velocity model. This knowledge about the vehicles will be used to learn μ_D, and Σ_D. By using a least squares regression, with the previous L estimates as an input, $\{\tilde{\mathbf{x}}_{k-L}, \tilde{\mathbf{x}}_{k-L+1}, \dots, \tilde{\mathbf{x}}_{k-1}\}$, a prediction of the current state of the vehicle, $\tilde{\mathbf{x}}_{k|k-1}$ can be obtained. While by doing this, the current observation is not explicitly used in the learning process, the contribution of previous observations is indirectly considered through the posterior of the previous time-steps.

Least Squares Regression

Lets consider for a moment the one-dimensional model of vehicle moving with a constant velocity. This can be modelled as

$$y_i = \beta_0 + \beta_1 x_i \tag{5.20}$$

where x_i, the input, corresponds to the i-th observation of the time, while y_i, the output, corresponds to the position of the vehicle. Naturally, the parameters β_0 and β_1 correspond to the initial position of the vehicle and it's velocity. By collecting a series of input and output observations, $\{x_i, y_i\}_{i=1}^{L}$, the parameters $\tilde{\beta} = [\beta_0\ \beta_1]^T$ can be determined using the closed-form expression for the Ordinary Least Squares (OLS) algorithm:

$$\tilde{\beta} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{Y} \tag{5.21}$$

where $\mathbf{X}$ is the $L \times 2$ matrix of the input observations, and $\mathbf{Y}$ is the $L \times 1$ matrix of the output observations. Extrapolating a new position of the vehicle, y_n, at time instant x_n can then by done by using the estimated parameters, $\tilde{\beta}$:

$$y_n = \begin{bmatrix} 1 & x_n \end{bmatrix} \tilde{\beta} \tag{5.22}$$

The extrapolated value, y_n, is only valid if x_n is sufficiently close from any of the regressors values, x_i.

The described OLS algorithm is adequate to estimate static parameters. However, when the parameters are time-varying, the use of a recursive version of the OLS algorithm, is more appropriate. The Recursive Least Squares (RLS) algorithm is, as the name indicates, a recursive solution for the least-square problem, where new estimates of the parameters are updated with new input and output observations. Doing so is computationally more efficient, as the computation of the term $(\mathbf{X}^T\mathbf{X})^{-1}$ is no longer required. At the same time, the use of a forgetting factor can be viewed as giving less weight to older data and more weight to recent data, thus promoting the estimation of slowly varying parameters (Vahidi et al., 2005). The RLS algorithm can be implemented with the following set of recursive equations:

$$\begin{array}{rcl} \beta_k & = & \beta_{k-1} + L_k(y_k - \phi_k^T \tilde{\beta}_{k-1}) \\ L_k & = & P_{k-1}\phi_k(\lambda + \phi_k P_{k-1}\phi_k)^{-1} \\ P_k & = & (I - L_k\phi_k^T)P_{k-1}\frac{1}{\lambda} \end{array} \tag{5.23}$$

Besides recursively obtaining an estimate for the time-varying parameters β_k, the RLS also keeps track of a gain matrix, L_k, and a covariance matrix, P_k. The latter will be of use when generating the proposal density. The forgetting factor, λ, can in a way be understood as a counterpart of the number of observations L in the original OLS algorithm.

Proposal Density

The proposal density used in our DD-PF will be approximated by a Gaussian distribution, with it's mean, μ_D, and covariance matrix, Σ_D obtained by a least square regression. For every time-step of the filter, a new proposal density needs to be estimated, thus being obvious the advantages of a recursive estimator.

Only the horizontal position is considered. Therefore, at time k, the mean $\mu_{D,k}$ will consists on the predict positions of the vehicle, $\tilde{\mathbf{x}}_{k|k-1}$, where each of the element of $\mu_{D,k}$ being independently predicted with an appropriate regression. The inputs for each regression are the time instants and positions estimated by the filter in the previous time instant, as was made clear before. Accordingly, $\Sigma_{D,k}$ is a diagonal matrix, with the elements in the diagonal being proportional to L_k of the corresponding regression.

5.7 Simulated Examples

In this section the feasibility of the proposed Data-Driven Particle Filter is analysed. Using a series of simulation studies, the performance of the filter is assessed, demonstrating that the DD-PF can be advantageous when compared to traditional SIR-PF. Not only the SIR-PF is the most widely used implementation of the PF framework, but also it has been often used in the literature concerning TBN for sensor limited systems. The goal for the simulations is to demonstrate the superior performance of the DD-PF for sensor limited systems, when subject to un-modelled disturbances. Such disturbances can be caused by lack of proper knowledge of the system, for example using sensors with unmodelled error sources, or by external sources, for example when the vehicle is subject to currents that can not be measured. The simulations will compare the results obtained between the DD-PF and the SIR-PF.

The simulations presented only consider the vehicle motion in the horizontal plane. The major concern will obviously be the asymptotic convergence of the filter, with the obtained results for each simulated scenario reflecting the ensemble of 100 independent runs of the filters. In order to facilitate the comparison between different filters, the time-indexed Root Mean Square (RMS) error metric for a two-dimension position, defined as follows, will be used.

$$RMS_k = \sqrt{\frac{1}{M}\sum_{r=1}^{M}(\hat{x}_k - x_{k,r})^2 + (\hat{y}_k - y_{k,r})^2} \tag{5.24}$$

(x_k, y_k) refers the correct position of the vehicle at time instant k, while $\left(\hat{x}_{k,r}, \hat{y}_{k,r}\right)$ refers to estimated position at time instant k, obtained in the r-th MC run. By using this metrics, the obtained RMS position error is averaged for all the M runs performed, thus obtaining the MC ensemble average of the RMS error. The computational complexity of the filter is also of interest, as the available computational processing power is often limited. Therefore, results attesting the lower complexity of the proposed approach are also going to be presented.

Parameter	Value
Filter Settings	
Number of Particles (N)	100
Process Noise (σ_v)	$\sqrt{5}$
Measurement Noise (σ_w)	1
Resampling Threshold (N_{thr})	$0.6N$
Initial Position variance (σ_{p_0})	$\sqrt{5}$
Sensor Settings	
Number of beams (M)	4
Sensor Noise (σ_s)	0.2
Proposal Density Learning Settings	
Forgetting Factor (λ)	0.9

Table 5.1: Parameters of the Data-Driven PF

The DD-PF was envisioned having in mind improving the performance of TBN algorithms for sensor limited systems that are based on PFs. Therefore, the simulations presented will try to demonstrate that. The main parameters for both the SIR Particle Filter and the Data-Driven Particle Filter used throughout the simulations are listed in Table 5.1.

The trajectory of the AUV was simulated as follows. First, the velocities of the vehicle were generated in the vehicle's reference frame. Accordingly, it was considered the vehicle to have only surge velocity (u), constant and with a value of $1ms^{-1}$. No contributions in the sway (v) or heave (w) were considered. Following to that the attitude of the vehicle was generated, with the roll (ϕ) and pitch (θ) of the vehicle being considered zero. For the the simulated linear trajectories the heading (ψ) of the vehicle was considered to be always constant, while for the circular trajectories it was considered the heading to be varying at a constant rate. These quantities can be understood as inertial sensor readings, and will be used to generate the position of the vehicle.

With $C_b^n(\phi, \theta, \psi)$ being the rotation matrix, from the body frame to the navigation frame, and considering that

$$\begin{bmatrix} \Delta x_k \\ \Delta y_k \end{bmatrix} = \begin{bmatrix} \Delta_k & 0 & 0 \\ 0 & \Delta_k & 0 \end{bmatrix} C_b^n(\phi, \theta, \psi) \begin{bmatrix} u \\ v \\ w \end{bmatrix}, \tag{5.25}$$

than the position of the vehicle can then be recursively updated as follows:

$$\begin{bmatrix} x_k \\ y_k \end{bmatrix} = \begin{bmatrix} x_{k-1} \\ y_{k-1} \end{bmatrix} + \begin{bmatrix} \Delta x_k \\ \Delta y_k \end{bmatrix} \tag{5.26}$$

Because only the horizontal position is considered, the selection matrix is used to disregard any positional updates in the z direction. This set of data, positions, velocities and orientations will be considered as ground truth of the simulations. For the sake of simplicity it was chosen to don't affect any of the sensor readings with noise of any kind. At the same time, in order to simulate the disturbances affecting the system, another set of data was prepared, but with the influence of constant disturbances, $D = \begin{bmatrix} d_x \, d_y \end{bmatrix}^T$, such that

$$\begin{bmatrix} \tilde{x}_k \\ \tilde{y}_k \end{bmatrix} = \begin{bmatrix} \tilde{x}_{k-1} \\ \tilde{y}_{k-1} \end{bmatrix} + \begin{bmatrix} \Delta x_k \\ \Delta y_k \end{bmatrix} + \begin{bmatrix} d_x \\ d_y \end{bmatrix} \tag{5.27}$$

This last data set, with the influence of the disturbances, is the one who is going to be used to generate the sonar sensor measurements. The initial positions for both the data sets is the same, but some uncertainty for the initial position is considered, with a variance similar to what GPS receivers can provide. Two sets of simulations were prepared, the first one concerning a linear trajectory of the vehicle, and the second one concerning circular trajectories.

The sonar sensor measurements are generated at the real positions of the vehicle, under the influence of the disturbances, $\tilde{\mathbf{x}}_k$. These measurements are generated at a rate coincident with the time step chosen for our filter. The simulated sonar consisted on a four beam sonar sensor, in a Janus configuration, similar to a DVL. For that purpose, a synthetically generated bathymetric map was used, from which sonar sensor measurements are simulated. The ranges returned from each of the beam were them corrupted with Gaussian noise.

Linear Trajectories

The first set of simulations were performed in order to evaluate the performance of the DD-PF for a vehicle moving on a linear trajectory, but subject to external disturbances. Considering the vehicle surge velocity to be of $1ms^{-1}$, the disturbances were set up to be of $0.3ms^{-1}$ in both X and Y directions. The trajectory performed by the vehicle can be seen in Figure 5.7.

Figure 5.7 depicts the desired trajectory, in green, and the real trajectory with the effect of the disturbances, in black. Additionally, these trajectories are overlaid on the contour levels of the topography of the bottom. It can be seen that while the desired trajectory, in green, is only of 250 meters, the absolute error in position of the vehicle when subject to disturbances is roughly of 100 metres. This is equivalent to a disturbance of around 40% the distance travelled (DT) by the vehicle, an already quite significant disturbance, and in line with some sensor-limited AUVs available.

Figure 5.8 compares the output of both the SIR-PF and the DD-PF for the scenario that was just described, showing the evolution of the positions along the time. Again, the black and green lines correspond, respectively, to the real and desired trajectories. The output of the SIR-PF is shown in blue, while the output of the DD-PF is shown in red. It can be seen that the outputs of both filters are able to follow the real trajectory. However, it seems that the DD-PF provides more smooth and accurate trajectories than the SIR-PF.

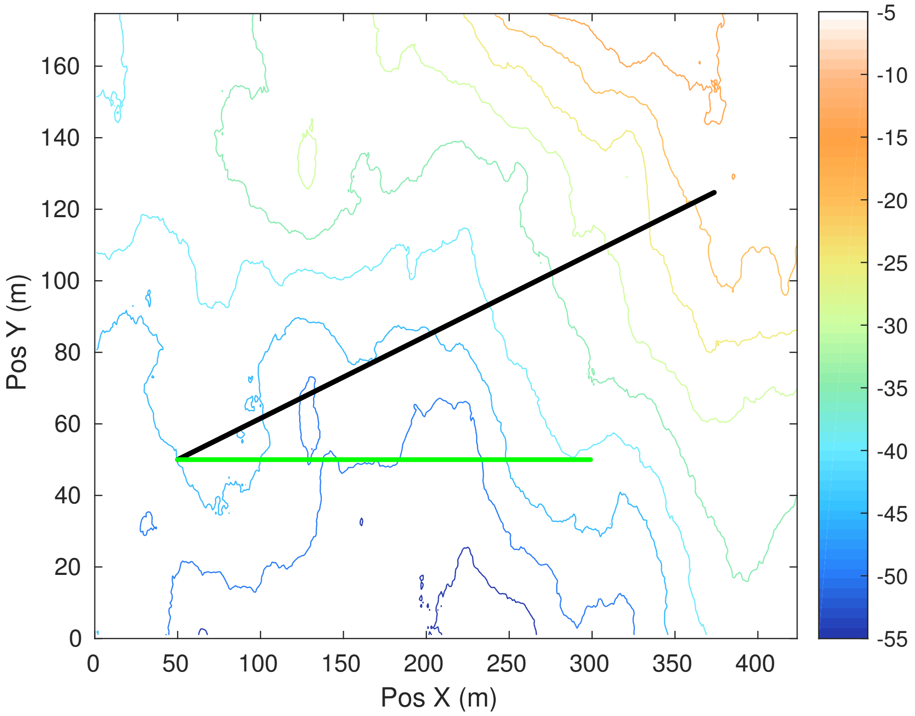


Figure 5.7: Simulated linear trajectories: desired trajectory, in green, and real trajectory with disturbances, in black.

To better compare the results, the ensemble RMS of 100 independent runs of both filters was computed. This was obtained using (5.24), thus computing the average root mean square error between the output of the filter and the real trajectory. Figure 5.9 compares the RMS errors of both filters. From there, it is clear the filters have similar performances in terms of RMS error, with both of them being able to closely follow the real trajectory of the vehicle, with values for the RMS error below $3m$. To put this value in perspective, this is similar to what can be achieved using simple GPS receivers.

However, Figure 5.9 also shows that the DD-PF can be more precise with an RMS error of only 2.1 meters, comparing with the 2.7 meters obtained by the SIR-PF. At the same time, from these plots it is also possible to infer that the position outputs of the DD-PF are relatively smoother than what was obtained with the SIR-PF. This can be relevant if, for example, the output of the TBN filter is going to be used as an input to a main navigation filter as suggested by some authors, for example Meduna et al. (2010).

Another metric of interest to compare the filters is the computational complexity of the filter. This is particularly relevant for sensor-limited systems, due to the limited computational power available on-board the vehicles. In order to compare the SIR-PF wit the DD-PF in terms of their complexity, two metrics are going to be used, namely the total number resampling steps performed in each run, and the elapsed time for each run of the filter. The average number of these two metrics should provide an indication on the efficiency of filters.

Figure 5.10 shows these two metrics for the SIR-PF and for the DD-PF, with the former being

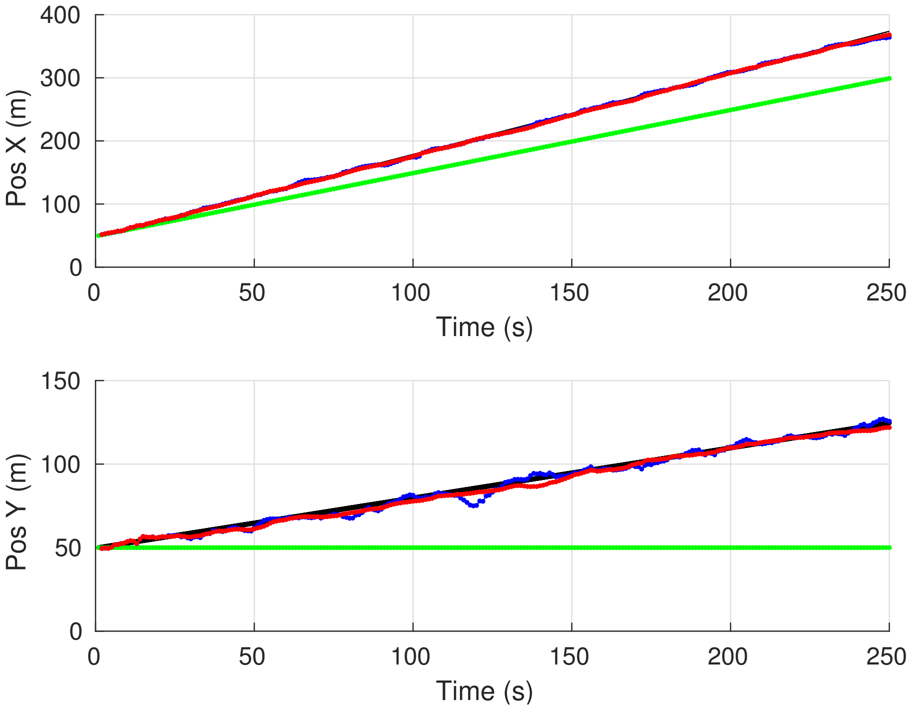


Figure 5.8: Output of the filters along the time for linear trajectories. SIR-PF in blue and DD-PF in red. Green and black are the desired and real trajectories, respectively.

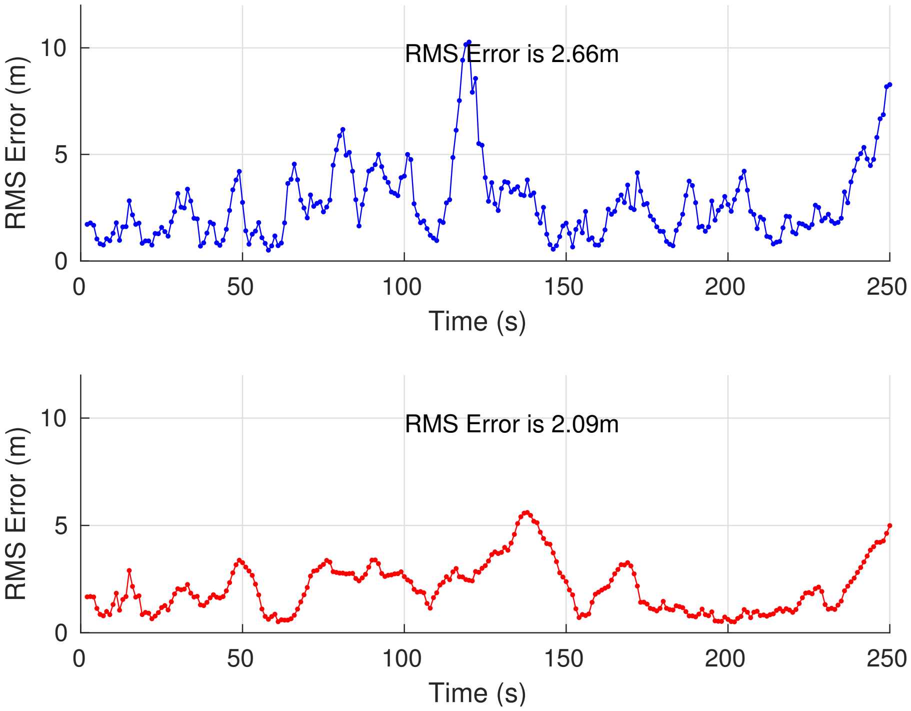


Figure 5.9: Comparison of the ensemble RMS Error of the filters for linear trajectories. SIR-PF in blue, and DD-PF in red.

represented in blue, and the latter in red. Figure 5.10a shows the number of resampling steps for each of the MC runs. Recalling form Algorithm 5.2, resampling of the particles happens whenever $N_{eff} < N_{thr}$, with N_{eff} being an indicator of the degeneracy of the algorithm. From Figure 5.10a it is possible to conclude that the DD-PF is more efficient than the SIR-PF, requiring almost half the number of resampling steps. In fact, the DD-PF requires resampling of the particles in approximately only 40% of the time steps, comparing to 75% of the SIR-PF. This is indeed a good indicator that the learned proposal density is close to the true posterior density.

The number of required resampling steps is an interesting metric to consider, as it is known that the resampling stage represents an important share of the total computational complexity of Particle Filters. However, the number of times a resampling algorithm is run does not provide any information on the increase in the global computation time of the filter. Figure 5.10b shows, in absolute terms, the processing time of both the SIR-PF and the DD-PF for each entire run. Naturally, processing time of the SIR-PF is higher than the DD-PF, which is a direct effect of the higher number of resampling steps required. Figure 5.10b shows that, on average, the SIR-PF requires at least 17% more processing time than the DD-PF. These values are obviously implementation dependent, with different implementations likely to obtain different values. Nevertheless, they are strong indicator of the efficiency of the DD-PF.

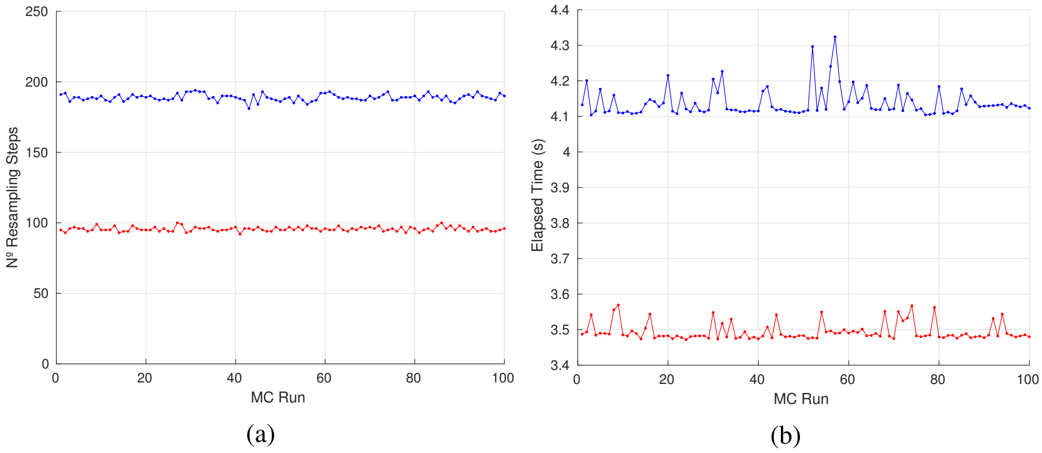


Figure 5.10: Comparison the complexity of the filters for linear trajectories. SIR-PF in blue and DD-PD in red.

The results just presented demonstrate that under certain conditions the performance of the DD-PF can be superior to the SIR-PF. This is not only in terms of the RMS error, but also in terms of its efficiency. Nevertheless, these simulation results refer to only a single scenario. Therefore, in order to analyse the consistency of the DD-PF with different levels of disturbances, another batch of simulations was run.

As previously mentioned, the simulations that were just presented correspond to the case when the vehicle is following a linear trajectory while being subject to disturbances that amount to approximately 40% of the distance travelled. This was the main scenario behind the derivation of the DD-PF, and corresponds to disturbances already very appreciable. Simulations for the

Disturbance (% DT)	SIR-PF		DD-PF	
	RMS (m)	Resampling Steps	RMS (m)	Resampling Steps
$\approx$ 40%	2.66	188	2.09	96
$\approx$ 15%	2.87	189	1.18	90
0%	2.99	196	1.05	86

Table 5.2: Data-Driven PF: Summary of the simulations for a linear trajectory, comparing the SIR-PF with the DD-PF

same scenario, but with smaller disturbances were also performed, in order to verify if the results obtained by the DD-PF would be in accordance with previous simulations. The same tests were repeated, but this time with disturbances amounting to 15% and 0% of the distance travelled. A summary of the obtained results can be found on Table 5.2.

From all the different sets of simulations, considering various levels of disturbances, the DD-PF was always able to outperform the SIR-PF. In all the situations, the RMS error achieved by the DD-PF was smaller, while at the same time the required number of resampling steps was roughly half of the SIR-PF. This was verified regardless of the magnitude of the disturbances affecting the system. It should be noted however the parameters shown in Table 5.1 remained the same in all the simulations. Therefore, while a high level of process noise is required when the disturbances are high, this is not the case when there are no disturbances. If this would be fine-tuned for each particular situation, the RMS error achieved by the SIR-PF for the case with no disturbances would likely be different.

5.7.1 Circular Trajectories

Previous simulations provided good indication that the DD-PF could outperform the SIR-PF for underwater TBN algorithms. This happens for scenarios considering only linear trajectories, covering a wide range of disturbances, from situations when no disturbances are observed, and up to disturbances of around 40% of the distance travelled. In here, a complementary set of simulations was performed, in order to evaluate the performance of the DD-PF for circular trajectories.

The simulations here presented are similar to the ones presented above, and using the same bottom topography. In this series of simulations the surge velocity of the vehicle is still considered to be constant and of $1ms^{-1}$, while no sway or heave velocity is considered. The same happens with both roll and pitch of the vehicle. However, in here the heading is not considered constant, but instead having a constant rate of change. This will induce circular trajectories. At the same time, the level of disturbances present will also be considered high, in line with the first scenario presented, and equivalent to 40% of the travelled distance. Figure 5.11 shows the trajectories considered. Once again, the green line corresponds to the desired one, while the black refers to the real one, with the influence of the disturbances.

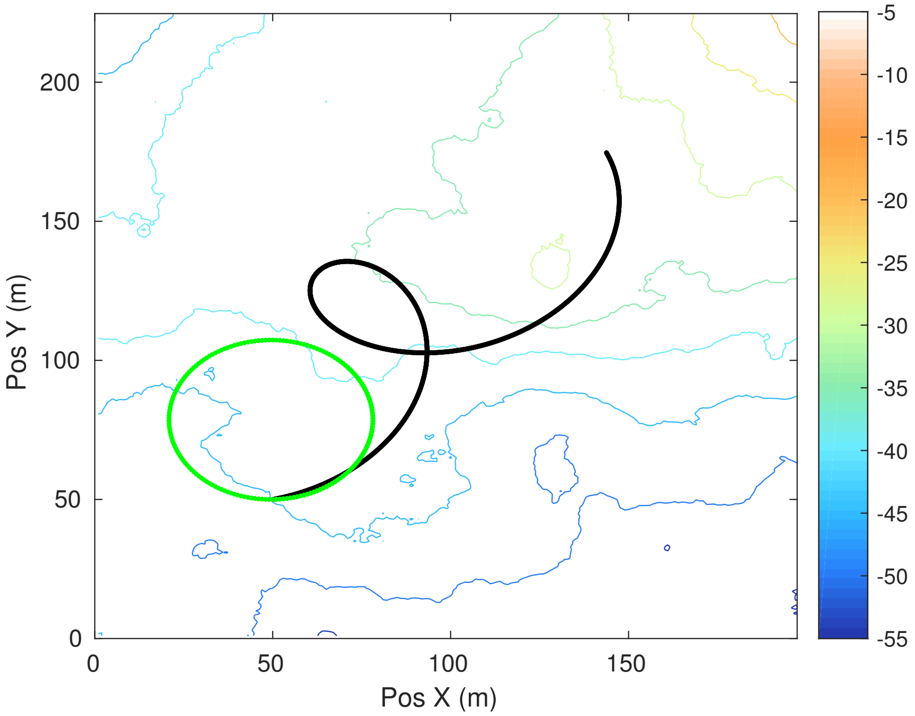


Figure 5.11: Simulated circular trajectories: desired trajectory, in green, and real trajectory with disturbances, in black.

Similarly to before, Figures 5.12 and 5.13 present, respectively, the position evolution of the filter's ouput, and the ensemble RMS error of both the SIR-PF and the DD-PF. As before, the green and black lines refer to the desired and real trajectories, while the blue line refers to the SIR-PF and the red line to the DD-PF.

Figure 5.12 shows the output of the filters under analysis, and compares them to the desired and real position of the vehicle along the time. In the figure it is possible to see that both filters are able to closely follow the true position of the vehicle. On the other hand, Figure 5.13 shows the ensemble RMS error of the two filters. By analysing this plot it is possible to conclude that in these simulations the SIR-PF, with an ensemble RMS of 2.33m, outperformed by a narrow margin the DD-PF, which achieved an ensemble RMS of 2.46m. Nevertheless, and in accordance to what happened previously, the plot of the ensemble RMS also allows to infer that the trajectories obtained by the DD-PF are rather smooth.

As before, the complexity of the two filters was also evaluated, by comparing the average number of resample steps per run of the filter. It was already demonstrated that the resampling step contributes decisively for the total complexity of Particle Filters. Figure 5.14 compares the number of resampling steps per run for this set of simulations. Once again, the blue points refer to the SIR-PF while the red ones refer to the DD-PF. Similarly to the linear trajectories scenario, also in here the number of resampling steps required by the DD-PF is significantly lower than what the SIR-PF requires.

By analysing Figure 5.14 in detail, it is possible to note that the DD-PF requires, on average,

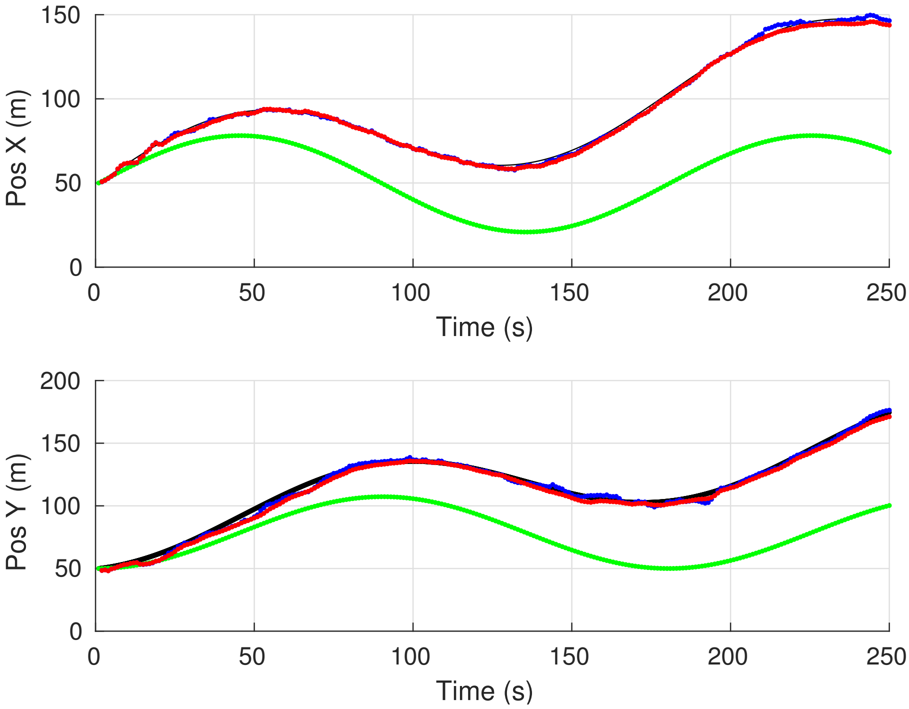


Figure 5.12: Output of the filters along the time for linear trajectories. SIR-PF in blue and DD-PF in red. Green and black are the desired and real trajectories, respectively.

137 resampling step, while for the same simulations the SIR-PF requires on average 229 resampling steps. In relative terms, that means that the on average the DD-PF resamples its particles on 55% of its iterations, while this number goes to as high as 92% of the iterations for the SIR-PF. Once again, this is originated by the fact that strong disturbances are present.

By comparing the numbers here obtained, with the ones obtained for linear trajectories, it is possible to note an increase in the required resampling step for both the SIR-PF and the DD-PF. This is in fact expected, as the process model considered does implicitly considers only linear trajectories. Interestingly, this increase is on absolute terms equal for both filters, requiring on average 41 additional resampling steps for scenarios with circular trajectories.

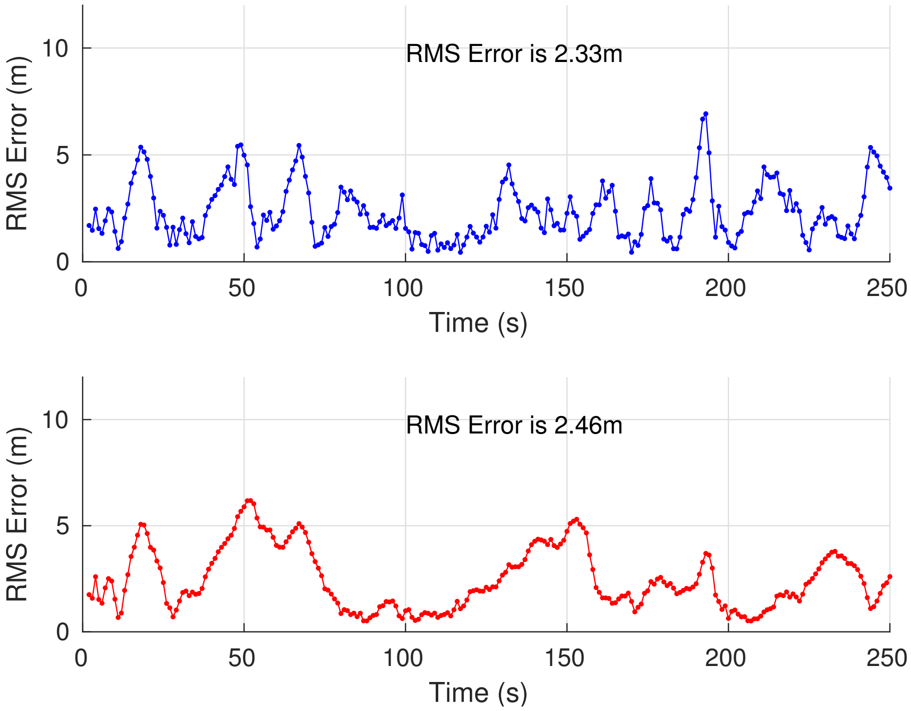


Figure 5.13: Comparison of the ensemble RMS Error of the filters for circular trajectories. SIR-PF in blue, and DD-PF in red.

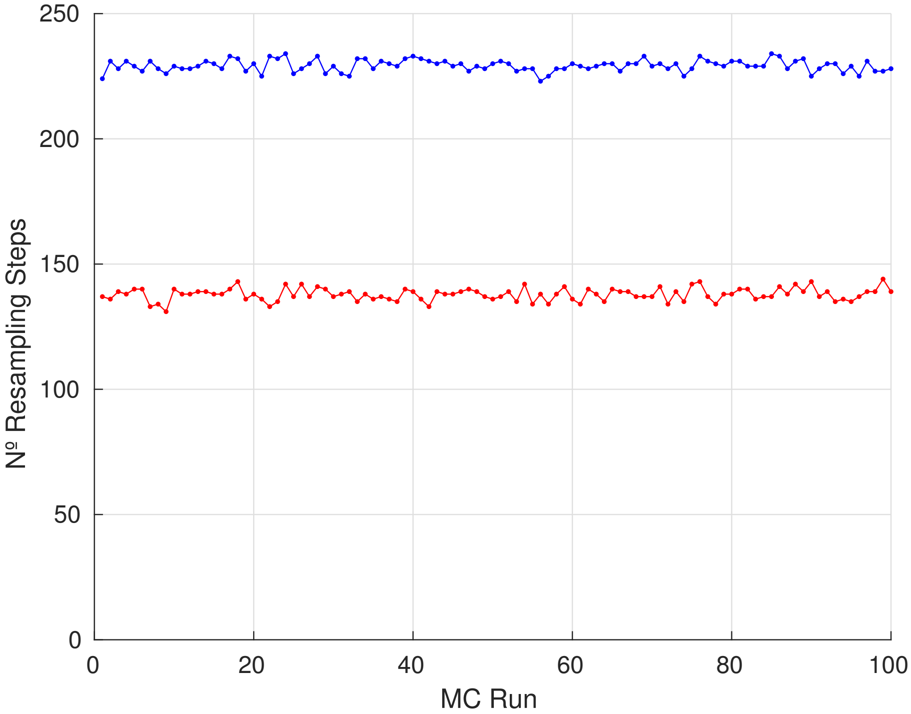


Figure 5.14: Comparison the complexity of the filters for circular trajectories. SIR-PF in blue and DD-PD in red.

5.8 Conclusions

This chapter discussed Terrain Based Navigation algorithms for underwater sensor-limited systems and, in particular, on the use of Particle Filters to address this problem. Due to the inherent strong linearities, non-linear Bayesian Filters like the Particle Filter are particularly suited to address TBN.

This chapter started with a brief introduction to Terrain Based Navigation. After a brief description, contextualizing the appearance of such methods, the particularities of TBN for sensor limited systems were presented, along with some background literature review. After that, the usual state-space formulation of the problem, including a detailed description of both the process and measurement models used was presented. Solving for the TBN problem requires some sort of non-linear filtering strategy, and the Particle Filter is one of the most used strategies present in the literature.

Therefore, the general particle filter framework was briefly uncovered. The SIR-PF, arguably the most widely used implementation of the particle filter framework, is also used in the majority of the approaches of TBN for sensor limited systems. Some simulation studies were performed to assess how the different parameters of the SIR-PF could influence the result of a TBN algorithm. The goal was to understand if by varying only the number of particles, the process noise and the measurement noise were it was possible to accurately estimate not only the position of the vehicle, but also its heading. It was verified that even in situations were the position estimates are obtained, the same is not possible for heading, with the estimated values always far from the actual heading. At the same time, it was verified that the process noise has paramount importance on the performance of the filter. If the process noise is small enough, the filter is dominated by the noisy inputs, but if its magnitude is high enough, than the filter is able to compensate them. An indirect effect of increasing the process noise is in the need to resample the particles more frequently.

Those initial simulations allowed for a better understanding TBN algorithms based on PFs. In fact, the need to increase the process noise, as mentioned before, only arises if the proposal distribution fails is far from the ideal posterior distribution. Observing that, the Data-Driven Particle Filter was proposed. While the SIR-PF uses the process model as proposal density, the proposed DD-PF tries to estimate the proposal distribution by learning from the data. It has been demonstrated by simulations that the DD-PF can outperform the usual SIR-PF, not only in terms of RMS error, but also in efficiency, requiring significantly less computational power. This can be very relevant mostly due to the limited processing power available on-board sensor-limited systems.

While the simulations for the DD-PF were focused only a two-dimensional position estimation problem, the DD-PF is still applicable for situations requiring a complete navigation solution, including estimation of the full pose of the vehicle. In such situations, the DD-PF could be used, for example, to generate position updates to a main navigation filter. A more integrated solution can be envisioned by using a Rao-Blackwellized version of the navigation filter, using the DD-PF approach when dealing with the non-linear structure of the system. A a natural extension of this work would be to study how the additional states could be used for a more efficient learning of the proposal distribution. While the approach here followed used linear regression as a way to approximated the learned distribution, would be interesting to study alternative learning methods.

Part III

Acoustic Navigation for Multiple Vehicles

Chapter 6

Tracking multiple AUVs

This chapter presents a novel method for the acoustic tracking of multiple Autonomous Underwater Vehicles. While the problem of tracking a single moving vehicle has been addressed in the literature, tracking multiple vehicles is a problem that has been overlooked. The proposed approach is based on a Probability Hypothesis Density Filter, thus overcoming the data association difficulties that affect traditional acoustic localization networks. The tracker presented is able not only to successfully estimate the positions of the vehicles, but also their velocities. Moreover, the tracker estimates are labelled, thus providing a way to establish track continuity of the targets.

6.1 Introduction

Nowadays, there are significant research efforts focused towards the development of algorithms that allow fleets of AUVs to achieve a common goal, navigating in a coordinated fashion. The use of multiple vehicles allows the parallelization of tasks that otherwise wouldn't be possible, thus reducing operations time. The potential for efficiency gains is even greater if the various vehicles are collaborating for the completion of a task. The use of teams of collaborating AUVs has been foreseen for different applications, for example archaeological missions (Tsiogkas et al., 2014) or mine counter-measures operations. (Prins and Kandemir, 2008). Even though there is an extensive and growing literature on cooperative control theory, there are only a few cases demonstrating complete multi-AUV cooperation in field trials in water. Examples of this are the efficient mapping of a given area using multiple vehicles, proposed by Paull et al. (2015). Teams of cooperating AUVs have also been reported to perform adaptive environmental sampling tasks, namely by having multiple vehicles performing plume tracking quickly and with high temporal and spatial resolution (Schulz et al., 2003). In a somewhat similar mission, sea trials with a fleet of ten autonomous underwater gliders deployed as an adaptive, coordinated ocean sampling network were reported (Fiorelli et al., 2006; Leonard et al., 2010). Similarly to the adaptive sampling problem, a distributed multi-vehicle patrolling approach was also proposed by Marino et al. (2015).

With such developments, it is reasonable to expect that in a near future new applications will arise requiring the operation of multiple AUVs concurrently, cooperating to achieve a desired goal. With the increase of such multi-vehicle missions of underwater vehicles, the problem of online tracking multiple AUVs becomes even more relevant. In fact, for most of the classical missions for AUVs, the ability to track the vehicles during operation time is not only desirable, but even critical for some more uncertain hazardous scenarios like with the military or oil industry applications.

Tracking AUVs can be done by listening to the acoustic signals exchanged between the vehicle and an acoustic network, previously deployed in the area of operations. The AUVs needs to emit an acoustic signal, that is then detected by each one of the beacons that compose the acoustic network. Frequently, if the beacons are also able to emit their own acoustic signals, they can also provide navigational aids to the vehicle. These beacons often constitute Long Baseline (LBL) acoustic networks. When a target emits an acoustic signal, it will be detected at different times by each one of the beacons, and according to their distance to the target. By combining the Time-of-Flight (TOF) of the acoustic signals, as detected by each one of the beacons, the position of a vehicle can be computed by using multilateration techniques.

In such configurations, tracking more than one vehicle requires that each one of the vehicles emits signals that can be easily distinguishable between each other. A natural way to comply with this requirement is to have the vehicles using different frequency modulated signals. Alternatively, time-division multiplexing schemes can also be employed. However, both options can have serious drawbacks, particularly when addressing situations with several vehicles. When using time multiplexing schemes, time slots are attributed to each one of the devices operating on the network, so they can emit acoustic signals. For operations with multiple vehicles, the number of time slots is increased, which in turn also increases the time interval between two consecutive signal emission for a given vehicle. As the number of vehicles increase, this can significantly degrade the tracker performance. On the other hand, resorting to different frequency signals is also not a scalable approach. Increasing the number of distinct frequency signals is cumbersome and costly, as it requires the development of specific hardware for emission and detection of the signals. This is even more complicated by observing that the acoustic signals used are usually in a very confined band, approximately within 10 to 30 kHz.

Motivated by this, the main contribution of this chapter is then the derivation of a suitable algorithm for acoustically tracking multiple AUVs that can cope with such limitations. The proposed tracker, which is based on a Probability Hypothesis Density (PHD) filter, doesn't require data association, which means that it can estimate the positions and velocities of the different vehicles, even when they are emitting similar and otherwise undistinguishable acoustic signals. Moreover, by using a suitable labelled formulation of the problem, the tracker is also able to provide trajectory estimates for each one of the vehicles. Another feature of the proposed method is the ability to address situations where the number of vehicles to track is unknown and varying in time, a scenario that can be easily envisioned for future multi-vehicle scenario. Up to the authors knowledge, it is the first time that such an approach has been proposed and experimentally validated.

This chapter proceeds as follows. Section 6.2 provides a review of work related to the topic of acoustic underwater target tracking. In Section 6.3 the single target tracking case is introduced.

Details on the FISST framework and derivation of the labelled Sequential Monte Carlo implementation of the PHD filter that will be used can be found, respectively, in Sections 6.4 and 6.5. Section 6.7 details the experimental setup used to collect the data used for the validation of the proposed approach, and presents the results obtained and finally, Section 6.8 discusses the attained results and presents some final remarks.

6.2 Related Work

The focus of this chapter is on the problem of tracking multiple AUVs, which is of relevance for the majority of AUV operations. This section provides a brief state-of-the-art review of the topic of acoustic tracking of AUVs. However, AUV tracking can also be considered as part of the bigger and more general area of tracking underwater acoustic sources. Research in this area has been investigated in other areas other than the AUV community, for example in the area of Underwater Wireless Sensor Networks (UWSNs). This will also be reviewed in this section. Nevertheless, the vast majority of the research so far has considered only the single target scenario.

Tracking of AUVs is perhaps the most known application of underwater target tracking for the robotics community. The methods for tracking a single vehicle are well established and have been fairly addressed in the literature. On its essence, they all rely on computing the position of the vehicle from a set of acoustic ranges between vehicle and a set of beacons, deployed whether at the surface or at the bottom of the sea. Examples of AUV tracking situations are, for instance, the work of (Watanabe et al., 2009), which described a tracking method to accurately estimate the position of an AUV relative to a mother ship, using a Super-Short Baseline Navigation acoustic network. On the other hand, Melo and Matos (2008) demonstrated how tracking an AUV can be performed with an inverted LBL network with only two acoustic beacons. Notwithstanding, in line with the vast majority of the literature both the references only address the problem of tracking a single AUV.

Some commercially available AUV tracking systems provide a way for simultaneously track multiple vehicles. One example of this is the ATACS system (Odell et al., 2002), which uses an identifier code transmitted with each signal, thus overcoming the problem of data association. This however, requires the existence of a data link between the nodes in the acoustic network. A different alternative is adopted by the GAPS systems (Napolitano et al., 2005), which makes use of different frequency modulated acoustic signals to be able to uniquely identify each of the nodes in the network.

Though not exactly the problem of tracking multiple vehicles, some closely related problems are also worth mentioning. For example, the problem of localizing multiple underwater acoustic sources using AUVs have been discussed in the literature. Choi and Choi (2015) address the problem of using a manoeuvring AUV equipped with several hydrophones to detect multiple acoustic sources, but considers only the case of a constant and known number of sources. Braca et al. (2014) propose the use of multiple AUVs as sensor nodes of a multistatic surveillance reconfigurable acoustic network with the purpose of detecting underwater targets in anti-submarine warfare environments. The use of UWSNs for acoustic tracking of underwater manoeuvring targets has also been addressed by several authors (Zhang et al., 2014; Wang et al., 2012). However, in all the aforementioned cases the authors only consider the single target case.

6.2.1 Tracking of Multiple Targets

From the majority of the methods in the literature that consider underwater target tracking, only a few are able to fully address the problem of multi-target tracking. Only seldom scenarios with multiple targets are considered and experimentally validated. The main obstacle in such cases is the ability to uniquely associate acoustic signals with the source that emitted such signals. This problem can be circumvented whether by transmitting a unique identifier code for each vehicle, or using different frequency coded signals, one for every target. While these approaches are proven, they are far from being optimal. Not only they require dedicated hardware, but also its scalability is limited, imposed by the scarcity of available frequency bands. In this chapter a different approach to the problem of tracking multiple AUVs is presented. In specific, the focus is on tracking multiple AUV targets, all emitting similar acoustic signals, which are not possible to distinguish between each other. Only a few authors addressed problems closely related to the one here in analysis.

Kreucher and Shapo (2011) describe a surveillance application on which a passive acoustic sensor array is used to monitor a given surveillance region and detect and track moving targets in the two dimensional space. The paper describes a Bayesian approach to track multiple targets using only acoustic bearing measurements from multiple passive arrays. The required Bayesian multitarget probability density was approximated using non-parametric methods and demonstrated using both simulated and real data.

The detection and tracking of multiple targets using multistatic sonobuoy arrays was also addressed by Morelande et al. (2015), using a Sequential Monte Carlo (SMC) approximation of the Multiple Hypothesis Tracker (MHT). However, in this case only simulation results were presented. SMC methods have also been adopted by Georgy et al. (2012); Georgy and Noureldin (2014) to address the problem of Multitarget Tracking in both multistatic active and passive sonobuouy systems. The authors detail a solution that is able to track an unknown time-varying number of multiple targets and keeping continuous tracks, in scenarios that depend on either bearing observations only, or both bearing and Doppler observations. Both papers present extensive simulation results, for scenarios with sixteen stationary receivers and two moving targets. On top of that, in the former experimental results were also presented.

The performance of several other multitarget trackers has also been assessed in similar environments, namely the Multi Target Tracker (MTT) (Lang et al., 2009) and a Gaussian-Mixture Cardinalized Probability Hypothesis Density Filter (GM-CPHD) (Erdinc et al., 2008; Georgescu et al., 2009), among others. The latter is in fact the most closely related to what is proposed in this chapter. In fact, the CPHD filter shares with PHD filter the same roots on the Random Finite Set (RFS) framework developed by Mahler (2008).

6.3 The single target case

The scenario under analysis in this section is the one of tracking a single vehicle when using an LBL based system. LBL systems are one of the most for robust, reliable and accurate configurations of Acoustic Navigation systems, and have been briefly described in Section 2.3.1. In this section, the motion and sensor models used for the single target tracking problem will be presented. This is done for the sake of clarity, as the concepts here presented also hold for the case of tracking multiple vehicles. The remainder of this section is loosely based on the work by Melo and Matos (2008), but with some adaptations found suitable for the present problem. The interested reader should refer to that article, and the references therein, for more depth in the topic.

Tracking an AUV requires it to be active, meaning that the vehicle needs to emit acoustic signals in order to be detected. The configuration here considered consists not only on the vehicle, an Autonomous Underwater Vehicle, but also on an Inverted LBL network with set of two acoustic beacons, or buoys. It is considered that the length of the baselines, which is the distance between each of the buoys, is long enough so that the depths that at which the AUV navigates are considered to be negligible. Thus, only the two-dimensional motion of the AUV on the horizontal plane is contemplated. If this would not be the case, then an additional acoustic buoy would be required, but the underlying principles would still apply.

We assume that the clocks sources of both the vehicle and the set of buoys are synchronized and with drifts that are small enough, so that OWTT techniques can be used throughout the entire duration of the missions, without any major concerns. What this assumption means is that all the systems, buoys and vehicle, share a common clock source and are aware of the exact time instant each of the systems emits a given acoustic signal. AUVs are then set to emit acoustic signals synchronously, and at a rate of one per second. Considering a speed of sound in the water of approximately 1500 m/s, this restricts the operations into an area of around 1500meters of distance to each of the buoys, a fairly mild assumption for shallow water missions. With this setup it is possible to directly compute ranges to each of the beacons from the ToF of the signals that each of the beacon detects. Figure 6.1 illustrates the scenario just described. The distances d_1 and d_2, which are the distances from the AUV to Buoy 1 and Buoy 2, respectively, can be easily obtained from the time-of-flight of the acoustic signals. Because the clock sources are synchronous,the time-of-flight will be equivalent to the time interval between the emission time instant, and the time instant when each of the buoys detects the received signal.

The range, or distance from the AUV to any of the beacons, with known positions, can then easily obtained as

$$d_i = c(t_{R_i} - t_{E_i}) \tag{6.1}$$

with c being the speed of sounds and $t_{E_i})$ and t$_{R_i}$ the emission and reception times, respectively.

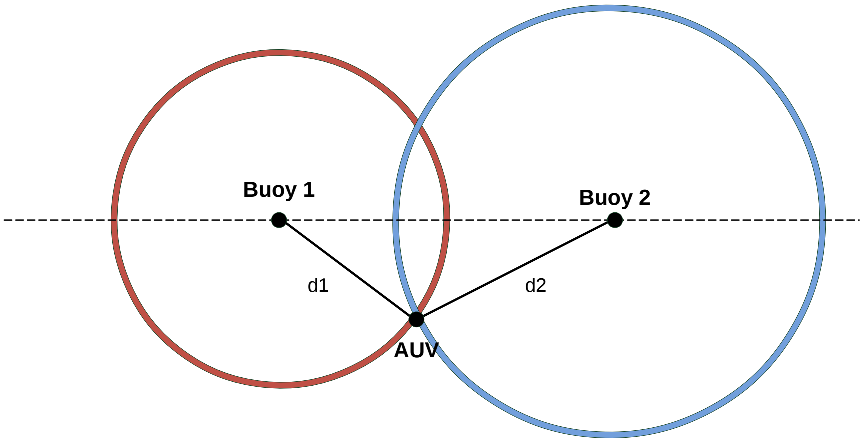


Figure 6.1: Schematic view of the setup required for tracking external AUVs

6.3.1 Target and Sensor Model

The system that was just described consists on an AUV navigating and periodically emitting an acoustic signal. The ToF of this signal are obtained by the two acoustic beacons deployed on specific positions, and then converted to range observations of this vehicle. Such scenario configures a typical target tracking scenario. The behaviour of the whole system can be described by the means of the single target dynamical model, f_k and the single target measurement model, (6.2) and (6.3), respectively.

$$\mathbf{x}_k = f_k(\mathbf{x}_{k-1}, \mathbf{v}_{k-1}) \tag{6.2}$$

$$\mathbf{z}_k = h_k(\mathbf{x}_k, \mathbf{n}_k) \tag{6.3}$$

In the equations above, $\mathbf{x}_k$ refers to the state vector of a target. It is usually considered that the interesting state variables to be estimated are the vehicle's horizontal position and velocity. The state vector $\mathbf{x}_k$ can then be defined as:

$$\mathbf{x}_k = \begin{bmatrix} x_k & y_k & \dot{x}_k & \dot{y}_k \end{bmatrix}^T. \tag{6.4}$$

Naturally, x_k and y_k denotes the position of the vehicle while $\dot{x}_k$ and $\dot{y}_k$ refer to the vehicle's velocity. It is further assumed that the dynamic model of the target follows a Gaussian constant velocity model, according to (6.5).

$$\begin{bmatrix} x_{k+1} \\ y_{k+1} \\ \dot{x}_{k+1} \\ \dot{y}_{k+1} \end{bmatrix} = \begin{bmatrix} 1 & 0 & \Delta & 0 \\ 0 & 1 & 0 & \Delta \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_k \\ y_k \\ \dot{x}_k \\ \dot{y}_k \end{bmatrix} + \mathbf{v}_k. \tag{6.5}$$

In the equation above Δ is the sampling interval and $\mathbf{v}_k \sim \mathcal{N}(0,Q)$ is the process noise, with matrix being Q the process noise covariance. On a given sampling period interval, each of the acoustic beacons produces one range observation of the target, that is related with the state vector $\mathbf{x}_k$ according to the measurement equation

$$z_{k,i} = h_i(\mathbf{x}_k) + \mathbf{n}_{k,i} \tag{6.6}$$

where h_i is the real valued function responsible for computing the expected range between the vehicle estimated current position, and the position of each of the beacons $(x_{0,i}, y_{0,i})$, and $\mathbf{n}_{k,i} \sim \mathcal{N}(0,\sigma_i)$ is the measurement noise in acoustic beacon i, assumed to be Gaussian:

$$h_i(\mathbf{x}_k) = \sqrt{(x_k - x_{0,i})^2 + (y_k - y_{0,i})^2} \tag{6.7}$$

A Bayesian estimation method is employed to estimate the position of the vehicle from the range measurements. Due to the non-linearity of the range measurements with respect to the system state, an Extended Kalman Filter is the most commonly used method. However, the use of an Unscented Kalman Filter or even a Particle Filter would also be suitable. Because the range observations are naturally noisy, with phenomenons like reflections and multipaths being detected by the acoustic transducers, outlier rejection strategies need to be employed. For this purpose, gating of the measurements can be done by comparing them with the normalized innovation squared (Matos et al., 1999; Fulton et al., 2000). A range measurement is considered valid if

$$\nu^T S^{-1} \nu < \gamma \tag{6.8}$$

where $\nu = \left[z_k - h_i(x_k)\right]$ is the innovation vector and S^{-1} corresponds to the innovation covariance matrix. The parameter γ is given by an appropriate χ^2 distribution, and it can be easily recovered from a distribution table for the desired confidence level. Alternatively, a suitable value can also be empirically determined.

Extending the single target case for the multiple target case is not straightforward, mostly due to the difficulty of correctly associate the acoustic signals emitted to it's source, one of the vehicles present. Moreover, it is desirable to have a framework that is able to jointly estimate number of targets and the target trajectory. The Random Finite Set framework, detailed in the following section, will be used to address exactly this problem.

6.4 RFS and the PHD Filter

Multi-object filtering applications, like the problem of tracking multiple vehicles, have been widely addressed, particularly by the radar tracking community. The objective of multi-object filtering is to jointly estimate the number of objects and their states from a set of observations. Both the Multiple Hypothesis Tracking (MHT) and the Joint Probabilistic Data Association (JPDA) have been presented in the literature as classical approaches for this problem (Bar-Shalom and Li, 1995; Stone et al., 2013). However, these traditional algorithms, rooted on the Bayes filtering paradigm present a number of pitfalls when addressing such scenarios.

Being based on the Kalman Filter, these algorithms rely on a vector representation, which requires stacking states and measurements from the different targets. This isn't a satisfactory representation when both the number of targets and measurements are random and varying. Additionally, a data association step, on which an explicit associations between measurements and targets is established, is required. In some situations this can be computationally very demanding, or even intractable.

An alternative formulation for the multiple target estimation problem, and one that only recently emerged, can be achieved by using random set theory to formulate the general multisensor multitarget Bayes filter. On such approaches, both the collection of individual targets states and the collection of measurements are modelled as Random Finite Sets (RFS), to obtain a set valued version of the general Bayes Filter. Loosely speaking, a random finite set can be thought of as a probabilistic representation of a collection of spatial point patterns that accounts for uncertainty in both the number of objects and their spatial locations. The usage of random finite sets, opposed to random vectors, is a more natural formulation for situations like tracking a varying number of targets or even target (dis)appearance and spawning, the presence of clutter and association uncertainty, false alarms, missed detections and even extended targets.

Developed by Mahler, the Finite Set Statistics (FISST) is a unified framework for data fusion, which has foundations on random set theory, and closely related to point process theory, the theoretical foundation for stochastic multi-object systems. FISST provides a set of mathematical tools that allows direct application of Bayesian inferencing to multi-target problems (Vo et al., 2005). The aim of FISST is to transform multisensor-multitarget problems into single-sensor single-target problems, by bundling all sensors into a single "meta-sensor", all targets into a single "meta-target" and all observations into a single "meta-observation" (Mahler, 2013). The multiple objects are treated as an RFS whose set members all belong to the same state space, and the detections are treated as an RFS whose set members all belong to the same observation. This is illustrated in Figure 6.2 (Granstrom et al., 2014). In that way, it is possible to construct true multisensor-multitarget likelihood functions and true multitarget Markov transition densities from the motion models and measurement models of individual targets and sensors.

Analogously to the traditional recursive single target Bayes Filter, presented in Section 2.5, the multisensor-multititarget Bayes filter propagates a multitarget Bayes posterior density $p_k(X_k|Z_{1:k})$

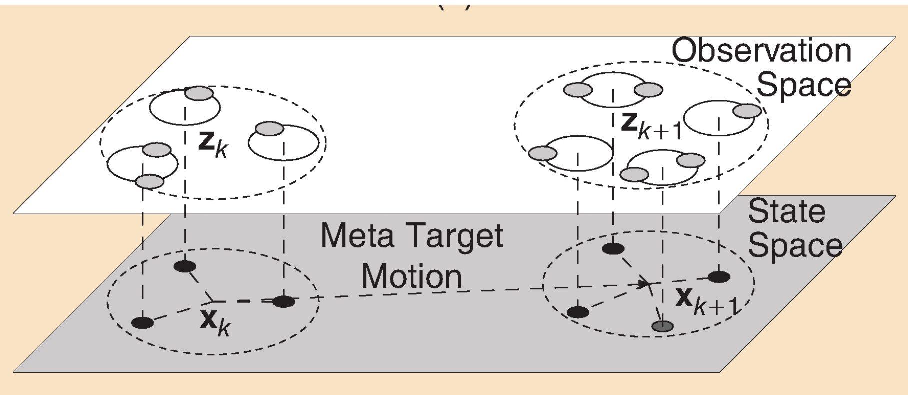


Figure 6.2: Illustration the basic concept of FISST theory, according to which the multisensor-multitarget problem is transformed in a single "meta sensor"-"meta target" problem

through time, conditioned on the sets of measurements up to time k, $Z_{1:k}$:

$$p_{k|k-1}(X_k|Z_{1:k-1}) = \int f_{k|k-1}(X_k|X)p_{k-1}(X|Z_{1:k-1})\delta X \tag{6.9}$$

$$p_k(X_k|Z_{1:k}) = \frac{g_k(Z_k|X_k)p_{k|k-1}(X_k|Z_{1:k-1})}{\int g_k(Z_k|X_k)p_{k|k-1}(X_k|Z_{1:k-1})\delta X} \tag{6.10}$$

Both the set of tracked objects X_k and the set of observations Z_k, at instant k, are modelled as random finite sets. The main difference between the recursion in (6.9-6.10) and standard clutter-free single-target filtering is that the dimensions of the random finite sets X_k and Z_k can change with time.

$$X_k = \{\mathbf{x}_{k,1}, \dots, \mathbf{x}_{k,M(k)}\} \tag{6.11}$$

$$Z_k = \{\mathbf{z}_{k,1}, \dots, \mathbf{z}_{k,N(k)}\} \tag{6.12}$$

In (6.11) $M(k)$ refers to the number of targets at instant k, while $N(k)$ in (6.12) refers to the the number of observations at the same instant. In fact, using random finite sets for modelling the multitarget state and the multitarget measurements provides an easy way to address situations like target birth and spawning, high density of clutter measurements.

The set of targets being tracked at a given time instant, $\mathbf{X}_k$, is composed by the collection of targets that survive from the previous time step, $\mathbf{S}_{k|k-1}$, together with the collection of spawned targets, $\mathbf{B}_{k|k-1}$, and the collection of new targets appearing only the in current time step, Γ_k, according to (6.13).

$$\mathbf{X}_k = \left(\bigcup_{x\in\mathbf{X}_{k-1}} \mathbf{S}_{k|k-1}(\mathbf{x})\right) \cup \left(\bigcup_{\mathbf{x}\in\mathbf{X}_{k-1}} \mathbf{B}_{k|k-1}(\mathbf{x})\right) \cup \Gamma_k \tag{6.13}$$

Similarly, the set of observations received at a given time instant (6.14) is a collection of both the measurements observed due to the present targets, Θ_k, which also includes including the probability of a missed detections, and the clutter measurements that may exist $\mathbf{K}_k$ in that time instant.

$$\mathbf{Z}_k = \mathbf{K}_k \cup \left(\bigcup_{z \in \mathbf{Z}_k - 1} \Theta_k(\mathbf{x}) \right) \tag{6.14}$$

Going back to the multitarget Bayes recursion (6.9-6.10), $f_{k|k-1}$ is the multitarget transition density, and $g_k(Z_k|X_k)$ is the multitarget likelihood. Even though the general multisensor-multitarget Bayes filters in intractable for the general case, with the use of appropriate calculus tools introduced by FISST, it is possible to derive approximations suitable to be implemented, like the PHD Filter.

6.4.1 PHD Filter

The first moment of an RFS is known as intensity function or Probability Hypothesis Density.The PHD filter, initially proposed by Mahler (2003), is perhaps the most popular approximation to the full Bayesian multitarget filter. In order to alleviate the complexity of propagating the full posterior, the PHD filter propagates only the first-order statistical moment of the RFS of states. In a way, the PHD filter can be considered the multitarget counterpart of the constant gain Kalman Filter, that also only propagates the first order moment of a distribution. Such approximation makes the multitarget Bayes filter computationally tractable. However, for this approximations to hold some assumptions must be observed, namely the signal to noise ratio (SNR) has to be high and all the targets should move independently of each other.

A probability hypothesis density function is characterized by the property in (6.15), which means that integrating a given PHD function $\mathbf{D}_k$ over the entire set of state spaces gives the estimated number of elements within the set, $\hat{N}_{k|k}$ Additionally, the peaks of $\mathbf{D}_k$ identify the likely position of the targets.

$$\int_S \mathbf{D}_{k|k}(\mathbf{X}|\mathbf{Z}^{(k)})d\mathbf{X} = \hat{N}_{k|k} \tag{6.15}$$

The PHD filter consists on two equations: the predictor equation (6.16-6.17) and the corrector equation (6.18-6.19). These equations are the core of the general PHD filter. While the PHD filter predictor equations allow the current PHD to be predicted and extrapolated, the corrector equations allow the predicted PHD to be updated with the new observations (Mahler, 2008).

$$\begin{aligned}\mathbf{D}_{k|k-1}(\mathbf{X}|\mathbf{Z}^k) &= \gamma_k(\mathbf{x}_k) + \\ &\int \phi_{k|k-1}(\mathbf{x}_{k-1})\mathbf{D}_{k|k}(\mathbf{X}|\mathbf{Z}^k)d_{\mathbf{x}_{k-1}}\end{aligned} \tag{6.16}$$

$$\phi_{k|k-1}(\mathbf{x}_{k-1}) = p_{S,k}(\mathbf{x}_{k-1})f_{k|k-1}(\mathbf{x}_k|\mathbf{x}_{k-1}) \tag{6.17}$$

$$\mathbf{D}_{k|k} \cong \mathbf{L}_{\mathbf{Z}_{k+1}}(\mathbf{X})\mathbf{D}_k(\mathbf{X}|\mathbf{Z}^k) \tag{6.18}$$

$$\begin{aligned} L_{\mathbf{Z}_{k+1}} &= (1 - p_D(\mathbf{X})) + \\ & \sum_{Z \in \mathbf{Z}_k} \frac{p_D(\mathbf{X}) L_Z(\mathbf{X})}{\lambda_k c_k(Z) + \int p_D L_Z(\mathbf{X}) \mathbf{D}_{k|k-1} d\mathbf{X}} \end{aligned} \tag{6.19}$$

In the equations above, $p_{S,k}$ refers to the probability a given target has to survive, from one time step to the following, $f_{k|k-1}$ refers to the transition density of a single target and γ_k refers to the intensity of spontaneous births. Moreover, c_k refers to the clutter spatial distribution and λ_k to the clutter rate. Also, p_D is the probability of detection of a target, and L_Z the measurement likelihood.

6.4.2 The Sequential Monte Carlo PHD (SMC-PHD) Filter

A closed formed solution for the PHD filter (6.16-6.19), has been derived by Vo et al. (2005). This filter, the Gaussian Mixture PHD filter (GM-PHD) admits only scenarios on which the targets evolve and generate observations independently, but also that follow a linear gausian dynamical model. Additionally, is is also assumed that the sensors follow linear and Gaussian measurement models. Recalling the single target scenario described in Section 6.3, there are non-linearities present both in the measurement model (6.3), thus this is not a suitable approach. Because of that, an approximation to the PHD filter recursion is more adequate.

The Sequential Monte Carlo PHD filter is an approximation to the general PHD recursion that, analogously to standard Particle Filters, uses randomly distributed particles to approximate the intensity functions that represent the PHD predictor and corrector equations. For that reason, another possible designation for the SMC-PHD is Particle PHD filter. In fact, for the case when there is only one target with no birth, no death, no clutter and unity probability of detection, the PHD filter reduces to the standard particle filter.

Considering the particle approximation, the PHD predictor equation can be rewritten as in (6.20), where the approximation is done with L_{k-1} particles, corresponding to the RFS containing the surviving targets, and J_k new particles, representing the RFS of the birth targets. For the general case the birth particles should cover the entire space of observation however, it is often the case that the prior knowledge regarding the location where new targets may appear is incorporated.

$$\mathbf{D}_{k|k-1}(\mathbf{x}_k) = \sum_{i=1}^{L_{k-1}+J_k} w_{k|k-1} \delta_{\mathbf{x}_{k-1}^{(i)}}(\mathbf{x}) \tag{6.20}$$

where

$$\mathbf{x}_{k|k-1} \sim \begin{cases} q_k(.|\mathbf{x}_{k-1}, Z_k), & \text{if } 1 \leq i \leq L_{k-1} \\ p_k(.|Z_k), & \text{if } L_{k-1} < i \leq L_{k-1} + J_k \end{cases} \tag{6.21}$$

and

$$w_{k|k-1} = \begin{cases} \frac{\phi_{k|k-1}(\mathbf{x}_k^{(i)}|\mathbf{x}_{k-1}^{(i)})}{q_k(\mathbf{x}_k^{(i)}|\mathbf{x}_{k-1}^{(i)}, Z_k)} w_{k-1}, & \text{if } 1 \leq i \leq L_{k-1} \\ \frac{\gamma_k(\mathbf{x}_k^{(i)})}{J_k p_k(\mathbf{x}_k^{(i)}|Z_k)}, & \text{if } L_{k-1} < i \leq L_{k-1} + J_k \end{cases} \tag{6.22}$$

In the same way, the SMC approximation for the PHD corrector can be rewritten as in (6.23):

$$\mathbf{D}_{k|k}(\mathbf{x}_k) = \sum_{i=1}^{L_{k-1}+J_k} w_{k|k} \delta_{\mathbf{x}_{k-1}^{(i)}}(\mathbf{x}) \tag{6.23}$$

where

$$w_{k|k} = \left[(1 - p_D(\mathbf{x}_k^{(i)}) + \sum_{\mathbf{z} \in \mathbf{Z_k}} \frac{p_D(\mathbf{x}_k^{(i)}) g_k(\mathbf{z}|\mathbf{x}_k)}{\mathbf{K}_k + c_k(\mathbf{z})} \right] w_{k|k-1} \tag{6.24}$$

with $\mathbf{K}_k$ being the intensity of the clutter measurement RFS and

$$c_k = \sum_{i=1}^{L_{k-1}+J_k} p_D(\mathbf{x}_k^{(i)}) g_k(\mathbf{z}|\mathbf{x}_k) w_{k|k-1}. \tag{6.25}$$

The particle transition density $\phi_{k|k-1}$ and measurement likelihood $g_k(z|\mathbf{x}_k)$ are obtained reusing the previously derived single target dynamical model (6.5) and measurement model (6.6), respectively. Further details on the derivation and convergence properties of the SMC-PHD filter can be found in (Vo et al., 2005; Clark, 2006)

Following the prediction and correction stages of the SMC-PHD filter, there is a need to resample the particles, just like in a standard particle filter, in order to prevent phenomenons of sample impoverishment. As with the PF, there is a common preference towards the use of systematic resampling, since this algorithm is easy to implement, has a linear computational complexity and, from a uniform distribution perspective, is theoretically superior (Hol et al., 2006). The systematic resampling method has already been presented in Chapter 5, and is detailed in Algorithm 5.1. The resampling stage of the SMC-PHD filter differs only from the traditional resampling strategies adopted in standard particle filters in that the normalized weights do not sum up to one, but instead, they sum up to the expected number of targets $\hat{N}_{k|k}$ (Clark, 2006), as in (6.26).

$$\hat{N}_{k|k} = \sum_{i=1}^{L_{k-1}+J_k} w_{k|k}^{(i)} \tag{6.26}$$

Because in the prediction stage of the algorithm there are always a number J_k of birth particles that are introduced, the number of particles is always increased on every time step of the filter. In the resampling stage of the algorithm the number of particles in the filter is downscaled to a number that is proportionally to the estimated number of targets. The filter estimate for the total number of targets, N_t^k, is given by the nearest integer of the to the total particle mass, here defined as $N_t^k = int(\hat{N}_{k|k})$.

Though not a integral part of the original SMC-PHD algorithm, target estimation plays an important role as it is the step where the locations of each of the targets are obtained. While in principle any general clustering techniques could be employed, there has been a strong preference of the community on using the two following methods. One way to perform this is to estimate the number of present targets in the current time-step, and then perform k-means clustering on the entire set of particles. Another alternative would be to model the posterior particle distribution as a multi-modal Gaussian, and then use an expectation-maximization algorithm to determine its parameters.

6.5 Refinement of PHD Filter

The previous section introduced the RFS framework and the standard formulation of the SMC-PHD filter, which was deemed appropriate for the problem of tracking multiple underwater vehicles. Nevertheless, the standard SMC-PHD filter still lacks some of characteristics that would be interesting for such application. For example track continuity method are highly appreciated when tracking multiple vehicles. In this section some refinements to the original PHD Filter will be presented, that are applicable when tracking multiple underwater vehicles with range only acoustic signals. Besides presenting a suitable track-labelling method, allowing track continuity to be established, this section also presents a method to address ambiguity of range measurements, preventing the appearance of ghost targets.

6.5.1 Track Labelling

In multi-target applications it is often necessary not only to obtain an estimated position of the multiple objects, but also to attribute a unique label to each target, so that each label is consistently associated with the same target along time. This allows to estimate not only the position of the targets but also their paths, or trajectories. Thus, the advantages of track labelling algorithms are obvious. The RFS formulation just described in the previous subsections gives no information on the track, meaning that there is no association between the estimated targets on a given time step, to the ones in previous step. Nevertheless, it is already possible to find in the literature some approaches for track labelling in PHD filters.

For example (Clark and Bell, 2007) suggested two alternative labelling methods. The first method is based on assigning labels to individual particles of the SMC-PHD filter. Alternatively, an estimate-to-track method method is also proposed, relying on finding the best association between estimated states and the predicted estimates. Similarly to this, Lin et al. (2006) proposed a track-labelling strategy on which the association between tracks and labels is obtained by optimizing the cost of associating the peaks to tracks.

A labelling solution for the specific case when at most one target is allowed to be born was presented by Ma et al. (2006). The solution for this particular problem was based on augmenting the the state vector, ξ_k, with a state variable that indicates the track identity γ_k.

$$\mathbf{x_k} = \left[\xi_k^T \, , \, \gamma_k \right]^T \tag{6.27}$$

Naturally, this transformation of the state vector also requires a convenient modification of the process model for the surviving particles. Considering that the target label remains constant, (6.5) needs to be changed accordingly:

$$\mathbf{x}_{k+1} = \begin{bmatrix} A & 0 \\ 0 & 1 \end{bmatrix} \mathbf{x}_k + \begin{bmatrix} B \\ 0 \end{bmatrix} \mathbf{w}_k \tag{6.28}$$

This simple strategy of adding a track label was also demonstrated to help on the state estimation process. Based on this work, a further refinement was made to address the general RFS multiobject tracking scenario, with the concept of Labelled RFS being introduced by Vo and Vo (2013).

With the notation introduced, the use of clustering algorithms, otherwise required to estimate the expected state of a given target, is no longer required. This is of particular relevance for SMC implementations of the PHD filter. Making use of the label variable introduced, state estimation can then be computed as:

$$\hat{\xi}_k = \frac{1}{N_k(\mathbb{R}^n, \mathtt{l})} \int \xi_k N_k(d\xi_k; \mathtt{l}) \tag{6.29}$$

That is, the expected state vector of the track $\mathtt{l}$, $\hat{\xi}_k(\mathtt{l})$ at time k, conditioned on the hypothesis that the track $\mathtt{l}$ is present in that time instant. $N_k(:, \mathtt{l})$ is the number of times that the track $\mathtt{l}$ is present at time k. For a target to be present it is usually tested if $N_k(:, \mathtt{l})$ is above a threshold, generally 0.5.

The complexity of SMC implementations is known to grow exponentially with the dimension of the state vector, therefore for the problem under analysis it would be interesting to keep dimension of the state vector to a minimum. On the other hand, in order to address a track continuity method the state vector could be augmented to include an additional state variable indicating the track identity. Having this in mind, the single target tracking case presented above, in Section 6.3, is now going to be reformulated.

Going back to the predictor equations of the SMC-PHD filter (6.20 - 6.22) and recalling that the transition density $\phi_{k|k-1}(\mathbf{x}_{k-1}) = p_{S,k}(\mathbf{x}_{k-1}) f_{k|k-1}(\mathbf{x}_k|\mathbf{x}_{k-1})$, the single-target dynamical model must be redefined to include the target label, $\mathtt{l}_k^i$. In addition, only system states concerning the targets position on the horizontal plane, x and y will be included in the state vector. With this changes, the state vector then becomes $\mathbf{x}_k = [x_k, y_k, \mathtt{l}_k]^T$. With this redefinition, the target motion model then becomes

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \begin{bmatrix} v_{x_{i,k}} \Delta \\ v_{y_{i,k}} \Delta \\ 0 \end{bmatrix} + v_k \tag{6.30}$$

with the quantities $v_{x_{i,k}}$ and $v_{y_{i,k}}$ corresponding to the estimated velocities of the different targets, and Δ to the time step duration. Because the complexity of SMC methods, like the one here in use, is known to scale up exponentially with the dimension of the state vector, it was decided to restrict it to a minimum. The different velocities of each target will be determined once the position of the targets is estimated, as detailed further ahead.

6.5.2 Observation Set

As for the corrector equations of the SMC-PHD filter (6.23 - 6.25), the measurement model $g(\mathbf{z}|\mathbf{x}_k)$ stems directly from the single target measurement model (6.6). However, the elements $z_{i,k}$ of the measurement set Z_k, have a slightly more intricate formulation. Because there is to association between detected signals and targets, all the ranges detected by the beacons need to be combined, in order to accommodate and adequate observation state.

At a given time step, each of the beacons will have a random number of detections, that are the result of the acoustic signals emitted by the vehicles, but also from possible clutter measurements that might exist. Thus, for a given beacon j, its corresponding detections during time period k will be

$$b_k^j = \{r_{1,k}^j, \dots r_{m,k}^j\} \tag{6.31}$$

where $r_{m,k}^j$ is the m-th range detected by beacon j at time k. The measurement set Z_k will then consist off all the possible combinations of the detections by every beacon. In that way, and considering only two beacons as previously specified, the i-th element of the measurement set Z_k will then simply be

$$z_{i,k} = \begin{bmatrix} r_{m,k}^1 & r_{n,k}^2 \end{bmatrix}^T, \tag{6.32}$$

and the number of elements of the measurement set will be $m \times n$. This process is of combinatorial nature, which can present some problems if the number of beacons is very high. However, this is not likely to be the case, due to the particular conditions of our application.

The general SMC-PHD filter assumes that new targets can be birthed from the entire observation space. Though a convenient assumption, this means a huge number of newborn particles must be drawn from a uniform density across the whole surveillance area. In this implementation an alternative path was chosen, as it is reasonable to assume that for applications where multiple AUVs are used, the positions from where the vehicles are usually launched in the water are known. For this reason it was assumed that new targets can appear spontaneously according to a Poisson Point Process with intensity function $\gamma = \mathcal{N}(.; x_\gamma, Q_\gamma)$ where x_γ and Q_γ represent the centre and variance of the location where AUVs are launched.

The intensity of clutter measurements $\mathbf{K}_k$ is modelled as a Poisson RFS uniformly distributed over the surveillance region as

$$\mathbf{K}_k = \lambda_k V u(z), \tag{6.33}$$

where λ_k is the average number of clutter returns per unit volume, V is the volume of the surveillance region, and $u(z)$ is the uniform density over the surveillance region. λ_k will be varying over time, and dependent on the number of targets navigating. The considered average number of clutter measurements present is proportional to $(\#AUV)^b - 1$, where b is the number of beacons.

6.5.3 Target Estimation

The fact that information about each of the targets label is now incorporated in the state vector of every particle simplifies a great deal the task of estimation each of the targets states. Obtaining the particles tracking a particular target resorts only to gather all the particles associated with a specific label. Hence, (6.29) can be approximated as

$$\hat{\xi}_k(\mathtt{l}) = \frac{1}{\hat{N}_k(l)} \sum_{i=1}^{L_k} w_k^i \mathbf{1}_{\Gamma_{\mathtt{l},k}}(\mathbf{x}_k^i) \xi_k^i \tag{6.34}$$

so that the state of given target can be estimated in an analogous way to the traditional particle filters. In (6.34) $\mathbf{1}_A$ refers to indicator function, or characteristic function, defined on a set **X**. This function indicates the membership of an element in a subset **A** of **X**, as follows:

$$\mathbf{1}_A(x) := \begin{cases} 1 & \text{if } x \in A, \\ 0 & \text{if } x \notin A. \end{cases} . \tag{6.35}$$

Additionally, $\Gamma_{\mathtt{l},k}$ is the set of all particles at time k that have the associated label $\mathtt{l}$.

Accordingly, the number of targets estimated by filter is given by the sum of all the weights, as in the particle filter. In the same way, a target with label $\mathtt{l}$ is estimated to be present if the sum of weights of the particles associated with that label is above a certain threshold, usually 0.5:

$$\hat{N}_k(\mathtt{l}) = \sum_{i=1}^{L_k} w_k^i \mathbf{1}_{\Gamma_{\mathtt{l},k}}(\mathbf{x}_k^i) \tag{6.36}$$

It is also on the Target Estimation step that newborn particles are promoted to new targets. All the particles without labels yet are summed and, if they are above a given threshold, they are promoted to a new target and their label is assigned. Despite the simplicity of the target estimation process, some care needs to be taken in order to prevent undesirable situations, like the appearance of ghost targets. This is particularly relevant since the resampling step of the SMC-PHD filter is agnostic to labels.

6.5.4 Deghosting

The position of the targets can be computed by using multilateration techniques, that combine the ranges measurements detected by each one of the different beacons,as made clear in Section 6.5.2. While one of these combinations will correspond to the actual position of a given target, the remaining ones will be considered as clutter measurements, and should be disregarded. However, in some situations, combining range measurements that are originated from disparate targets can generate a ghost target. Therefore, a deghosting strategy should be enforced, preventing such situations.

In target tracking scenarios involving multiple sensors, and particularly in multilateration applications, the appearance of ghost targets is recurrent. This happens because observations are naturally unlabelled, thus it is not possible to establish from which target they have been originated. The combination of sensor observations originated in different targets causes the arise of ghost targets. It is therefore very important to be able to disambiguate between real targets and ghost targets. Deghosting is the name give to the different techniques that are used to distinguish and removing ghost targets from true targets. In the literature, different deghosting approaches have been suggested, as for example the works by Mazurek (2008); Yang et al. (2013).

In our specific application, ghost targets are likely to arise whenever two or more targets are equally distant from one of the acoustic beacons. Because it is not possible to distinguish between the acoustic signals emitted by each of the vehicles, from that point onwards it is likely that a ghost target arises. Based on empirical evidence, a suitable deghosting heuristics has been implemented.

The followed strategy is based on monitoring the particle divergence for each target, $\Sigma_{\mathtt{l},k}$. If a ghost target arises, then the particles following a given target will divide and diverge, with a group of particles tracking the real target, and another group of particles tracking the ghost target. Therefore, if the divergence of the particles is above a certain threshold, then action is needed in order to prevent the appearance of a ghost target.

The implemented heuristics uses a k-means clustering algorithm to identify and partition the particles into two different clusters, $P^1_{\mathtt{l},k}$ and $P^2_{\mathtt{l},k}$. The two partitions will then be compared, with the partition with the highest cumulative weights corresponding to the true target, and the other one corresponding to the ghost target. Consequently, the particles associated with the ghost target will be disregarded, while the particles tracking the true targets will be resampled to the number of particles per target, N_p. The implemented heuristics is detailed in Algorithm 6.1.

Algorithm 6.1 Deghosting strategy heuristic

```
$\{\mathbf{x}_k^{(i)}, w_k^{(i)}\}_{i=1}^{N_p} \leftarrow$ DEGHOSTING$(\{\mathbf{x}_k^i, w_k^i\}_{i=1}^{N_p})$
for $\mathtt{l} = 1, \ldots, N_t^k$ do
        $\chi_{\mathtt{l},k} = \bigcup_{(\gamma_k = \mathtt{l})} \{\xi_k^i, w_k^i\}_{i=1}^{L_{k-1}+J_k}$
        $\Sigma_{\mathtt{l},k} = cov(\chi_{\mathtt{l},k})$
    if $tr(\Sigma_{\mathtt{l},k}) > \zeta$ then
        $\{P^1_{\mathtt{l},k}, P^2_{\mathtt{l},k}\} = k\text{-}means(\chi_{\mathtt{l},k})$
        if $\sum_{P^1_{\mathtt{l},k}} w_k^i > \sum_{P^2_{\mathtt{l},k}} w_k^i$ then
            $\{\mathbf{x}_k^{(i)}, w_k^{(i)}\}_{i=1}^{N_p} = resample(P^1_{\mathtt{l},k})$
        else
            $\{\mathbf{x}_k^{(i)}, w_k^{(i)}\}_{i=1}^{N_p} = resample(P^2_{\mathtt{l},k})$
        end if
    end if
end for
```

6.5.5 Velocity Estimation

Estimating the velocity of targets will have a paramount relevance in the performance of the tracker. The velocity will be important on the propagation of the particles that correspond to each of the targets on the most accurate direction, but also on preventing the appearance of ghost targets. Once the position estimates for each of the active targets have been computed, the individual velocities of each target will be determined. This can be done by using a Least Square Estimator

(LSE) with forgetting factor, similarly to what was done before, in Section 5.6. The advantages of using an additional estimator for the velocity, instead of augmenting the state vector detailed above, are mostly in terms of computational complexity. Additional, it is likely that the by using the LSE results in smoother and less noisy velocity estimates.

We consider that all the vehicles move in straight lines, and their movement in both x and y directions can be independently described by the following equations:

$$\begin{aligned} x(t) &= x_0 + v_x t \\ y(t) &= y_0 + v_y t \end{aligned} \tag{6.37}$$

In (6.37), $x(t)$ and $y(t)$ are the current targets positions, while t is naturally the time instant. On the other hand, x_0, y_0, v_x and v_y are the parameters to be estimated.

The estimation of the velocity occurs in three different moments. Whenever a target is first detected there is no prior information about its direction or speed, therefore random velocities are assumed in all directions:

$$v_{i,x}, v_{i,y} \sim \mathcal{N}(v_{ref}, \sigma_{ref}) \tag{6.38}$$

In a second moment, when there is already a number w of previous position estimates of a given target, the parameters are estimated using the general LSE estimator (6.39), where θ is the vector of parameters to be estimated and X and Y are the vectors of model variables and observations, respectively. This calculation provides the first estimation about the targets velocities.

$$\theta = (X^T X)^{-1} X^T Y \tag{6.39}$$

On a third moment, when there have been already more than w previous position estimates, a Recursive LSE (RLSE) with forgetting factor is implemented. The RLSE with forgetting has been widely used in estimation and tracking of time-varying parameters in various fields of engineering (Vahidi et al., 2005). Not only the RLSE requires less computational power, but the use of a forgetting factor is more appropriate for estimating time-varying parameters, providing somewhat smoother estimates with less delay, as more weight is given to more recent observations. The RLSE can be implemented with the following equations:

$$\begin{aligned} \theta_k &= \theta_{k-1} + L_k(y_k - \phi_k^T \hat{\theta}_{k-1}) \\ L_k &= P_{k-1}\phi_k(\lambda + \phi_k P_{k-1}\phi_k)^{-1} \\ P_k &= (I - L_k\phi_k^T)P_{k-1}\frac{1}{\lambda} \end{aligned} \tag{6.40}$$

The RLSE, in (6.40), presents a similar structure to the Kalman Filter. It consists on the equations that recursively compute the parameters θ_k, the gain, L_k and the covariance, P_k.

6.5.6 Implementation

As a summary of this section, the pseudo-code for the entire AUV tracker is provided, in Algorithm 6.2. The recursive algorithm can be informally described by the different stages: particle

Algorithm 6.2 Multiple AUV Tracker

```
$\{w_0^{(i)}, \mathbf{x}_0^{(i)}\}_{i=1}^{N_b} \leftarrow \textsc{Initialization}(N, \mathbf{D}_{0|0}, \hat{T}_0)$
for k = 1, ... do
    $\{\mathbf{x}_{k|k-1}^{(i)}, w_{k|k-1}^{(i)}\}_{i=1}^{L_{k-1}+J_k} \leftarrow \textsc{Prediction}(\{\mathbf{x}_{k-1}^{(i)}, w_{k-1}^{(i)}\}_{i=1}^{L_{k-1}})$    ▷ (6.21-6.22)
    $\{\mathbf{x}_{k|k}^{(i)}, w_{k|k}^{(i)}\}_{i=1}^{L_{k-1}+J_k} \leftarrow \textsc{Correction}(\{\mathbf{x}_{k|k-1}^{(i)}, w_{k|k-1}^{(i)}\}_{i=1}^{L_{k-1}+J_k})$    ▷ (6.24-6.25)
    $\{\mathbf{x}_k^{(i)}, w_k^{(i)}\} \leftarrow \textsc{Resampling}(\{\mathbf{x}_{k|k}^{(i)}, w_{k|k}^{(i)}\}_{i=1}^{L_k})$    ▷ Algorithm 5.1
    $\{\mathbf{x}_k^{(i)}, w_k^{(i)}\}_{i=1}^{L_{k-1}+J_k} \leftarrow \textsc{Deghosting Heuristics}(\{\mathbf{x}_k^{(i)}, w_k^{(i)}\}_{i=1}^{L_k})$    ▷ Algorithm 6.1
    $\{p_k^{(t)}\}_{t=1}^{N_t^k} \leftarrow \textsc{Target Estimation}(\{\mathbf{x}_k^{(i)}, w_k^{(i)}\}_{i=1}^{L_k})$    ▷ (6.36 - 6.34)
    $\{v_k^{(t)}\}_{t=1}^{N_t^k} \leftarrow \textsc{Velocity Estimation}(\{p_k^{(t)}\}_{t=1}^{N_t^k})$    ▷ (6.37 - 6.40)
end for
```

prediction, measurement correction, resampling step, target estimation and velocity estimation. On the prediction stage, the L_{k-1} particles that survived from the previous time step are propagated according the target dynamical model, and additional J_k birth particles are added to the particle set. All the $L_{k-1} + J_k$ are weighted according to (6.22). Following the prediction stage, in the corrector stage all the particles are weighted according to the measurement set (6.24). After that, the resampling step takes place, where particles with low weights are replaced by copies of the particles with higher weights. Finally, the targets positions can be estimated from the set of resampled particles, and from such position estimates the individual velocities of each of the targets is predicted.

6.6 Experimental Setup

This section provides the details of the experimental setup used in the field trials, that allowed the experimental evaluation of the tracker. The ideal setup would consist on a set acoustic beacons, and a set of multiple AUVs, navigating in open-water scenarios. However, with such setup it would be hard to assess the performance of the filter tracking multiple vehicles, due to the lack of the necessary ground-truth data. In order to overcome this, in the configuration under analysis it was chosen to replace the AUVs with ASVs.

The ASVs can mimic the behaviour of AUVs if they are equipped with an acoustic transducer that always remains underwater, and at a constant depth. By doing so, they can be used as AUV surrogates. On one hand this allows to have access to a series of acoustic underwater slant ranges obtained between moving vehicles and beacons. On the other hand, because the ASVs are equipped with GPS receivers, it is possible to have access to their GPS position and velocity, which will be used as ground truth of these experiments. The devices used in these trials, the ASVs Gama and Zarco, and the acoustic beacons, all have been presented in Section 2.6.

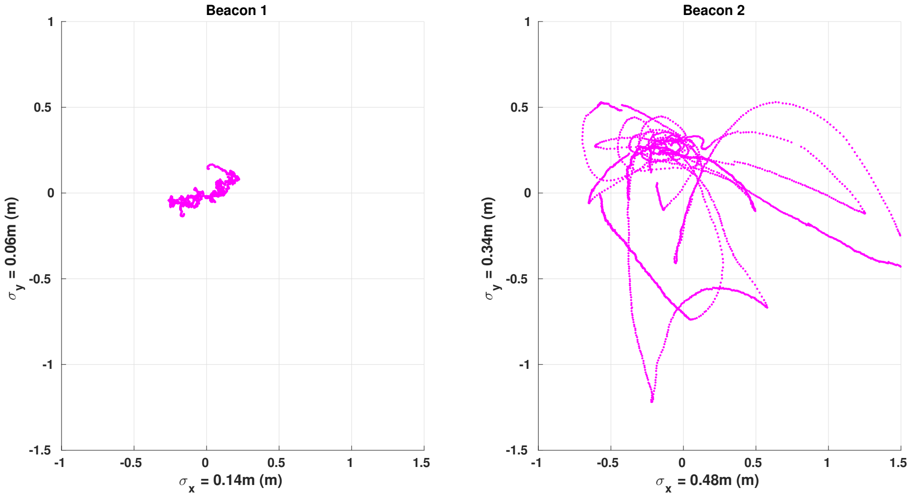


Figure 6.3: Dispersion of the position measurements for the two beacons. It can be seen that the dispersion of the position is bigger for beacon 2, with $\sigma_x = 0.48m$ and $\sigma_y = 0.34m$.

6.6.1 GPS Measurements

All the devices used, beacons and vehicles, are equipped with GPS receivers which provide accurate position data throughout the duration of the trial. This data will serve as ground truth of the whole experiment, and will be compared with the position of the targets estimated by the tracker. Therefore, is of utmost importance to understand how the variance of the GPS position measurements can affect the results obtained

While it is assumed that the position of the beacons remains the same, that is not necessarily true. Even though the beacons are moored, they can nevertheless be affected by water currents that might occur. At the same time, it is known that positions obtained by GPS have some intrinsic error that should be considered. Figure 6.3 plots the dispersion of the position measurements for the two beacons used, during the entire duration of the trial. The dispersion of the positions is different for the two beacons, probably reflecting the spatial variability of the currents. Nevertheless, the obtained standard deviation always remains below 0.5m, as indicated in the plots.

The GPS of the systems is also going to be used for clock synchronization purposes. As previously mentioned, similar clock synchronization procedures have been commonly described with OWTT acoustic navigation systems. The clock synchronization, essential for the good performance of the systems, can be achieved by using the PPS signals available from the GPS receivers used. It was experimentally verified that synchronism between all the receivers can be achieved up to $25ns$. Considering sound speeds of around $1500m/s$, this small difference in the PPS signals induces positions variations below the millimetre scale, which is considered to be suitable for the present application.

6.6.2 Range Measurements

Both the ASVs used in this field trials, but also the beacons, are equipped with appropriate acoustic systems, that have been characterized in Section 2.6.4. These systems are responsible to emit the acoustic signals and precisely timing their detection. Then, the acoustic signals can be converted to ranges, provided that the speed of sound in the area of operations is known. Prior to the experimental validation here detailed, the necessary procedures to estimate an accurate speed of sound in the vicinity of the area of operations were performed. This procedure is detailed in Appendix A.

For these field trials, it was assumed that the sound speed profile was locally homogeneous, meaning that the speed of sound is assumed to remain constant in the whole area of operations. Furthermore, it was also assumed than the slant ranges obtained by this method correspond directly to a distance on the horizontal plane. This is, in fact, approximately true, considering the vehicles were operating in shallow waters. Moreover, because the transducers of all the devices were all mounted having the same depth, only the horizontal position is considered.

In the field trials described in this section, two acoustic beacons were used to detect the acoustic signals emitted by the vehicles. A set of slant range measurements collected by these buoys, from this point onwards referred to B_1 and B_2, are depicted in Figure 6.4. The ranges correspond to the acoustic signals emitted by two distinct vehicles, corresponding to the blue and red colours in the figure. This clear distinction between the signals was achieved by having the two vehicles emitting signals in different frequencies, and was used only for a better data analysis and processing. Despite that, it should be noted that for the remaining of the analysis, the range measurements used in the filter were stripped down from any identifier that could potentially identify the origin of any of the signals.

In Figure 6.4 the red and blue points correspond, respectively, to the ranges obtained by B_1 and B_2 originating from each of the vehicles. These ranges are the actual data used in the tracker for estimating the position of the two vehicles. It is clear from the figure that there is a continuous trend line for each of the vehicles, corresponding to their actual trajectories. However, it can also be observed that a high number of clutter measurements have been observed, particularly by B_1. Such outlier points are expected and common when dealing with underwater acoustic signals, and they arise from multipath phenomenons that affect acoustic signals. Multipaths usually correspond to situations when the acoustic signals suffer a reflection, on either the bottom and the surface, or the margins, before being detected. The fact that B_1 detects a lot more reflections than B_2, can probably find an explanation on the geometrical configuration of the setup.

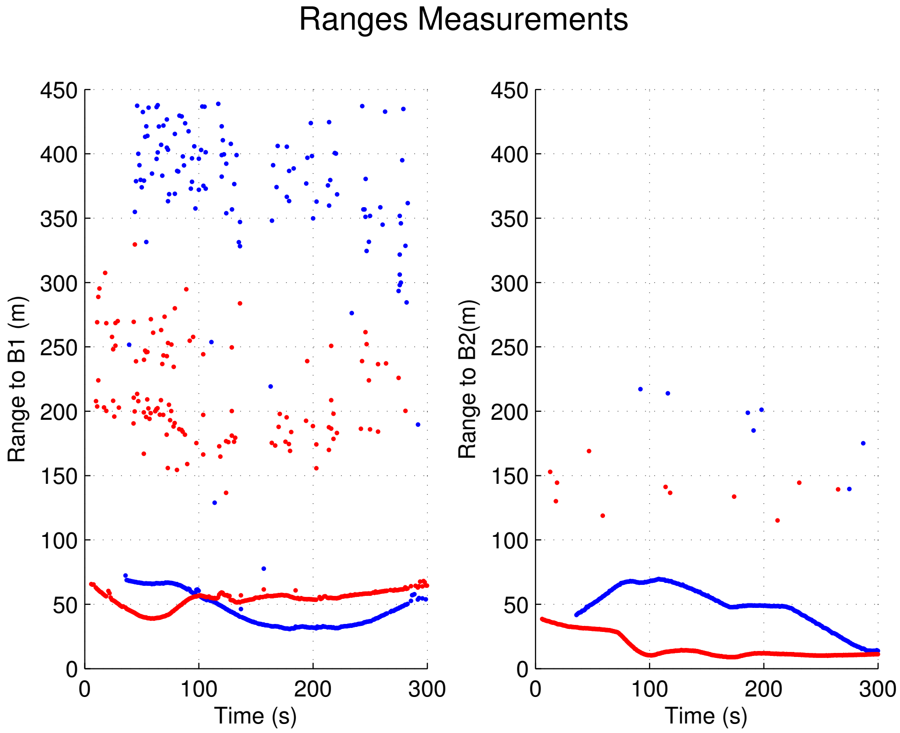


Figure 6.4: Dispersion of GPS Position Measurements from both beacons.

6.7 Field Trials

The work presented is this chapter is devoted to the development of a tracker that is able to acoustically track multiple AUVs navigating simultaneously. In a series of trials performed in February 2015, in a location in the Douro river, a few kilometres upstream from Porto, in Portugal, a set of acoustic slant range measurements were collected, between all the devices. This measurement set was used for the for the experimental validation of the AUV tracker, derived in the previous sections, and the results of it will be presented in this section.

The main goal of the presented filter is to estimate, in real time, the position of multiple vehicles using the acoustic ranges between each of the vehicles and a set of buoys, or acoustic beacons. The tracking results were obtained by using the collected set of range measurements, shown above, with the Multiple AUV Tracker derived in the previous sections. The different parameters of the tracker used are specified in Table 6.1.

One of the missions performed for the experimental evaluation of the tracker, and the one that will be here presented, consisted on having two vehicles navigating simultaneously in a predefined area, under surveillance of two moored acoustic beacons. The vehicles were set to start emitting acoustic signals at different times, in order to illustrate the ability of the filter to detect new vehicles entering the surveillance area. At the same time, the vehicles were navigating with arbitrary varying velocities and in different directions, but also being in stationary.

An overview of one of the trajectories performed by each of the vehicles can be seen in Figure 6.6. The real trajectories, given by GPS, are shown in dashed black line, while the estimated ones

Parameter	Value
Filter Settings	
Particles Per Target (N)	1000
Particles Per Birth (M)	1000
Predictor Settings	
Prob. of Survival (p_s)	0.99
Prob. of Birth (p_b)	0.01
Process Noise Variance ($\sigma^2_{x,y}$)	0.1
Corrector Settings	
Prob. of Detection (p_d)	0.6
Measurement Noise Variance ($\sigma^2_{r,B}$)	0.8
Clutter Intensity (λ_k)	$10^{-5}\sqrt{\vert\mathbf{Z}_k\vert)}$

Table 6.1: Parameters of the Multiple AUV Tracker

are marked in blue and red, respectively. At the same time, the position of each of the buoys, marked with B_1 and B_2, is also indicated, as well as the area where new targets are expected to be birthed .

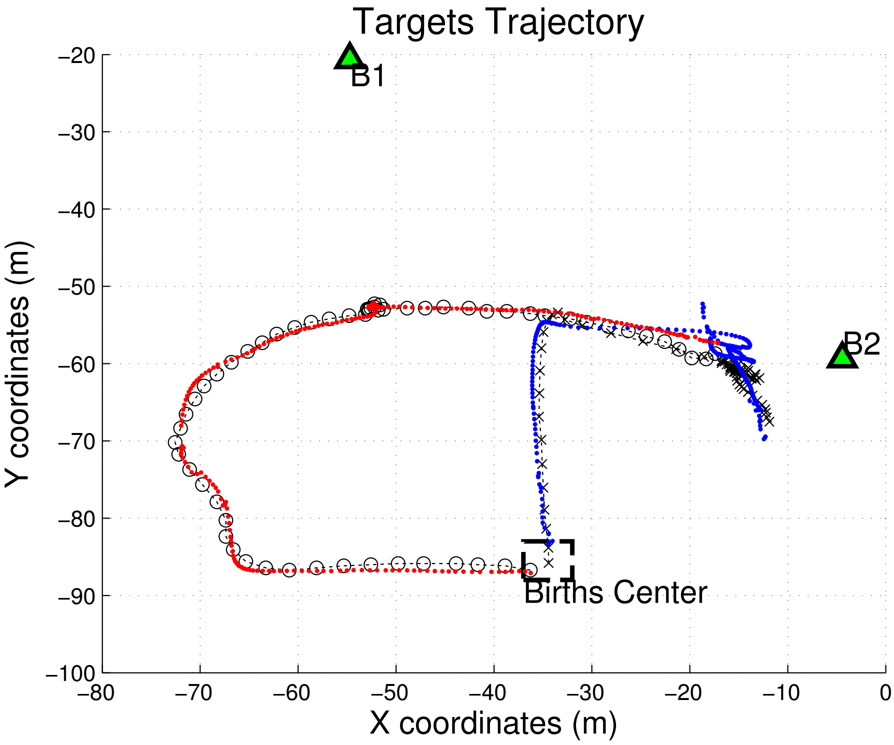


Figure 6.5: Overview of the mission: trajectories of the vehicles and position of the beacons

Figure 6.6 shows the time evolution of the estimated positions of the two targets, and compare it with the ground truth, given by the GPS position of the targets. This is useful for better understanding the accuracy of position estimations given by the tracker. Besides the estimated position, the plot also includes the standard deviation of the the estimation. It can be seen that the estimated trajectories of the vehicle closely resemble the trajectories given by the ground truth data.

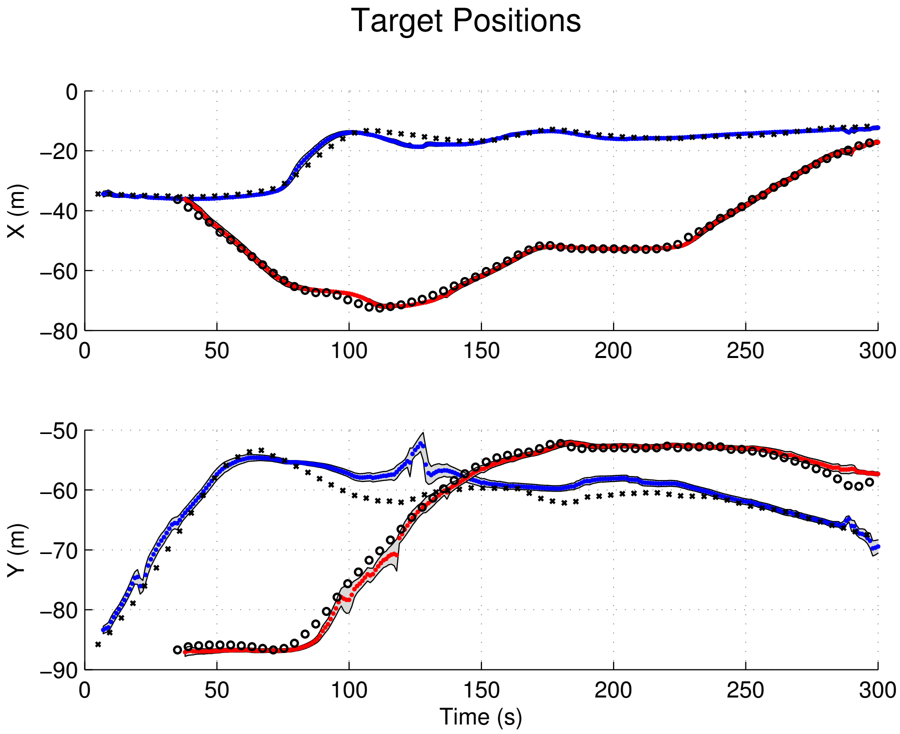
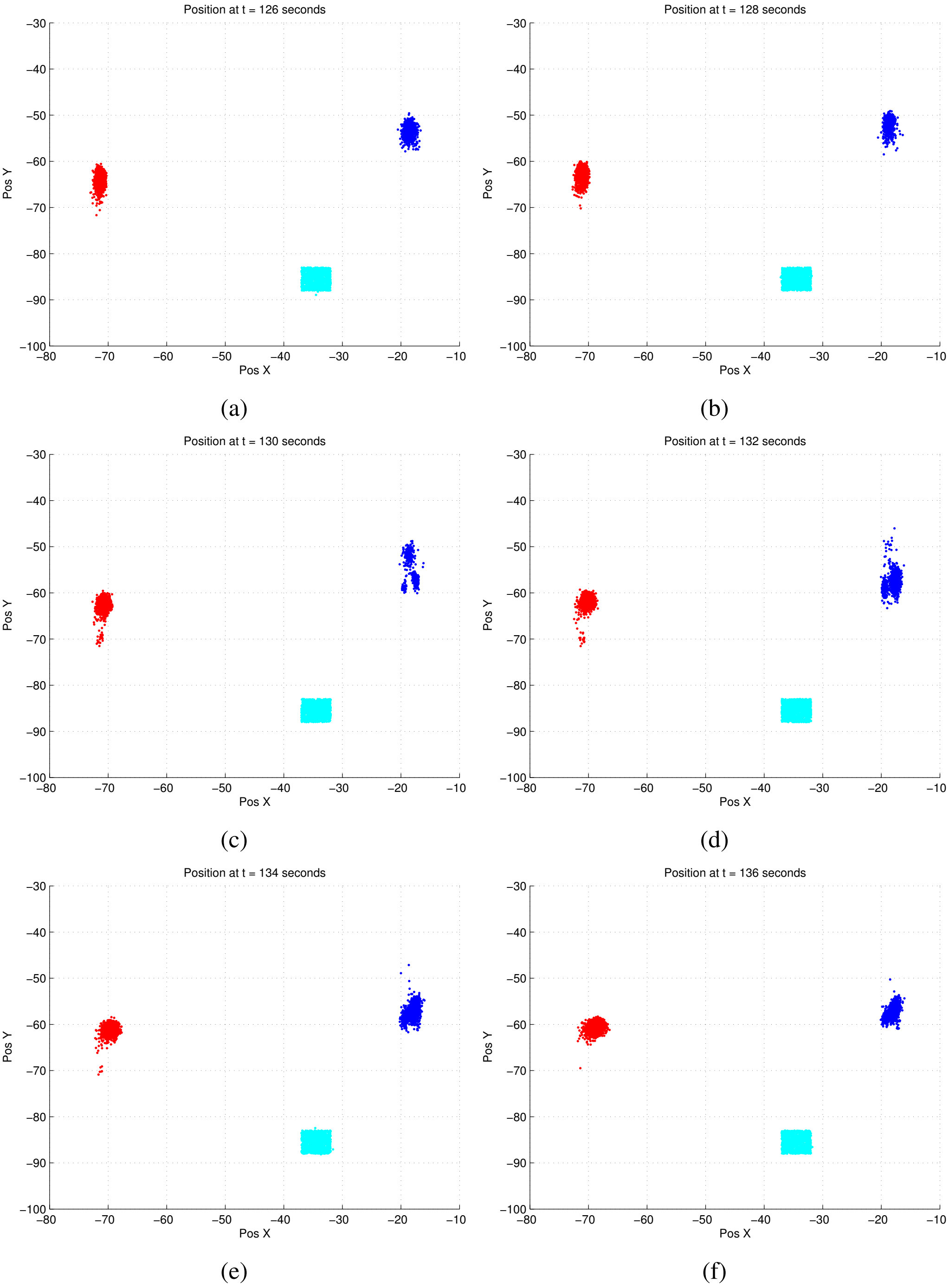


Figure 6.6: Time evolution of the position of the targets, in blue and red, respectively. Dashed line is the ground truth.

A closer look into Figure 6.6 reveals that in between seconds 125 and 135, approximately, the position of the target in blue colour diverges from the ground truth for some time, but then quickly recovers. At the same time, in this period an increase in the variance of the estimated position is noticeable. This behaviours is caused by a situation on which the ranges received by one of the beacons, in this case B_1, from both the vehicles have similar magnitude. This corresponds to a situation where both vehicles are at equal distances to one of the buoys, and configures a situation on which ghost targets are likely to arise, as described in the previous section. Figure 6.7 details the behaviour of the filter in that situation, by showing the distribution of the particles tracking each of the vehicles.

The sequence in Figure 6.7 illustrates the behaviour of the developed deghosting heuristics, preventing the creation of a new ghost target. At $t = 126$, in 6.7a, the particles that track each of the targets are concentrated around the actual position of the targets. In 6.7b and 6.7c it can be seen that the particles corresponding to each of the targets start to be less concentrated and more spread in space. It can even be perceived two different clusters of particles being formed in the vicinity of each of the targets. This is more visible for the target in blue than for the target in red.



(a) (b)

(c) (d)

(e) (f)

Figure 6.7: Illustration of a situation where the deghosting algoritm successfully eliminates ghost targets

The deghosting strategy implemented prevents the creation of ghost targets, and in 6.7d, 6.7e and 6.7f the success of such strategy can be observed, with the two apparent clusters becoming only one.

Figures 6.5, 6.6 and 6.7 demonstrate the good tracking performance of the Multiple AUV Tracker, as it is easy to see that the estimated positions of both targets closely resemble the ground-truth GPS data, with their differences being well within the expected. In fact, the obtained root mean square (RMS) errors for the entire duration of this experiment were, respectively of $1.98m$ for the target in blue, and $1.20m$ for the target in red. Figure 6.8 shows the absolute error in position between estimated and ground truth positions.

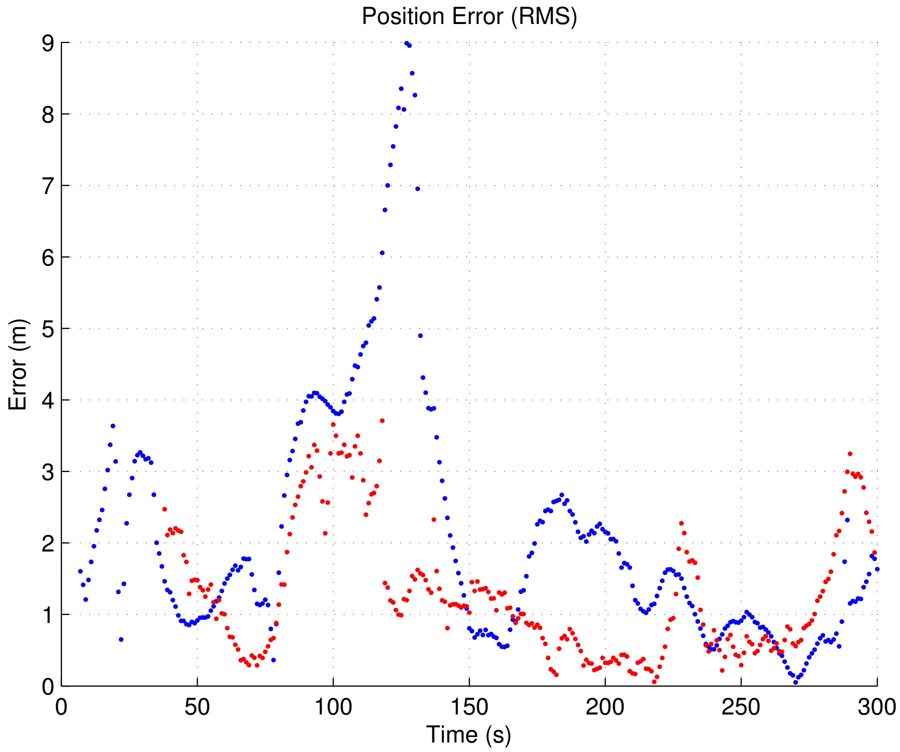


Figure 6.8: Absolute errors between ground truth and estimated positions

By analysing Figure 6.8, it can be concluded that the absolute error in position stays well below the 2 meters for several occasions, which corresponds to situations where the targets move in straight lines. Conversely, when the vehicles change direction, this error increases. The situation depicted in Figure 6.7, when the occurrence of a ghost target is prevented, has obviously repercussions on the error in tracking the vehicle, which corresponds to the peak present in Figure 6.8, with the error being of around $10m$ for one of the targets. Nevertheless, it can also be observed that as the tracker recovers from this situation, also the tracking error decreases. It should be noted that RMS errors for both the vehicles is comparable to the precision obtained by common GPS receivers while operation in single mode precision.

Finally, a comparison between the estimated velocities of each of the targets and the velocities provided by the navigation layer of each of the vehicles can be seen in Figure 6.9. While this is an interesting comparison, it should be noted that the ground-truth velocities are provided by the navigation layer of the on-board software of the vehicles. Therefore, it should not be understood as ground truth. In fact, such values are itself also estimated and because the navigation instruments available are not very accurate, they might not reflect the exact speed of the vehicles. Nevertheless, as it can be seen in Figure 6.9, the speed profiles are moderately similar, especially for the red target. As before, the velocities estimated by the tracker are more close to the ones provided by the navigation layer when the vehicles are moving in for straight-line.

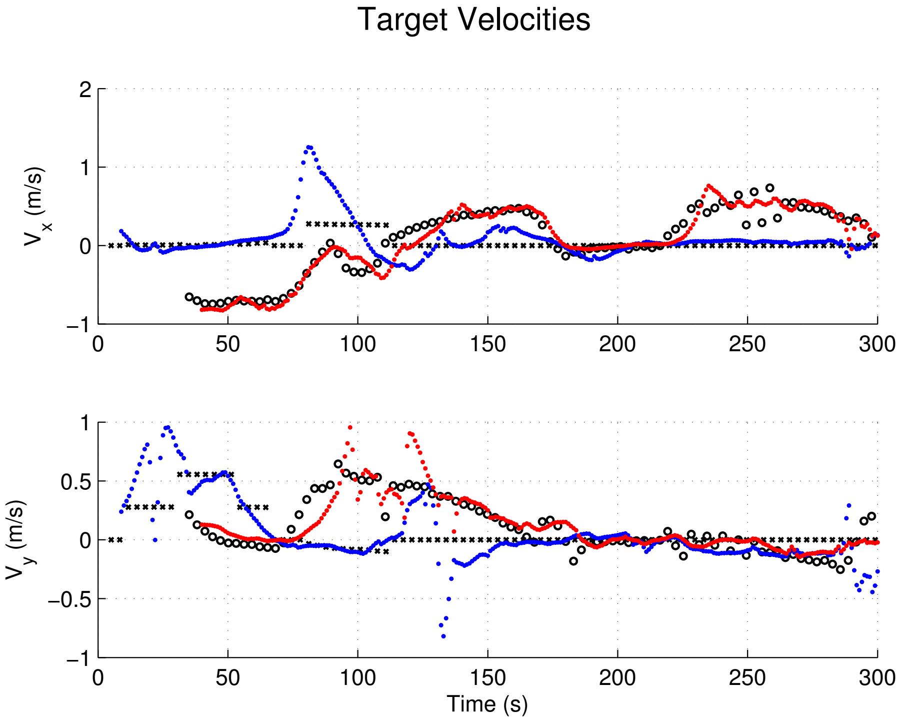


Figure 6.9: Trajectories of targets t_1, t_2, and t_3, in blue, red and green, respectively

6.8 Conclusion

Operations with multiple vehicles are likely to become very appealing in a near future, not only in terms of flexibility and efficiency, but also in terms of performing a set task that otherwise wouldn't be possible. While the problem of providing navigational aids to multiple vehicles has been addressed in the past, tracking multiple AUVs is a problem that has been overlooked. In this chapter an effective AUV tracker able to track multiple vehicles in real time was presented. Moreover, its performance was successfully demonstrated in real world scenarios. Up to the authors knowledge, this is the first time that a similar approach has been proposed and experimentally validated.

It was demonstrated that the problem of tracking of multiple AUVs can be successfully tackled using PHD filters. The RFS nature of the PHD filter does not require a specific association between measurements and the targets that produced them, thus making it appropriate for this problem. Also, this strategy avoids the data association problem. The tracking results achieved were very positive, with RMS errors under 2 meters, which is of similar precision to what can be obtained with a GPS receiver. Moreover, the proposed filter is also able to keep information of the track continuity along the time. Such results were obtained despite the quite adverse trajectories performed by the vehicles, very close to each other, and with the vehicles moving with arbitrary low speeds. It should also be underlined that such results have been achieved despite the unfavourable geometry of the acoustic network, with only two beacons and with a baseline not very long.

While the results achieved are very encouraging, it can be argued that the results are limited to situations with only two targets. While this is true, there is no indication that the developed tracker would have a disparate behaviour if more targets would be present in the area under surveillance. While the computational complexity of the filter depends exponentially on the number of clutter measurements, and this number would obviously increase with the number of targets being tracked, the implementation of an adequate gating strategy, as it has been proposed elsewhere, is able to properly deal with this issue. On the other hand, the processing time of the algorithm, for the full duration of the scenario in analysis, is well below the actual time, thus no real time performance issues are likely to arise.

Chapter 7

Towards LBL systems for multiple vehicles

In this chapter, the use of LBL acoustic networks for operations with multiple AUVs will be discussed. An alternative scheme to standard LBL configurations is proposed that is able to fully address the problem of simultaneous navigation and tracking of multiple vehicles. The solution presented uses the One-Way-Travel-Time of acoustic signals to compute the ranges between all the devices, beacons and vehicles. Moreover, the suitable algorithms for both the navigation of multiple vehicles, but also their external tracking will be derived. In a way, the present chapter is a natural extension to the previous one, as it was motivated by the development of the Multiple AUV Tracker.

7.1 Introduction

The main navigation solutions for current AUVs has been discussed in Section 2. They consist on a combination of Inertial Navigation, with any other methods that are able to bound the drift inherent to the process of dead-reckoning. While other methods exist, the most common way for AUVs to acquire position navigation aids is by resorting to Acoustic Navigation. The use of such techniques is able to provide navigation aids to the vehicles, but they also give the possibility for externally tracking the trajectory of the vehicles. Such functionality can be of extreme importance, as it allows the real-time monitoring of the operations. This is particularly important for operations in dangerous or more demanding scenarios, as it happens in the oil or military industries.

AUVs are already a reliable and cost-effective solution to autonomously perform a range of underwater tasks. Examples for this can be, among others, bathymetric tasks, environmental surveying or even mine countermeasures operations. However, the efficiency of such operations could be even higher if multiple cooperating vehicles are used. Notwithstanding, Acoustic Navigation methods have been envisioned for the single-vehicle scenario and don't have an obvious extension to the multiple vehicle case. In this chapter, an extension to more traditional LBL acoustic navigation schemes is proposed, that is able to address simultaneously the problem of navigation and external tracking of multiple AUVs.

The proposed approach is an extension of traditional LBL methods, and does not require a data communication link between the vehicles and beacons. Both the hardware and the software necessary to implement the desired acoustic network will be discussed. The focus will be on the implementation of an Inverted LBL network for multiple vehicles, using the GPS enabled intelligent buoys presented in Section 2.6.3 as acoustic beacons. Nevertheless, if traditional LBL networks are preferred, with the acoustic beacons deployed in the seabed, the validity of the approach here proposed still holds.

The organization of this chapter is as follows. Section 7.2 provides background information on relevant work found in the literature. Then, in Section 7.3, the proposed Acoustic Network is described, and implementation details are presented. Section 7.4 presents a generic navigation filter to be implemented in the AUVs, that is able to cope with the requirements of the proposed acoustic navigation scheme. Similarly, a central algorithm able to track the moving vehicles is presented in Section 7.5. An example of such algorithm would be the Multiple AUV Tracker, proposed in the previous chapter. Furthermore, an experimental validation of obtained results is provided in section 7.6 and, finally, Section 7.7 concludes this chapter, discussing the proposed approach.

7.2 Background

LBL systems are one of the most robust, reliable and accurate configurations of Acoustic Navigation systems. LBL acoustic networks provide navigational aids to an AUV in terms of ranges to each one of the beacons that compose the network. A few restrictions need to be observed when operating LBL networks. First, the location of each of the beacons needs to be determined prior to operations, in a calibration process that can be sometimes cumbersome. Additionally, the signals emitted by each one of the participants in the network must be uniquely identifiable. This is typically accomplished through the assignment of unique frequency modulated signals to each beacon.

During the missions, the vehicle acoustically interrogates the beacons and measures the round-trip travel-time to each beacon. The time difference between sending the query and receiving the reply, also known as Two Way Travel Time (TWTT), is then converted to the slant range between AUV and a given beacon. These ranges, from the vehicle to each one of the beacons, then pose a spherical navigation constraints to the general problem of relative positioning of the vehicle. By eavesdropping the acoustic signals exchanged between the AUV and each one of the beacons, it is also possible to external track the position of the vehicle. In fact, externally tracking an AUV is a feature that is often sought when dealing with acoustic navigation systems, as it allows real-time monitoring of the position of the vehicle.

7.2.1 Operations with Multiple Vehicles

Scaling up LBL systems for addressing the navigation and external tracking of multiple vehicles has no easy and standard solution. Relatively naive approaches relying on Time Division Multiple Access (TDMA), assigning time slots to each one of the vehicles to interrogate be beacons are not desirable. Doing so would necessarily decrease the rate at which each AUV would receive position updates from the acoustic beacons, leading to unsatisfactory degradation of the system performance. Simultaneously, external tracking features require that the vehicles need to be active, that is, the vehicles need to emit acoustic signals, and the beacons also need to somehow be able to associate each vehicle with the detected signals

Matos and Cruz (2006) suggested an ingenious solution to the problem. This approach consisted on having each of vehicles emitting distinguishable acoustic signals, by using signals modulated with different frequencies. While this approach works, in the sense that it allows both the navigation and tracking of multiple vehicles, it requires the use of different signals for each vehicle and beacon, which naturally comes with a cost. It is not a very scalable solution, and limited to only a few vehicles at most, as the number of distinguishable acoustic signals available for both beacons and vehicles is very limited.

An alternative approach for the problem of simultaneous navigation of multiple AUVs was presented in Baker et al. (2005). There, the authors described a method suitable to a fleet of AUVs, on which a leader vehicle is navigating conventionally with an LBL network. Other vehicles of the fleet should be able to intercept this signals, but also to obtain their relative angular heading relative to the sources of the signals. With this information it is possible for all the vehicles to determine their position relative to their leader. In Kottege and Zimmer (2011) the authors considered a similar problem, by proposing a fully decentralized localization system that enables the operations of a swarm of AUVs. The approach consisted on the use of the developed localization sensor, capable of producing instantaneous estimates of relative azimuth, range, and heading of neighbouring AUVs. It should be noted however that such approach provides only relative position estimates, and requires that all the vehicles to be in close proximity to each other.

Recent advances in underwater acoustic communications have also brought acoustic modems to play a relevant part in underwater navigation capabilities. In fact several authors have proposed different frameworks to enable Cooperative Navigation, for example Fallon et al. (2010). In Cooperative Navigation, teams of AUVs localize themselves more accurately by sharing position estimates and uncertainty. However, such approaches require a data link between the vehicles. While this is a interesting approach, acoustic communications are still characterized by small bandwidth, low data rates and high latency and, particularly for shallow waters and adverse environmental conditions, reliable underwater communications can be quite challenging for long distances.

Traditional LBL systems are based on measuring the Two-Way-Travel-Time (TWTT) of an acoustic signal, on which a vehicle interrogates a set of beacon and waits for their reply. This has obvious disadvantages when using multiple vehicles. In order to overcome this fundamental issue of LBL systems, it has been proposed to use systems based on the One-Way-Travel-Time of acoustic signals. In this kind of systems, the beacons broadcast their acoustic signals at time instants that are know by all the devices operating. Such systems have the obvious advantage of being applicable to the navigation of multiple vehicles. Perhaps one of the initial proposals for OWTT LBL navigation systems was discussed in Eustice et al. (2007). There, the development of a synchronous-clock acoustic communication/navigation system for underwater vehicles that would enable navigation for fleet of AUVs was discussed. Other articles followed focused on the estimation framework for the vehicles that would be able to fuse LBL OWTT ranges with data from other navigation sensors Eustice et al. (2011); Webster et al. (2012, 2013). The main drawback initially pointed out to OWTT based systems was the requirement for the clocks of all the devices, beacons and vehicles, to be synchronized. Because the OWTT measurement is based on the difference between the sender's emission time and reception time on the receiver, it is crucial that the clocks of all the nodes that constitute the network are synchronized to within an acceptable tolerance. While initially OWTT systems meant increased complexity in terms of hardware design of the clock sources, but also in terms on synchronization algorithms, nowadays there are already the technological solutions that can cope, at least partially, with such issues in a fairly straightforward way. Moreover, filtering algorithms have emerged that estimate the unknown offset between different clock sources, thus overcoming an explicit synchronization between vehicles and beacons Batista (2015). Although OWTT techniques do in fact enable the navigation of multiple vehicles, they fail to address the external tracking problem.

7.3 Acoustic Network

From the different acoustic positioning methods, LBL systems stand out mostly due to their ability to provide position solutions with constant accuracy. However, from the previous section, it is clear that no obvious solution to the problem of navigation and external tracking of multiple AUVs has been envisioned yet for these systems. Traditional LBL positioning systems use the TWTT travel times and convert them to ranges to two or more widely spaced stationary beacons, situated whether on the sea bottom or at the surface. However, TWTT based systems are not easily scalable to cope with multiple vehicles. On the other hand, it has been demonstrated that systems based on OWTT can easily provide navigational aids to multiple vehicles by broadcasting their signals on the network (Eustice et al., 2011), but they don't address the external tracking problem. In this sections, an acoustic network scheme able to address the problem under consideration will be derived. The main idea is to combine the strengths of the different acoustic navigation methods proposed in the literature. On top of the acoustic network here described, this solution is also dependent on suitable navigation and tracking algorithms, which will be discussed in the following sections.

The configuration of the acoustic network here considered consists on a set of acoustic beacons deployed in a fixed predefined location, whether at the surface or at the bottom. While usually at least four different beacons are required for a complete three dimensional localization of the vehicles relative to the beacons, under some assumptions fewer beacons are required. In the remainder

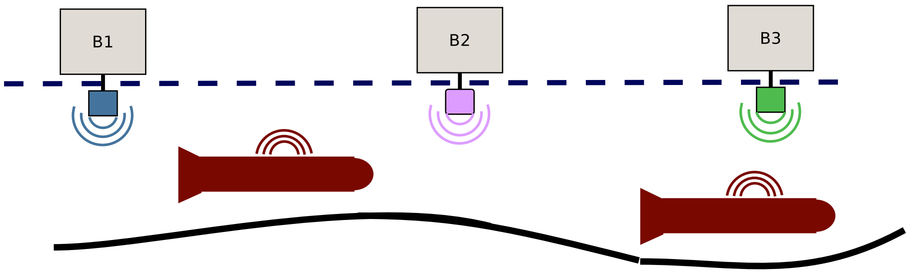


Figure 7.1: Illustration of the proposed acoustic network: beacons emitting different acoustic signals, and vehicle emitting similar acoustic signals

of this chapter only a configuration with two beacons will be considered, but what follows still holds if more beacons are considered. In order to be able to operate with only this small number of beacons, some premises must be verified. First, the baseline length must be long enough when compared to the depths at which the AUVs navigate. Also, vehicles must also be able to independently measure its own depth which relative accuracy. Additionally, it is considered that the vehicles are not allowed to cross the baseline. With this assumptions, in practice we are reducing the position estimation problem of the different vehicles to the horizontal plane only. For more informations regarding this process, refer to (Melo and Matos, 2008) and the references therein.

The network of beacons will provide navigational aids to a fleet of AUVs, that can contain a varying number of vehicles, without any limitation. All the nodes in the network, vehicles and beacons, should have their clock sources synchronized. The idea is that at predefined instants of time all the beacons in the network emit an acoustic signal, to be detected and distinguishable by all the vehicles present. This should be accomplished by having the beacons emitting different frequency modulated signals. Simultaneously, all the vehicles should also emit acoustic signals, but all the vehicles can emit similar ones. Then, a centralized estimation algorithm is used that combines the signal detections of all the beacons, otherwise undistinguishable, and estimates the position of each one of the vehicles. In total, a number of $N+1$ distinguishable acoustic signals is required for this system, where N is the number of beacons in use. It is relevant to note that this number remains constant with the number of vehicles in use.

Figure 7.1 illustrates the proposed topology for the acoustic network. The operation of the network then relies on all the nodes, both beacons and vehicles, converting OWTT of the detected acoustic signals to ranges. Thus, the system draws insights from hyperbolic positioning schemes. Naturally, conversion of times to ranges implies knowledge of the local speed of sound profile.

The rate at which all the nodes are requested to be emitting will limit the area of operations. If, for example, this rate is $1Hz$, then all the nodes will only consider ranges of up to $1500m$, considering that a speed of sound of around $1500ms^{-1}$. Naturally, decreasing this rate allows wider areas of operation but, at the same time, also decreases the rate at which new measurement updates are received by each vehicle. Compared to standard LBL algorithms OWTT systems, like the one here proposed, in general halve this interval of time, which is of obvious interest.

As it can be noted from the description above, this network has similarities with traditional

LBL implementations, in the sense that different beacons must use distinguishable signals, and also because vehicles are required to be active. On the other hand, this network also borrows from OWTT networks, as the interrogation cycle from LBL is replaced by synchronous emission of signals by all the nodes. However, in order for all of this to work, the following assumptions must hold. First all the nodes must remain synchronous during the entire duration of the missions. In other words, this means that duration of the missions is limited to the maximum period during which the drift of all the clock sources remains small enough to be neglected. For the nodes in the network at the surface, the GPS PPS signal can be used as a reference clock signals, allowing long operations without any appreciable drifts. Submerged nodes, on the other hand, should be equipped with high-precision low-drift crystal oscillators, already widely available in the market.

The second assumption stipulates that a locally homogeneous sound speed profile should be observed in the area of operations, which means that the sound velocity should be constant. While this is a moderately mild and common assumption, particularly for shallow water environment, it can be problematic if vehicles are navigating across multiple water layers. Finally, it is also required that the vehicles remain in a confined area, so that they can always be tracked by all the beacons.

All the acoustic nodes are required to have an acoustic system that is able to emit acoustic signals synchronously to an external trigger. At the same time, such acoustic system must also be able to precisely time the detection of signals. From this precise timings, ranges will be obtained by taking into account the speed of sound in the are of operations. It should be noted, however, that these range observations will predictably be noisy, as underwater acoustic propagation is often affected by various physical phenomenons. Therefore, multiple reflections of the same emitted acoustic signal are likely to be observed. This should be taken in consideration by the navigation and tracking algorithms, discussed in the following sections.

7.4 AUV Navigation

AUVs require some sort of Navigation algorithm that is able to perform sensor fusion. They should combine data from different sensors, in order to provide the best possible estimate of the vehicle's position and velocity. The navigation algorithm, which is highly dependent on the vehicle and the sensors used, is usually based on an implementation of the Bayes Filter, as presented in Section 2.5. This section will be devoted to the navigation filter. A simple and generic one, based on an Extended Kalman Filter (EKF), will be used for each of the moving vehicles. Thus, the focus in this section is not on obtaining the best possible navigation filter, but rather only on validating our proposal for the navigation of multiple vehicles, using OWTT range measurements. More advanced filters are likely to provide better results.

The navigation filter is responsible for propagating the vehicle's navigation equations and, when available, include information from the different sensors. In the case under analysis, the TOF of the signals, converted to the slant ranges between the vehicle and each of the acoustic beacons, will be used as indirect observations of the vehicle position and used to bound the error growth of the navigation equations. For the sake of simplicity only the horizontal plane position is considered, even though depth sensors are already reliable enough, and widely available, to provide accurate depth measurements. A relatively simple state vector was chosen, $\mathbf{x}_k = \begin{bmatrix} x_k & v_{x_k} & y_k & v_{y_k} \end{bmatrix}^T$. Even though an augmented state vector would allow to accommodate data from other navigation sensor like IMUs, DVLs, or even model-based pseudo measurements, this wont be further pursued here. The discrete model used to predict the AUV motion is a simple constant velocity model, given by (7.1).

$$\begin{bmatrix} x_{k+1} \\ v_{x,k+1} \\ y_{k+1} \\ v_{y,k+1} \end{bmatrix} = \begin{bmatrix} 1 & \Delta_t & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & \Delta_t \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_k \\ v_{x,k} \\ y_k \\ v_{y,k} \end{bmatrix} \tag{7.1}$$

The received range measurements from each one of the acoustic beacons, z_k are used to correct the Kalman Filter estimates that are propagated using the constant velocity model. The range r_i between vehicle and beacon i is a non-linear function $g(\mathbf{x}_k)$ of the system state:

$$r_i = \sqrt{(x_k - x_{0,i})^2 + (y_k - y_{0,i})^2} \tag{7.2}$$

where $(x_{0,i}, y_{0,i})$ are the position coordinates of beacon i. At time k, the received measurement z_k is the range r_i to any of the beacons that composed the acoustic network, possibly affected by some noise. Then, the measurement z_k is compared with the expected range, z^*, calculated based on the current position estimates for the vehicles, and used in the measurement update step of the filter.

Recalling from previous sections, a high number of outliers are expected to be present in the received range measurements. Therefore, a strategy is implemented to prevent such spurious measurements to corrupt the behaviour of the filter. The first step is to consider, during each period of reception, only the first range provided by each one of the beacons. Only the direct path of the signals is of interest, and the multiple reflections that an acoustic signals might have undergone should be discarded. Nevertheless, some spurious measurements might still occur. An additional validation gate of the received ranges is also implemented according to

$$(z_k - z^*)^T S_k^{-1} (z_k - z^*) \leq \gamma. \tag{7.3}$$

Because the difference $(z_k - z^*)$ is assumed to be Gaussian, the value for γ can be obtained from an appropriate χ^2 distribution table, for the desired confidence level. S_k is the corresponding innovation covariance matrix, obtained as

$$S_k = H_k^T P_k H_k + R_k \tag{7.4}$$

where H_k is the Jacobian of the measurement vector, P_k the error covariance matrix and R_k the covariance matrix of the observation noise.

A final note to the rate at which range measurements are received which is, on average, equal to the number of acoustic beacons per cycle of communications. Compared to standard LBL interrogation cycles, this rate is relatively higher, at least doubling the rate, which can be extremely relevant for the accuracy of the filter.

7.5 AUV Tracker

As mentioned above, the solution presented in this chapter for the simultaneous navigation and tracking of multiple vehicles requires all the nodes to synchronously emit acoustic signals. The signals emitted by the vehicles will be detected by all the beacons, and will enable vehicle tracking. The main difficulty is having to overcome the lack of an explicit association between the acoustic signals detected and the emitting node. Traditional estimators, based on the Bayes Filter, require this explicit association. Alternatively, an additional data-association step is sometimes introduced, but this task can be complex and expensive, particularly when the number of clutter measurements is hight.

Perhaps the biggest novelty in the approach here proposed is the use of a Random Finite Set based estimator to tackle the tracking problem. By using a RFS based approach, an explicit association between acoustic signals and vehicles is no longer required. Thus, it becomes relatively simpler having a centralized filter that is able to track multiple vehicles when the acoustic signals they emit are indistinguishable. Different filter based on random finite sets have been proposed, namely the Probability Hypothesis Density (PHD) Filter, the Cardinalized Probability Hypothesis Density (CPHD) Filter, the Multi-Bernoulli Filter, among others.

Any of the aforementioned filters would be appropriate to address the problem here under analysis. Even though, for the sake of simplicity, the choice will be on using a PHD filter, in a similar way to the previous chapter. The PHD Filter, initially proposed by Mahler (2003), is an approximation of the general multitarget-multisource Bayes Filter. Contrary to standard Bayesian Filtering techniques, such approach uses Random Finite Sets (RFS) to model both the targets states and the measurement equations. The use of RFS overcomes the need for an explicit data association between targets and sources, which is relevant for the current application. On the other hand, the use of RFSs significantly alleviates the effort of addressing situations like target (dis)appearance and spawning, extended targets, false alarms and missed detections.

The Multiple AUV Tracker, derived in the previous chapter, was able to track a varying number of vehicles. Recalling from there, the single target state vector was defined to be $\mathbf{x}_k^i = \begin{bmatrix} x_k^i & y_k^i & \mathtt{l}_k^i \end{bmatrix}$, where x_k^i and y_k^i refer to the position coordinates of the target while $\mathtt{l}_k^i$ is the label used to identify each target. While the use of a label allows for easy track labelling, and simplifies the target estimation process, this is not generally required. Naturally, the index i refers to the target i, while k refers to the time instant. Analogously to the navigation filter of the previous section, a constant velocity model was chosen as the target motion model. However, velocity will also be indirectly estimated subsequently. To do so, estimates of the positions and time will be used as input to a Recursive Least Square (RLS).

The measurement set $\mathbf{Z}_k$ is composed by the set of all the possible vectors $z_k^i = \begin{bmatrix} r_1 & \dots & r_n \end{bmatrix}^T$ that can be obtained from the observed ranges, where r_i is the range measurement to acoustic beacon B_i, as it was made clear in Section 6.5.2. The likelihood of the measurements is considered to be Gaussian. For a large number of vehicles appropriate gating strategies must be derived in order to accommodate the increase in computational complexity.

7.6 Results

In order to experimentally validate our approach, a series of tests were performed. For practical reasons, two independent experiments were performed in order to assess the performance of both the navigation and tracking algorithms separately. Details of these field trials and respective obtained results will be provided in this section.

The use of AUVs poses a problem when it comes to establishing a ground-truth to be able to compare the results with. For that reason, on these trials ASVs were used as AUV surrogates. As discussed in the previous chapter, by doing so it is possible to use the GPS position data of the vehicles as ground-truth to the experiment.

7.6.1 Navigation

The first instance of the field trials was devoted to assess the performance of the acoustic network while two vehicles were navigating simultaneously. Two ASVs and two acoustic beacons were employed. For the sake of simplicity of the missions, one of the ASVs was performing a station keeping manoeuvre, while the other one was moving around. The ranges obtained by each one of the vehicles can be seen in Figure 7.2. Besides the presence of some outliers (rejected ones in black), it can also be seen that Vehicle 1 has been affected by some shortage of range measurements, particularly visible between around the time instant $t = 50s$.

In Figure 7.3 it is possible to observe the trajectories of the vehicles, together with the position of the buoys. It is clear that even thought the trajectories obtained by the navigation filter do resemble the ground-truth data, there are still significant differences. Causes for that are probably the shortage of ranges that affected mainly Vehicle 1, together with the very simple motion model in use. It should be noted that no additional inertial sensor measurement has been fused.

Figure 7.4 provides a more detailed view on the position of the different vehicles along time, and compares it to the ground-truth data. There its more easily observed the effect that the lack of ranges have caused in the estimation of the trajectory of the vehicle in blue. Due to the assumed constant-velocity model, when no ranges are received, the filter just propagates the position of the vehicle according to the most up to date information. However, when new measurements arrive, after some time, the filter corrects previous estimates, with new positions getting close to ground-truth.

Finally, Figure 7.5 shows the plot of the absolute error in horizontal position along the time. A peak in position error for one of the vehicles can be observed, at around $t = 50s$, but shortly after

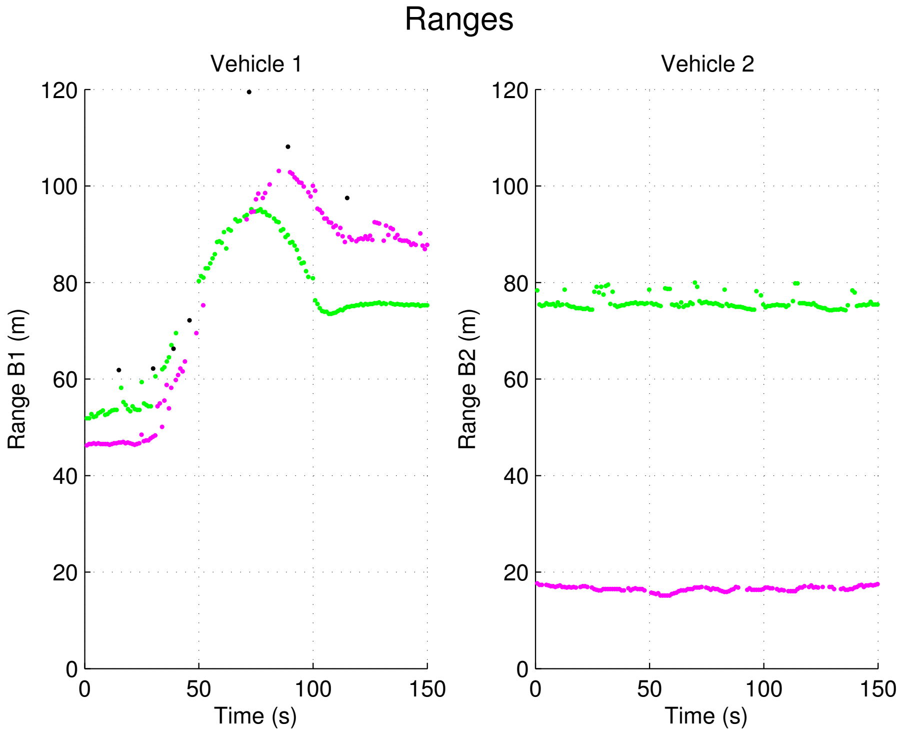


Figure 7.2: Ranges detected by each one of the vehicles. Green colour indicates ranges to B_1 and purple colour ranges to B_2

decreasing to less than $6m$. Based on that data, the obtained RMS error for each of the vehicles was of $4.1m$ and $2.8m$ respectively. Given the circumstances, this can be considered very encouraging results.

7.6.2 Tracking

The second goal of of the field trials was to assess the tracking results of the a centralized tracking algorithm. Therefore, the results here presented are based on the previous chapter. There, the Multiple AUV Tracker was derived, and its tracking performances were evaluated. The different parameters of the tracker used to obtain the results that are going to be presented are specified in Table 7.1.

The mission consisted on having two vehicles navigating simultaneously in a predefined area, under surveillance of two moored acoustic buoys. The vehicles started to emit acoustic signals at different times, in order to illustrate the ability of the filter to detect new vehicles entering the surveillance area. At the same time, the vehicles were navigating with arbitrary varying velocities and in different directions, including stationary situations. An overview of the trajectories performed by each of the vehicles can be seen in Figure 7.6. Figure 7.7 provides a more detailed view on the position of the different vehicles along time, and comparing it to the ground-truth data. As previously detailed, the obtained results were very positive, successfully demonstrating the tracking abilities of the proposed filter.

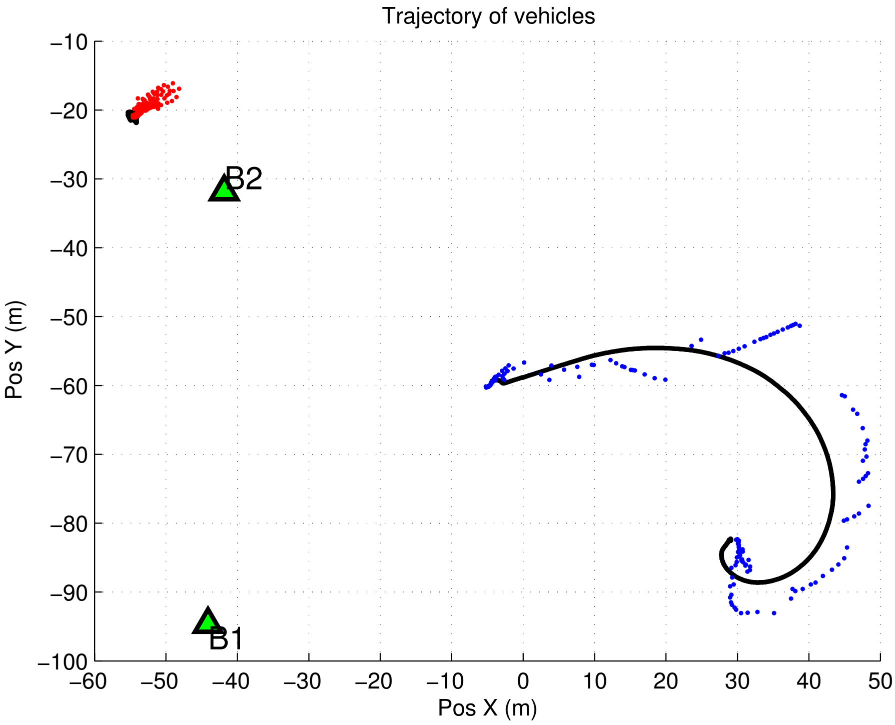


Figure 7.3: Navigation Results: trajectories of the vehicles. In black ground-truth given by the GPS and in blue and red estimated by the navigation filter

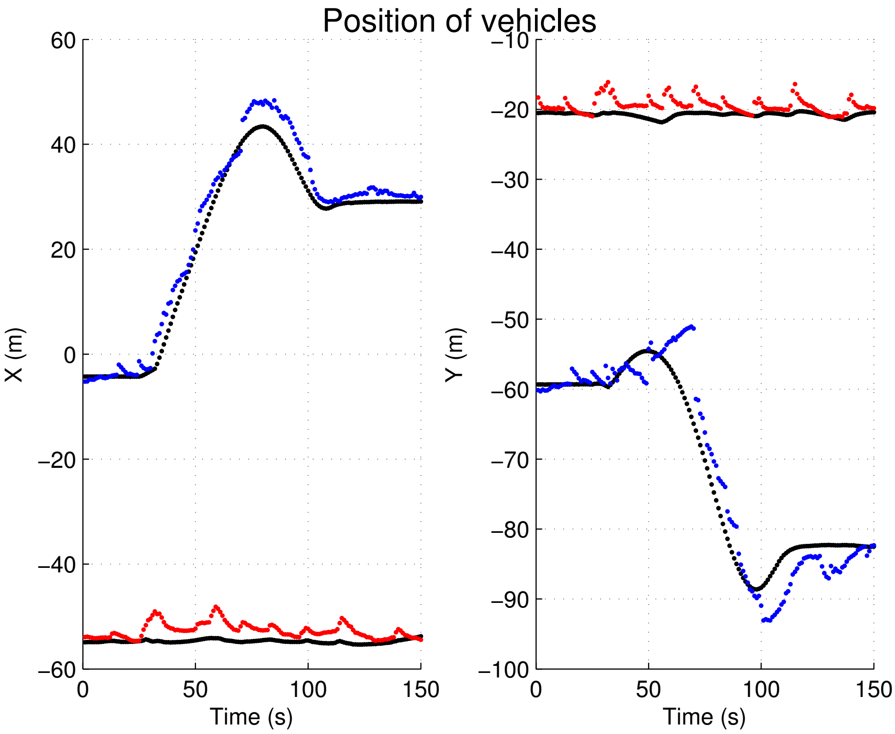


Figure 7.4: Navigation Results: evolution of the position of the vehicles. In black ground-truth given by the GPS and in blue and red estimated by the navigation filter

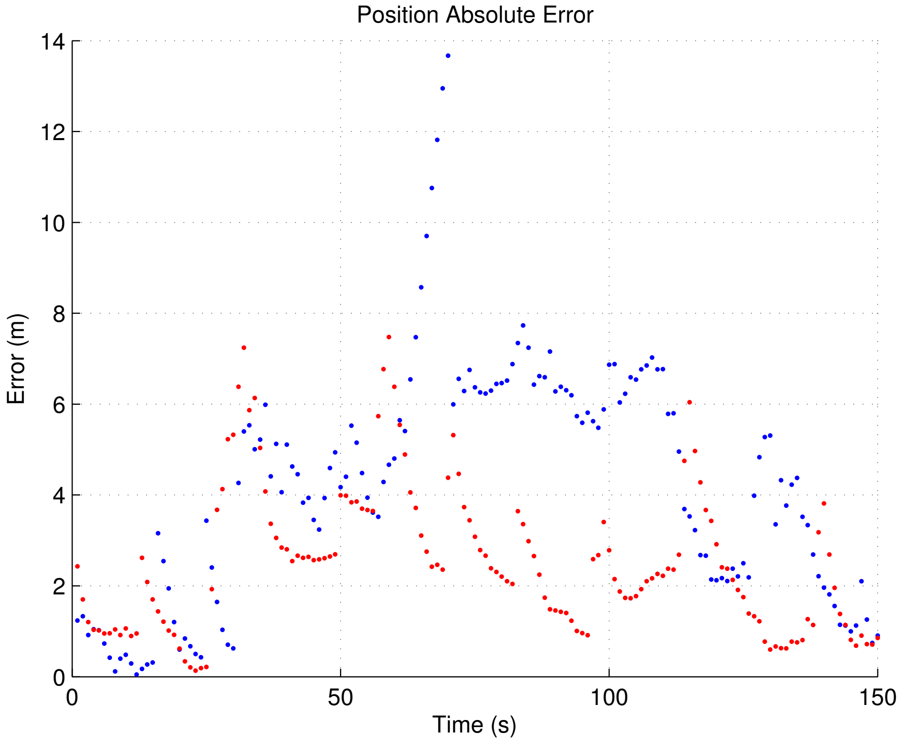


Figure 7.5: Navigation Results: absolute position error for each of the vehicles, when comparing to ground truth given by GPS.

Parameter	**Value**
Filter Settings	
Particles Per Target (N)	1000
Particles Per Birth (M)	1000
Predictor Settings	
Prob. of Survival (p_s)	0.99
Prob. of Birth (p_b)	0.01
Process Noise Variance ($\sigma^2_{x,y}$)	0.1
Corrector Settings	
Prob. of Detection (p_d)	0.6
Measurement Noise Variance ($\sigma^2_{r,B}$)	0.8
Clutter Intensity (λ_k)	$5 \times 10^{-5}\sqrt{\lvert\mathbf{Z}_k\rvert})$

Table 7.1: Parameters of the Multiple AUV Tracker

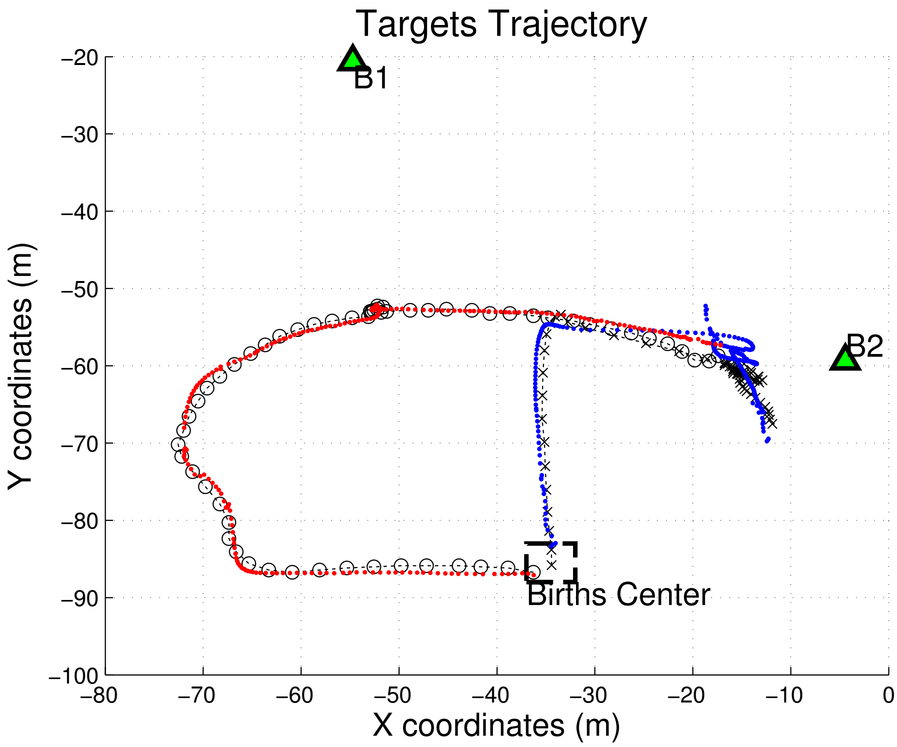


Figure 7.6: Tracking results: trajectories of the vehicles. In black ground-truth given by the GPS and in blue and red estimated by the centralized tracking filter.

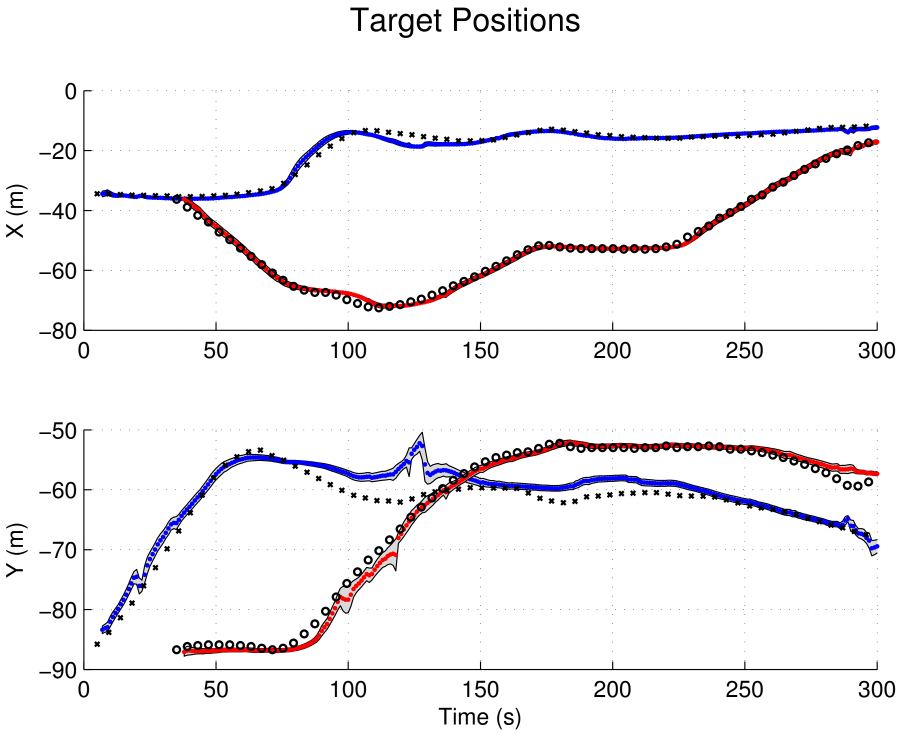


Figure 7.7: Tracking Results: evolution of the position of the vehicles. In black ground-truth given by the GPS and in blue and red estimated by the centralized tracking filter.

7.7 Conclusion

In this chapter the feasibility of a novel LBL network, that enables the navigation and external tracking of multiple vehicles, was studied. The proposed approach is composed out of three main components: the acoustic network itself, the navigation filters for each one of the vehicles, and the centralized tracker for multiple AUVs. Up to the authors knowledge, no similar approach has been previously proposed in the literature. The acoustic network is based on the use of a set of synchronized acoustic nodes, both vehicles and beacons. OWTT techniques are used to compute acoustic ranges.

Synchronously, all the nodes broadcast its own acoustic signals. The signals emitted by the beacons will provide navigation aids to a fleet of AUVs, while at the same time signals emitted by the vehicles will be used for tracking purposes. On the vehicle side a navigation filter based on EKF is responsible for fusing the selected range measurements with the general navigation solution of the vehicle. Finally, a centralized multi-vehicle tracker, collects the signals that all the beacons are able to detect, and is then responsible to estimate the position of each the vehicles. In here, this centralized filter was based on the Multiple AUV Tracker presented in the previous chapter.

Experimental evidence of the proposed approach success was provided. Even though the two sets of trial were performed independently, the provided results are very encouraging for the simultaneous experimental validation necessary. Nevertheless, future work will also include robustifiyng the proposed solution, for example by developing appropriate measurement gating strategies.

Part IV

Conclusions

Chapter 8

Conclusions

In this thesis different topics were considered, all related with the Navigation of Sensor-limited AUVs. Briefly, three main topics were covered, Bottom Following, Terrain Based Navigation, and acoustic Navigation and Tracking of multiple vehicles. Common to all these topics is the use of Bayesian Estimation techniques as a tool to address such problems. This chapter, that concludes the thesis, first summarizes the research undertaken in this thesis, in Section 8.1. Then Section 8.2 lists the main contributions, and Section 8.3. presents some suggestions for future research.

8.1 Summary

Sensor-limited AUVs were defined as being a class of vehicles characterized by being equipped with only a limited set of sensors, namely low-accuracy navigation sensors and low information range sensors. Due to the characteristics their sensors, vehicles falling in this class face two main difficulties. First, their navigation capabilities, that is estimating their own position and velocity, are low. Second, their ability to perceive the surrounding environment is limited, due to the low level of information that can be simultaneously acquired.

This thesis is divided in two distinct but complementary parts. This organization reflects the two topics that have been identified as fundamental to further improve the use and the possible applications for Sensor-limited AUVs. The first part, composed of Chapters 3, 4 and 5, is devoted to the development of Navigation algorithms based on natural features of the environment. By using information from the environment, in specific information from the terrain, the autonomy of these vehicles can be extended. The second part of this thesis, composed of Chapters 6 and 7, is devoted to the development of algorithms that can extend current Acoustic Navigation systems so they can support the simultaneous navigation and tracking of multiple vehicles. The general trend in robotics, and also for underwater robotics, goes towards the simultaneous use of vehicles, cooperating in a coordinated way for an efficient completion of a common goal. Therefore, the development of Acoustic Navigation algorithms that can cope with various vehicles is very relevant.

8.2 Main Contributions

The core of this thesis are Chapter 3 to 7. This section lists the main contributions of each chapter.

- The main contribution of Chapter 3 is the development of a Bottom Following behaviour, enabling the vehicle to follow a trajectory always parallel to the bottom. A sensor-based guidance approach was derived, that was able to estimate the depth to the bottom and the slope of the bottom using range measurements from a single-beam sonar. Additionally, this strategy was experimentally validated, using the MARES AUV, providing evidence of its success. Despite that, in some situations minor oscillations were observed with an identified cause. They were induced by the vertical controllers of the vehicle, that privileged the speed of the response rather than overshooting.

- The main contribution of Chapter 4 is the development of a MIMO Depth-Pitch controller for the MARES AUV. This contribution was motivated by the results obtained in the previous chapter. By using Eigenstructure Assignment techniques it was possible to fully specify the desired output time-response of the system. Thus, an appropriate Eigenstructure of the system was selected in order to avoid overshoot, ensuring a safe trajectory for Bottom Following missions.

- Chapter 5 addressed Terrain Based Navigation algorithms for Sensor-limited systems. Particle Filter implementations have been demonstrated to achieve the best results for Terrain Based Navigation for such kind of systems. The main contribution that stems from this chapter is the development of a Data-Driven Particle Filter. By learning a proposal distribution, the DD-PF was demonstrated to match and even outperform the SIR-PF, that has been widely used in the context of TBN for underwater vehicles. The DD-PF was also demonstrated to be more efficient than the SIR-PF, as it requires a lot less resampling iterations.

- The main contribution of Chapter 6 is the development of an algorithm capable of acoustically track multiple AUVs. In applications using multiple vehicles, it is sometimes desirable to externally track the vehicles in operations. The algorithm derived, the Multiple AUV Tracker, is based on a Probability Hypothesis Density filter, and is able to track multiple vehicles even if they are emitting otherwise undistinguishable acoustic signals. It should be noted that the tracker is able not only to estimate the position of multiple targets, but also their trajectory, as labels are used to uniquely identify every target. Experimental validation of the algorithm was performed, with mean absolute errors in terms of position comparable to GPS.

- The last contribution of this thesis was presented in Chapter 7. Making use of the Multiple AUV Tracker introduced in Chapter 6, an LBL Acoustic Network was proposed that is compatible with the simultaneous navigation and tracking of multiple vehicles. The proposed LBL system requires clock synchronization between all the nodes, so that ranges can be obtained from the OWTT. Additionally, all the vehicles require a suitable Navigation filter, and a Tracking algorithm is also required. Both these algorithms were detailed, and experimentally validated.

8.3 Future Research Directions

Navigation for Sensor-limited AUVs is a very challenging and dynamic subject. The research conducted throughout this thesis lead to some very explicit and particular results. However, there is a lot of unexplored areas to address in the future, that can extend what has been presented here.

The topic of Bottom Following is perhaps the more circumscribed topic from this thesis. Nevertheless, there is plenty of room for improvements. In the approach here proposed, a single beam sonar was used, mounted on a downward-facing configuration. If no others, this was done mostly for practical reason. However, different geometric configurations in the mounting of the sonar could be exploited, leading to more interesting results. For example if the sonar is mounted with a given angle towards the front of the vehicle, some prediction of the profile of the bottom ahead of the vehicle could be achieved. Another challenging problem would be the conjugation of the altimeter used with other sonar, for example other single beam sonars, mounted in a favourable position. Different configurations could be used to achieve a higher degree of perception of the environment. Combining different types of sonars, for example an altimeter with a sidescan sonar, though not straightforward could lead to potentially very compelling results.

Terrain Based Navigation for Sensor-limited systems was the second topic addressed in this thesis. In specific, Particle Filter based algorithms were studied. Despite the good results in terms of estimation of the position, it is known that Particle Filters can diverge from the true position. The performance of Particle Filters is in fact dependent on the choice of an accurate proposal distribution which, in general, is assumed to be the vehicle's motion model. However, this distribution can be different from the optimal proposal distribution. This is even more striking for the case of Sensor-limited systems, where the noise levels of the sensors are often large and incorrectly modelled. Future work in this area could, for instance, involve applying adaptive online learning algorithms that could better approximate the optimal proposal distribution. Resorting to typical Machine Learning algorithms, like Sparse Gaussian Processes, could potentially be a favourable research direction.

The use of a PHD filter for acoustically tracking multiple AUVs was the last topic of this dissertation, and has proved to be very effective. A direct outcome of such algorithm is the LBL network proposed also here in this thesis. The LBL network, detailed in Chapter 7, is able to provide simultaneous navigation and tracking functionality for multiple vehicles. Besides the somehow more direct extensions that directly apply to what was presented, for example introducing gating strategies for the range measurements, future research in this area can be easily foreseen. In the LBL network proposed, each of the different acoustic beacons emits a signal that must be easily distinguishable from what is used by the remaining beacons. An obvious future research direction would be trying to relax this requirement, analogously to what was proposed for the tracking case. Other possible research direction would be the development of the necessary algorithms that allow the vehicles themselves to track other vehicles, using the range-only or range and bearing measurements. This possibility would definitely increase the situational awareness of the vehicles in an unprecedented way.

Appendix A

Estimation of Speed of Sound

Having an accurate knowledge about the speed of sound is of utmost importance for Acoustic Navigation. In Acoustic Navigation methods, and in particular in LBL methods, the time-of-flight of the acoustic signals is transformed into ranges by taking into account the local speed of sound. However the speed of sound is known to vary with various parameters, like temperature, salinity or even depth. Therefore, in order to optimize the results of acoustic navigation algorithms, speed of sound should always be determined.

This Appendix outlines an experimental method to btain local estimates of the speed of sound, using data from different acoustic nodes. While there are accurate models for the speed of sound, they require knowledge about different properties of the environment. And if depth and temperature are relatively easy and cheap to measure, the same does not apply to salinity. Notwithstanding, it is often the case when it is not possible to measure such quantities in the operations scenario.

By assuming that the speed of sound is locally homogeneous, it is immediate that the range between any two nodes should be proportional to the ToF of an acoustic signal emitted by one of the nodes, and detected by the other. Naturally, the proportionality constant between this ToF and range will be the local speed of sound. If the range and ToF can be independently measured between nodes of an acoustic network, then the value of the speed of sound can be experimentally determined.

Recalling from Chapters 6 and 7, all the nodes are equipped with both GPS and Acoustic Navigation system. In order to estimate a value for the speed of sound in a configuration like this, the ranges between each two pairs of nodes must be compared with the respective ToF of an acoustic signal between each of the nodes. Then, the speed of sound can be obtained by using a Least Square Estimator to fit the data with a line. The slope of the line that best fits the data will provide the optimal value for the speed of sound.

Figure A.1 shows the results of such a process. It refers to an LBL configuration as described in Chapters 6 and 7, consisting of two moored acoustic beacons at the surface, and two moving vehicles. GPS data was collected for all the nodes. The vehicles were also synchronously emitting acoustic signals, that were detected by each one of the beacons. The collected data is plotted in Figure A.1. The blue points correspond to ToF of acoustic signals plotted against the corresponding GPS distance between the two nodes.

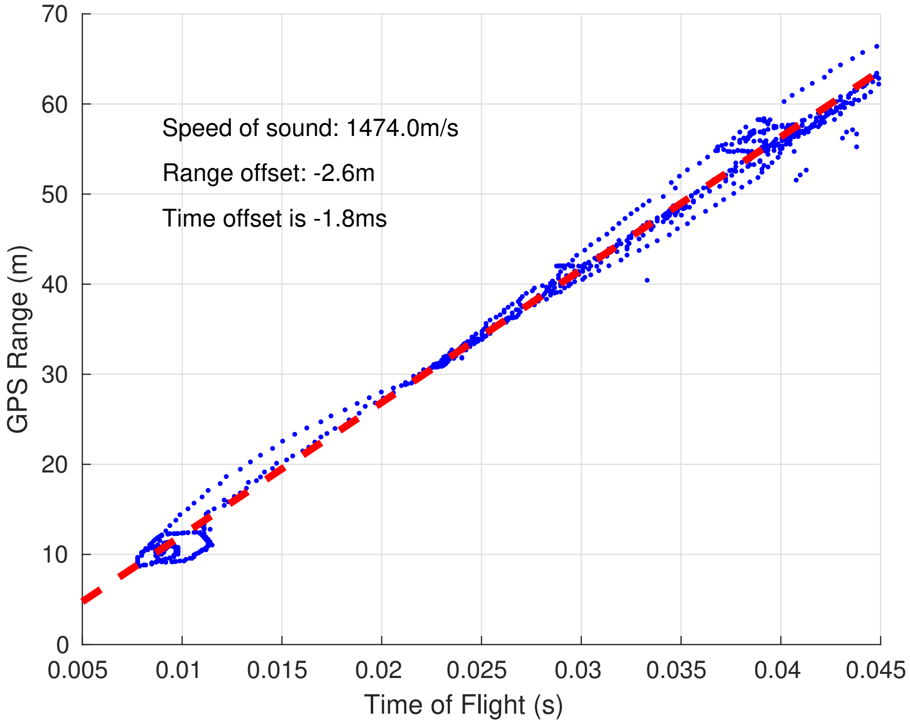


Figure A.1: Determination of the speed of sound. In blue, experimental data collected, in red least-squares fit of the data.

It can be observed in Figure A.1 that the red line fits the data points relatively well, leading to a value of $1474 m/s$ for the speed of sound. It should also be noted that this procedure also allows to estimate any systematic errors affecting the system. If this is the case, the line fitting the data will not cross the origin. In the specific case of our Acoustic System, this corresponds to the delay introduced by the filtering electronics in the reception of the signals. For our system, this systematic errors corresponding to range offset error of $2.6m$, which then corresponds to a delay in the acoustic system of around $1.8ms$.

While the determined value for the speed of sound is valid, it should be noted that it was obtained used GPS measurements, which are always affected by some errors. Therefore, the accuracy of the obtained value for the speed of sound is also affected by this errors. In order to increase the accuracy of speed of sound estimation, alternative ways to measure the distance between the nodes should be used, for example using differential or real-time kinematic GPS receivers. A more detailed discussion on obtaining an accurate speed of sound, and other error affecting LBL systems, can be found in Almeida et al. (2016b).

References

Adhami-Mirhosseini, A., Aguiar, A. P., and Yazdanpanah, M. J. (2011). Seabed tracking of an autonomous underwater vehicle with nonlinear output regulation. In *50th IEEE Conference on Decision and Control and European Control Conference*, pages 3928–3933.

Ahamada, I., Flachaire, E., and Clark, A. (2010). *Non-Parametric Econometrics*. Practical Econometrics. OUP Oxford.

Alexander, J., Cheng, Y., Zheng, W., Trawny, N., and Johnson, A. (2012). A Terrain Relative Navigation sensor enabled by multi-core processing. In *IEEE Aerospace Conference*, pages 1–11.

Almeida, R., Cruz, N., and Matos, A. (2016a). Man Portable Acoustic Navigation Buoys. In *Proceedings of of MTS/IEEE Oceans'16 Conference*, pages 1–6, Shanghai, China.

Almeida, R., Melo, J., and Cruz, N. (2016b). Characterization of Measurement Errors in a LBL Positioning System. In *Proceedings of of the MTS/IEEE Oceans'16 Conference*, pages 1–6, Shanghai, China.

Andry, A. N., Shapiro, E. Y., and Chung, J. C. (1983). Eigenstructure Assignment for Linear Systems. *IEEE Transactions on Aerospace and Electronic Systems*, AES-19(5):711–729.

Ånonsen, K. B. (2010). *Advances in Terrain Aided Navigation for Underwater Vehicles*. PhD thesis, Norwegian University of Science and Technology.

Anonsen, K. B. and Hagen, O. K. (2011). Recent developments in the HUGIN AUV terrain navigation system. In *Proceedings of of the MTS/IEEE Oceans'11 Conference*, pages 1–7, Kona, Hawai, USA.

Anonsen, K. B. and Hallingstad, O. (2006). Terrain Aided Underwater Navigation Using Point Mass and Particle Filters. In *IEEE/ION Position, Location, And Navigation Symposium*, pages 1027–1035. IEEE.

Anonsen, K. B., Hallingstad, O., and Hagen, O. K. (2007). Bayesian Terrain-Based Underwater Navigation Using an Improved State-Space Model. In *Symposium on Underwater Technology and Workshop on Scientific Use of Submarine Cables and Related Technologies*, number April, pages 499–505. IEEE.

Arulampalam, M., Maskell, S., Gordon, N., and Clapp, T. (2002). A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking. *IEEE Transactions on Signal Processing*, 50(2):174–188.

Bahr, A., Leonard, J. J., and Fallon, M. F. (2009). Cooperative Localization for Autonomous Underwater Vehicles. *The International Journal of Robotics Research*, 28(6):714–728.

Baker, B. N., Odell, D. L., Anderson, M. J., Bean, T. A., and Edwards, D. B. (2005). A New Procedure for Simultaneous Navigation of Multiple AUVs. In *Proceedings of MTS/IEEE Oceans'05 Conference*, pages 1–4, Washington, D.C, USA.

Bar-Shalom, Y. and Li, X.-R. (1995). *Multitarget-multisensor Tracking: Principles and Techniques*. Yaakov Bar-Shalom.

Batista, P. (2015). Long baseline navigation with clock offset estimation and discrete-time measurements. *Control Engineering Practice*, 35:43–53.

Bennett, A. A., Leonard, J. J., and Bellingham, J. G. (1995). Bottom following for survey-class autonomous underwater vehicles. *International Symposium on Unmanned Untethered Submersible Technology*, 1:327–336.

Bergem, O. (1993). *Bathymetric Navigation of Autonomous Underwater Vehicles using a Multibeam Sonar and a Kalman Filter with Relative Measurent Covariance Matrices*. PhD thesis, University of Trondheim.

Bergman, N. (1999). *Recursive Bayesian Estimation: Navigation and Tracking Applications*. PhD thesis, Linkoping University, Sweden.

Braca, P., Goldhahn, R., LePage, K. D., Marano, S., Matta, V., and Willett, P. (2014). Cognitive multistatic AUV networks. In *17th International Conference on Information Fusion*, pages 1–7.

Caccia, M., Bruzzone, G., and Veruggio, G. (1999). Active sonar-based bottom-following for unmanned underwater vehicles. *Control Engineering Practice*, 7(4):459–468.

Caccia, M., Bruzzone, G., and Veruggio, G. (2003). Bottom-Following for Remotely Operated Vehicles: Algorithms and Experiments. *Autonomous Robots*, 14(1):17–32.

Caccia, M., Veruggio, G., Casalino, G., Alloisio, S., Grosso, C., and Cristi, R. (1997). Sonar-based bottom estimation in UUVs adopting a multi-hypothesis extended Kalman filter. In *Proceedings of the 8th International Conference on Advanced Robotics*, pages 745–750.

Chen, Z. (2003). Bayesian Filtering: From Kalman Filters to Particle Filters, and Beyond. Technical report, Communications Research Laboratory, McMaster University, Hamilton, Ontario, Canada.

Choi, J. and Choi, H.-T. (2015). Multi-target localization of underwater acoustic sources based on probabilistic estimation of direction angle. In *Proceedings of of the MTS/IEEE Oceans'15 Conference*, pages 1–6, Genova, Italy.

Clark, D. E. (2006). *Multiple Target Tracking with The Probability Hypothesis Density Filter*. PhD thesis, Heriot-Watt University.

Clark, D. E. and Bell, J. (2007). Multi-target state estimation and track continuity for the particle PHD filter. *IEEE Transactions on Aerospace and Electronic Systems*, 43(4):1441–1453.

Cowie, M., Wilkinson, N., and Powlesland, R. (2008). Latest development of the TERPROM Digital Terrain System (DTS). In *IEEE/ION Position, Location and Navigation Symposium*, pages 1219–1229.

Cruz, N., Matos, A., Cunha, S., and Silva, S. (2007). Zarco - An autonomous craft for underwater surveys. In *Proceedings of the 7th Geomatic Week, Barcelona, Spain*.

Cruz, N. A. and Matos, A. C. (2008). The MARES AUV, a Modular Autonomous Robot for Environment Sampling. In *Proceedings of of the MTS/IEEE Oceans'08 Conference*, pages 1–6, Quebec City, Canada.

de Freitas, J. F. G., Niranjan, M., Gee, A. H., and Doucet, A. (2000). Sequential Monte Carlo Methods to Train Neural Network Models. *Neural Computation*, 12(4):955–993.

Dektor, S. and Rock, S. (2012). Improving robustness of terrain-relative navigation for AUVs in regions with flat terrain. In *IEEE/OES Autonomous Underwater Vehicles (AUV)*, pages 1–7.

Dektor, S. and Rock, S. (2014). Robust Adaptive Terrain-Relative Navigation. In *Proceedings of of the MTS/IEEE Oceans'14 Conference*, St. John's, Canada.

Devroye, L. and Györfi, L. (1985). *Nonparametric density estimation: the L1 view*. Wiley series in probability and mathematical statistics. Wiley.

Donovan, G. T. (2011). Development and testing of a real-time terrain navigation method for AUVs. In *Proceedings of of the MTS/IEEE Oceans'11 Conference*, pages 1–9, Kona, Hawai, USA. IEEE.

Donovan, G. T. (2012). Position Error Correction for an Autonomous Underwater Vehicle Inertial Navigation System (INS) Using a Particle Filter. *IEEE Journal of Oceanic Engineering*, 37(3):431–445.

Doucet, A., Godsill, S., and Andrieu, C. (2000). On sequential Monte Carlo sampling methods for Bayesian filtering. *Statistics and Computing*, 10(3):197–208.

Draper, C. S., Wrigley, W., Hoag, D. G., Battin, R. H., Miller, J. E., Koso, D. A., Hopkins, A. K., and Velde, W. E. V. (1965). Space Navigation Guidance and Control. Technical report, NATO's Advisory Group for Aerospace Research and Development, Cambridge, MA, USA.

Erdinc, O., Willett, P., and Coraluppi, S. (2008). The Gaussian Mixture Cardinalized PHD tracker on MSTWG and SEABAR datasets. In *11th International Conference on Information Fusion*, pages 1–8.

Eustice, R. M., Singh, H., and Whitcomb, L. L. (2011). Synchronous-clock one-way-travel-time acoustic navigation for underwater vehicles. *Journal of Field Robotics*, 28(1):121–136.

Eustice, R. M., Whitcomb, L. L., Singh, H., and Grund, M. (2007). Experimental Results in Synchronous-Clock One-Way-Travel-Time Acoustic Navigation for Autonomous Underwater Vehicles. In *IEEE International Conference on Robotics and Automation*, pages 4257–4264.

Faleiro, L. F. (1998). *The Application of Eigenstructure Assignment to the Design of Flight Control Systems*. PhD thesis, Loughborough University.

Fallon, M. F., Papadopoulos, G., and Leonard, J. J. (2010). A measurement distribution framework for cooperative navigation using multiple AUVs. In *IEEE International Conference on Robotics and Automation*, pages 4256–4263.

Fearnhead, P. (1998). *Sequential Monte Carlo methods in filter theory*. PhD thesis, Merton College, University of Oxford.

Ferreira, B., Matos, A., and Cruz, N. (2010a). Single beacon navigation: Localization and control of the MARES AUV. In *Proceedings of of MTS/IEEE Oceans'10 Conference*, pages 1–9, Seattle, USA. IE.

Ferreira, B., Matos, A., Cruz, N., and Pinto, M. (2010b). Modeling and control of the MARES autonomous underwater vehicle. *Marine Technology Society Journal*, 44(2):19–36.

Fiorelli, E., Leonard, N. E., Bhatta, P., Paley, D. A., Bachmayer, R., and Fratantoni, D. M. (2006). Multi-AUV Control and Adaptive Sampling in Monterey Bay. *IEEE Journal of Oceanic Engineering*, 31(4):935–948.

Fossen, T. I. (1994). *Guidance and control of ocean vehicles*. John Wiley & Sons, Ltd.

Fox, D. (2001). KLD-Sampling: Adaptive Particle Filters. In *In Advances in Neural Information Processing Systems 14*, pages 713–720. MIT Press.

Fulton, T. F., Cassidy, C. J., Stokey, R. G., and Leonard, J. J. (2000). Navigation sensor data fusion for the {AUV REMUS}. In *Proceedings of the Symposium on Underwater Robotic Technology, World Automation Congress*, Hawaii.

Gao, J., Xu, D., Zhao, N., and Yan, W. (2008). A potential field method for bottom navigation of autonomous underwater vehicles. In *7th World Congress on Intelligent Control and Automation*, pages 7466–7470.

Georgescu, R., Schoenecker, S., and Willett, P. (2009). GM-CPHD and MLPDA applied to the SEABAR07 and TNO-blind multi-static sonar data. In *12th International Conference on Information Fusion*, pages 1851–1858.

Georgy, J. and Noureldin, A. (2014). Unconstrained underwater multi-target tracking in passive sonar systems using two-stage PF-based technique. *International Journal of Systems Science*, 45(3):439–455.

Georgy, J., Noureldin, A., and Mellema, G. R. (2012). Clustered Mixture Particle Filter for Underwater Multitarget Tracking in Multistatic Active Sonobuoy Systems. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, 42(4):547–560.

Gordon, N. J., Salmond, D. J., and Smith, A. F. M. (1993). Novel approach to nonlinear/non-Gaussian Bayesian state estimation. *IEE Proceedings F - Radar and Signal Processing*, 140(2):107–113.

Granstrom, K., Lundquist, C., Gustafsson, F., and Orguner, U. (2014). Random Set Methods: Estimation of Multiple Extended Objects. *IEEE Robotics Automation Magazine*, 21(2):73–82.

Groves, P. D. (2013). *Principles of GNSS, Inertial, and Multisensor Integrated Navigation Systems*. GNSS/GPS. Artech House, second edi edition.

Gustafsson, F. (2010). Particle filter theory and practice with positioning applications. *IEEE Aerospace and Electronic Systems Magazine*, 25(7):53–82.

Hagen, O. K., Anonsen, K. B., and Saebo, T. O. (2011). Low altitude AUV terrain navigation using an interferometric sidescan sonar. In *Proceedings of of the MTS/IEEE Oceans'11 Conference*, pages 1–8, Santander, Spain.

Hagen, P. E., Hegrenæs, Ø., Jalving, B., øivind Midtgaard, Wiig, M., and Hagen, O. K. (2009). Making AUVs Truly Autonomous. In (Ed.), A. V. I., editor, *Underwater Vehicles*. InTech.

Hol, J. D., Schon, T. B., and Gustafsson, F. (2006). On Resampling Algorithms for Particle Filters. In *IEEE Nonlinear Statistical Signal Processing Workshop*, pages 79–82.

Hostetler, L. (1978). Optimal terrain-aided navigation systems. In *Guidance and Control Conference*.

Houts, S. E., Rock, S. M., and McEwen, R. (2012). Aggressive terrain following for motion-constrained AUVs. In *IEEE/OES Autonomous Underwater Vehicles*, pages 1–7.

Jircitano, A., White, J., and Dosch, D. (1990). Gravity based navigation of AUVs. In *Proceedings of the Symposium on Autonomous Underwater Vehicle Technology*, pages 177–180.

Johnson, A. E. and Montgomery, J. F. (2008). Overview of Terrain Relative Navigation Approaches for Precise Lunar Landing. In *IEEE Aerospace Conference*, pages 1–10.

Ju, B., Zhang, Z., and Zhu, J. (2010). A novel proposal distribution for particle filter. In *3rd International Congress on Image and Signal Processing*, volume 7, pages 3120–3124.

KARABORK, A. (2010). Terrain Aided Navigation. Master's thesis, Middle East Technical University.

Kendoul, F. (2012). Survey of advances in guidance, navigation, and control of unmanned rotorcraft systems. *Journal of Field Robotics*, 29(2):315–378.

Kim, T. and Kim, J. (2014). Nonlinear filtering for terrain-referenced underwater navigation with an acoustic altimeter. In *Proceedings of of the MTS/IEEE Oceans'14 Conference*, pages 1–6, Taipei, Taiwan.

Kinsey, J. C., Eustice, R., and Whitcomb, L. L. (2006). A Survey of Underwater Vehicle Navigation: Recent Advances and New Challenges. In *7th Conference on Manoeuvring and Control of Marine Craft (MCMC'2006)*, pages 1–12, Lisbon, Portugal.

Kottege, N. and Zimmer, U. R. (2011). Underwater acoustic localization for small submersibles. *Journal of Field Robotics*, 28(1):40–69.

Kreucher, C. and Shapo, B. (2011). Multitarget Detection and Tracking Using Multisensor Passive Acoustic Data. *IEEE Journal of Oceanic Engineering*, 36(2):205–218.

Lang, T., Dunne, D., and Mellema, G. (2009). An assessment of hierarchical data fusion using SEABAR'07 data. In *12th International Conference on Information Fusion*, pages 1560–1567.

LaPointe, C. E. G. (2006). *Virtual Long Baseline (VLBL) autonomous underwater vehicle navigation using a single transponder*. PhD thesis, Massachusetts Institute of Technology.

Larsen, M. B. (2000). Synthetic long baseline navigation of underwater vehicles. In *Proceedings of of MTS/IEEE Oceans'00 Conference*, volume 3, pages 2043–2050 vol.3, Providence, USA. IEE.

Lee, J., Kwon, J. H., and Yu, M. (2015). Performance Evaluation and Requirements Assessment for Gravity Gradient Referenced Navigation. *Sensors*, 15(7):16833–16847.

Leonard, J. J., Bennett, A. A., Smith, C. M., and Feder, H. J. S. (1998). Autonomous Underwater Vehicle Navigation. In *MIT Marine Robotics Laboratory Technical Memorandum*, pages 1–17, Cambridge, MA, USA. Dept. of Ocean Engineering, Massachussets Institute of Technology.

Leonard, N. E., Paley, D. A., Davis, R. E., Fratantoni, D. M., Lekien, F., and Zhang, F. (2010). Coordinated control of an underwater glider fleet in an adaptive ocean sampling field experiment in Monterey Bay. *Journal of Field Robotics*, 27(6):718–740.

Lin, L., Bar-Shalom, Y., and Kirubarajan, T. (2006). Track labeling and PHD filter for multitarget tracking. *IEEE Transactions on Aerospace and Electronic Systems*, 42(3):778–795.

Liu, F., Qian, D., Liu, F., and Li, Y. (2009). Integrated Navigation System Based on Correlation between Gravity Gradient and Terrain. In *International Joint Conference on Computational Sciences and Optimization, 2009*, volume 2, pages 289–293.

Ma, W.-K., Vo, B.-N., Singh, S. S., and Baddeley, A. (2006). Tracking an unknown time-varying number of speakers using TDOA measurements: a random finite set approach. *IEEE Transactions on Signal Processing*, 54(9):3291–3304.

Mahler, R. (2008). Random Set Theory for Multisource-Multitarget Information Fusion. In Liggins, M. E., Hall, D. L., and Llinas, J., editors, *Handbook of Multisensor Data Fusion*, pages 369–410. Taylor & Francis Group.

Mahler, R. (2013). "Statistics 102" for Multisource-Multitarget Detection and Tracking. *IEEE Journal of Selected Topics in Signal Processing*, 7(3):376–389.

Mahler, R. P. S. (2003). Multitarget Bayes filtering via first-order multitarget moments. *IEEE Transactions on Aerospace and Electronic Systems*, 39(4):1152–1178.

Marino, A., Antonelli, G., Aguiar, A. P., Pascoal, A., and Chiaverini, S. (2015). A Decentralized Strategy for Multirobot Sampling/Patrolling: Theory and Experiments. *IEEE Transactions on Control Systems Technology*, 23(1):313–322.

Matos, A. and Cruz, N. (2006). Simultaneous Acoustic Navigation of Multiple AUVs. In *7th IFAC Conference on Manoeuvring and Control of Marine Craft*, Lisbon.

Matos, A. and Cruz, N. (2009). MARES — Navigation, Control and On-board Software. In Inzartsev, A. V., editor, *Underwater Vehicles*, chapter 17, pages 315–326. InTech.

Matos, A., Cruz, N., Martins, A., and Pereira, F. L. (1999). Development and implementation of a low-cost LBL navigation system for an AUV. In *Proceedings of of the MTS/IEEE Oceans'99 Conference*, volume 2, pages 774–779 vol.2, Seattle, USA.

Mazurek, P. (2008). Deghosting Methods for Track-Before-Detect Multitarget Multisensor Algorithms. In Arreguin, J. M. R., editor, *Automation and Robotics*. I-Tech Education and Publishing.

Meduna, D. K. (2011). *Terrain Relative Navigation for Sensor-Limited Systems with Application to Underwater Vehicles*. Phd, Standord University.

Meduna, D. K., Rock, S. M., and McEwen, R. S. (2010). Closed-loop terrain relative navigation for AUVs with non-inertial grade navigation sensors. In *IEEE/OES Autonomous Underwater Vehicles*, pages 1–8. IEEE.

Meduna, D. K. D. K., Rock, S. M. S. M., and McEwen, R. S. (2008). Low-cost terrain relative navigation for long-range AUVs. In *Proceedings of of MTS/IEEE Oceans'08 Conference*, pages 1–7, Quebec City, Canada. IEE.

Melo, J. and Matos, A. (2008). Guidance and control of an ASV in AUV tracking operations. In *Proceedings of of the MTS/IEEE Oceans'08 Conference*, pages 1–7, Quebec City, Canada.

Morelande, M., Suvorova, S., Fletcher, F., Simakov, S., and Moran, B. (2015). Multi-target tracking for multistatic sonobuoy systems. In *18th International Conference on Information Fusion*, pages 327–332.

Morgado, M., Oliveira, P., Silvestre, C., and Vasconcelos, J. F. (2006). USBL/INS Tightly-Coupled Integration Technique for Underwater Vehicles. In *Proceedings of of 9th International Conference on Information Fusion*, pages 1–8.

Morice, C., Veres, S., and McPhail, S. (2009). Terrain referencing for autonomous navigation of underwater vehicles. In *Proceedings of of the MTS/IEEE Oceans'09 Conference*, pages 1–7, Bremen, Germany. IEEE.

Mu, H., Wu, M., Hu, X., and Ma, H. (2007). Geomagnetic Surface Navigation Using Adaptive EKF. In *2nd IEEE Conference on Industrial Electronics and Applications*, pages 2821–2825. IEEE.

Murangira, A., Musso, C., Dahia, K., and Allard, J. (2011). Robust regularized particle filter for terrain navigation. In *Proceedings of the 14th International Conference on Information Fusion*, number 4, pages 1–8.

Musso, C., Oudjane, N., and Gland, F. (2001). Improving Regularised Particle Filters. In Doucet, A., Freitas, N., and Gordon, N., editors, *Sequential Monte Carlo Methods in Practice*, Statistics for Engineering and Information Science, pages 247–271. Springer New York.

Napolitano, F., Cretollier, F., and Pelletier, H. (2005). GAPS, combined USBL + INS + GPS tracking system for fast deployable amp; high accuracy multiple target positioning. In *Proceedings of of the MTS/IEEE Oceans'05 Conference*, volume 2, pages 1415–1420 Vol. 2, Brest, France.

Nygren, I. (2005). *Terrain navigation for underwater vehicles*. PhD thesis, Royal Institute of Technology (KTH), School of Electrical Engineering.

Odell, D., Hertel, K., and Nielsen, C. (2002). New acoustic systems for AUV tracking, communications, and noise measurement at NSWCCD-ARD, Lake Pend Oreille, Idaho. In *Proceedings of of the MTS/IEEE Oceans'02 Conference*, volume 1, pages 266–271 vol.1, Biloxi, USA.

Paull, L., Huang, G., Seto, M., and Leonard, J. J. (2015). Communication-constrained multi-AUV cooperative SLAM. In *IEEE International Conference on Robotics and Automation*, pages 509–516.

Paull, L., Saeedi, S., Seto, M., and Li, H. (2014). AUV Navigation and Localization: A Review. *IEEE Journal of Oceanic Engineering*, 39(1):131–149.

Prins, R. and Kandemir, M. (2008). Time-constrained optimization of multi-AUV cooperative mine detection. In *Proceedings of of the MTS/IEEE Oceans'08 Conference*, pages 1–13, Quebec City, Canada.

Ren, Z., Chen, L., Zhang, H., and Wu, M. (2008). Research on geomagnetic-matching localization algorithm for unmanned underwater vehicles. In *International Conference on Information and Automation*, pages 1025–1029. IEEE.

Rice, H., Kelmenson, S., and Mendelsohn, L. (2004). Geophysical navigation technologies and applications. In *IEEE/ION Position Location and Navigation Symposium*, pages 618–624.

Salmond, D. and Gordon, N. (2005). An introduction to particle filters.

Schulz, B., Hobson, B., Kemp, M., Meyer, J., Moody, R., Pinnix, H., and St Clair, M. (2003). Field results of multi-UUV missions using ranger micro-UUVs. In *Proceedings of of the MTS/IEEE Oceans'03 Conference*, volume 2, pages 956–961 Vol.2, San Diego, USA.

Silvestre, C., Cunha, R., Paulino, N., and Pascoal, A. (2009). A Bottom-Following Preview Controller for Autonomous Underwater Vehicles. *IEEE Transactions on Control Systems Technology*, 17(2):257–266.

Stone, L. D., Streit, R. L., Corwin, T. L., and Bell, K. L. (2013). *Bayesian Multiple Target Trackin*. Radar/Remote Sensing. Artech House, second edi edition.

Stutters, L., Tiltman, C., and Brown, D. J. (2008). Navigation Technologies for Autonomous Underwater Vehicles. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 38(4):581–589.

Teck, T. Y. (2014). *Cooperative Algorithms for a team of Autonomous Underwater Vehicles*. PhD thesis, National University of Singapore.

Teixeira, F. C., Pascoal, A., and Maurya, P. (2012). A Novel Particle Filter Formulation with Application to Terrain-Aided Navigation. In *3rd IFAC Workshop on Navigation, Guidance and Control of Underwater Vehicles*.

Teixeira, F. C., Quintas, J., Maurya, P., and Pascoal, A. (2016). Robust particle filter formulations with application to terrain-aided navigation. *International Journal of Adaptive Control and Signal Processing*, pages n/a—-n/a.

Teixeira, F. J. C. M. (2007). *Terrain-Aided Navigation and Geophysical Navigation of Autonomous Underwater Vehicles*. PhD thesis, Universidade Técnica de Lisboa.

Thrun, S., Burgard, W., and Fox, D. (2005). *Probabilistic Robotics*. The MIT Press, Cambridge, MA, USA.

Tsiogkas, N., Papadimitriou, G., Saigol, Z., and Lane, D. (2014). Efficient multi-AUV cooperation using semantic knowledge representation for underwater archaeology missions. In *Proceedings of of the MTS/IEEE Oceans'14 Conference*, pages 1–6, St. John's, Canada.

Vahidi, A., Stefanopoulou, A., and Peng, H. (2005). Recursive least squares with forgetting for online estimation of vehicle mass and road grade: theory and experiments. *Vehicle System Dynamics*, 43(1):31–55.

Vaman, D. (2012). TRN history, trends and the unused potential. In *31st IEEE/AIAA Digital Avionics Systems Conference*, pages 1A3–1–1A3–16.

van der Merwe, R., Doucet, A., de Freitas, J. F. G., and Wan, E. (2000). The Unscented Particle Filter. Technical Report CUED/F-INFENG/TR 380, Cambridge University Engineering Department.

Vickery, K. (1998). Acoustic positioning systems. A practical overview of current systems. In *Proceedings of of the 1998 Workshop on Autonomous Underwater Vehicles, AUV'98.*, pages 5–17, Cambridge, USA. IEE.

Vo, B.-N., Singh, S., and Doucet, A. (2005). Sequential Monte Carlo methods for multitarget filtering with random finite sets. *IEEE Transactions on Aerospace and Electronic Systems*, 41(4):1224–1245.

Vo, B.-N. B.-T. and Vo, B.-N. B.-T. (2013). Labeled Random Finite Sets and Multi-Object Conjugate Priors. *IEEE Transactions on Signal Processing*, 61(13):3460–3475.

Wang, S., Zhang, H., Yang, K., and Tian, C. (2010). Study on the underwater geomagnetic navigation based on the integration of TERCOM and K-means clustering algorithm. In *Proceedings of of the MTS/IEEE Oceans'10 Conference*, pages 1–4, Sidney, Australia. IEEE.

Wang, X., Xu, M., Wang, H., Wu, Y., and Shi, H. (2012). Combination of Interacting Multiple Models with the Particle Filter for Three-Dimensional Target Tracking in Underwater Wireless Sensor Networks. *Mathematical Problems in Engineering*, 2012(829451):1–16.

Wang, Y. and Chaib-draa, B. (2012). An adaptive nonparametric particle filter for state estimation. In *IEEE International Conference on Robotics and Automation*, pages 4355–4360.

Watanabe, Y., Ochi, H., Shimura, T., and Hattori, T. (2009). A tracking of AUV with integration of SSBL acoustic positioning and transmitted INS data. In *Proceedings of of the MTS/IEEE Oceans'09 Conference*, pages 1–6, Bremen, Germany.

Webster, S. E., Eustice, R. M., Singh, H., and Whitcomb, L. L. (2012). Advances in single-beacon one-way-travel-time acoustic navigation for underwater vehicles. *The International Journal of Robotics Research*, 31(8):935–950.

Webster, S. E., Walls, J. M., Whitcomb, L. L., and Eustice, R. M. (2013). Decentralized Extended Information Filter for Single-Beacon Cooperative Acoustic Navigation: Theory and Experiments. *IEEE Transactions on Robotics*, 29(4):957–974.

White, B. A. (1998). Robust control of an unmanned underwater vehicle. In *Proceedings of the 37th IEEE Conference on Decision and Control*, volume 3, pages 2533–2534 vol.3.

Yang, R., Ng, G. W., and Bar-Shalom, Y. (2013). Tracking/fusion and deghosting with Doppler frequency from two passive acoustic sensors. In *16th International Conference on Information Fusion*, pages 1784–1790.

Yoerger, D. R., Bradley, A. M., Walden, B. B., Cormier, M.-H., and Ryan, W. B. F. (2000). Fine-scale seafloor survey in rugged deep-ocean terrain with an autonomous robot. In *IEEE International Conference on Robotics and Automation*, volume 2, pages 1787 –1792 vol.2.

Zhang, S., Chen, H., and Liu, M. (2014). Adaptive sensor scheduling for target tracking in underwater wireless sensor networks. In *International Conference on Mechatronics and Control*, pages 55–60.

Zhao, J., Wang, S., and Wang, A. (2009). Study on underwater navigation system based on geomagnetic match technique. In *9th International Conference on Electronic Measurement & Instruments*, pages 3–255–3–259. IEEE.