Thèse n° 7771

Cooperative Multi-Robot Systems for Aquatic Environmental Sensing

Présentée le 12 novembre 2021

Faculté de l’environnement naturel, architectural et construit
Laboratoire de systèmes et algorithmes intelligents distribués
Programme doctoral en robotique, contrôle et systèmes intelligents

pour l’obtention du grade de Docteur ès Sciences

par

Anwar Ahmad QURAISHI

Acceptée sur proposition du jury

Prof. A. Ijspeert, président du jury
Prof. A. Martinoli, directeur de thèse
Prof. A. Pascoal, rapporteur
Prof. G. Sukhatme, rapporteur
Prof. F. Mondada, rapporteur

École
polytechnique
fédérale
de Lausanne

2021

Acknowledgements

I have received a great deal of support from many people – family, friends, teachers and colleagues – that made this PhD journey possible. For that, I consider myself extremely fortunate.

First of all, I would like to express my thanks to my advisor, Prof. Alcherio Martinoli for his invaluable support and encouragement throughout this thesis. His constructive advice and feedback greatly influenced and shaped my research.

I would like to thank Emmanuel Droz, a dedicated and rigorous engineer, who helped design and fabricate various components of the robot and also set up a boat for facilitating our experiments. I learnt a lot by working alongside him. I would also like to thank Felix Schill and Alexander Bahr for their collaboration in the initial stages of this thesis.

I would like to express my gratitude to all my former and current lab colleagues, with whom I shared many memorable moments. I shall forever remember all the fun we had together. Thank you for making the lab a truly enjoyable workplace. I especially thank Corinne Farquharson, who not only made sure that the lab always functions smoothly, but also helped me (and all of us) with a variety of things, lab-related or otherwise.

I would like to acknowledge all the talented and hardworking students whose projects I had the opportunity to supervise. I appreciate their genuine enthusiasm and productive collaboration.

I am indebted to my closest and dearest friends, who live in Switzerland, India, and across the world, and yet always make themselves available whenever I need any kind of help, support or advice. I have been very lucky to have had them by my side.

My deepest and heartfelt gratitude goes to my family – my sister and my parents – for their unconditional love, boundless support, and constant inspiration. Thank you for always encouraging me to explore, experiment and learn. I am also deeply grateful to my loving grandparents, who have been extremely enthusiastic about my PhD from the start. I owe my family beyond what words can express. They deserve all the credit for making this PhD possible.

And finally, I am immensely thankful to my wife Aseena, a truly amazing partner and a friend, who has not only filled my life with love, but also supported me through the end of my PhD.

Lausanne, August 5, 2021 Anwar Quraishi

Abstract

The progress in the field of aquatic robotics generally lags behind that in ground and aerial robotics due to the severe challenges posed by the environment, especially when the operation needs to happen underwater. In particular, underwater robotics hardware has traditionally been expensive and large in size, preventing their proliferation beyond military and governmental applications. Nonetheless, recent technological achievements have led to the development of small-sized underwater robots and miniature acoustic systems for them. This has generated interest in studying the use of underwater robots for a variety of applications.

Yet, underwater robots still face a number of practical and technical challenges. Underwater environments are difficult to access. Radio communication and satellite-based positioning do not work below the surface. Turbidity in natural water bodies hampers visibility for camera-based localization. Acoustic communication, the usual mode of underwater communication, is low bandwidth, error prone and has high latency.

In this thesis, we have developed a collection of methods for facilitating cooperative, multi-robot operations with miniature underwater robots. The work done in this thesis lies at the intersection of aquatic and distributed robotics. We started with the integration of sensing and localization capabilities into a single underwater robot engaged in an environmental sensing mission. We then further developed localization, sensing and acoustic communication methods and tested them in outdoor, real-world environments using surface robots acting as surrogates for underwater robots. Finally, we moved on to multi-robot methods, developing cooperative localization and sensing algorithms. We tested our multi-robot methods in a simulated setting. The simulation was calibrated to reality using our experience with single-robot outdoor experiments.

We began with the demonstration of an autonomous underwater sampling mission. We showed that the robot could track and trace a region of interest, and gather high-quality spatial measurements from within it. Then, for individual robots, we developed an acoustic ranging and communication system to provide external position references from mobile acoustic beacons on the surface. We further presented a framework for optimization of parameters used for inertial localization, and showed improvement in localization accuracy.

Abstract

In a multi-robot system, we developed a method for cooperative localization using range-only relative measurements. We also presented an algorithm for optimal motion of mobile acoustic beacons on the surface. Both methods resulted in reduction of localization errors in individual robots. Finally, in simulated experiments, we compared the impact of the quality of localization and communication on a multi-robot adaptive environmental sampling method.

Keywords: Underwater robotics, distributed robotics, localization, acoustic ranging, acoustic communication, cooperative localization, adaptive environmental sampling.

Résumé

Les progrès dans le domaine de la robotique aquatique sont généralement à la traîne par rapport à la robotique terrestre et aérienne en raison des défis importants imposés par l'environnement, en particulier lorsque l'opération doit se dérouler sous l'eau. Le matériel de robotique sous-marine est traditionnellement cher et de grande taille, ce qui empêche sa prolifération au-delà des applications militaires et gouvernementales. Néanmoins, des récentes avancées technologiques ont conduit au développement de robots sous-marins de petite taille et de systèmes acoustiques miniatures pour ces derniers. Cela a suscité un intérêt pour l'étude de l'utilisation des robots sous-marins pour une variété d'applications.

Cependant, les robots sous-marins doivent encore relever un certain nombre de défis pratiques et techniques. Les environnements sous-marins sont difficiles d'accès. Les communications radio et le positionnement par satellite ne fonctionnent pas sous la surface. La turbidité des plans d'eau naturels nuit à la visibilité pour la localisation par caméra. La communication acoustique, le mode habituel de communication sous-marine, a une faible bande passante, est sujette à des erreurs et présente une latence élevée.

Dans cette thèse, nous avons développé un ensemble de méthodes pour faciliter les opérations coopératives avec des robots sous-marins miniatures. Le travail effectué dans cette thèse se situe à l'intersection de la robotique aquatique et de la robotique distribuée. Nous avons commencé par l'intégration de capacités de détection et de localisation pour un seul robot sous-marin, engagé dans une mission de détection environnementale. Nous avons ensuite développé des méthodes de localisation, de détection et de communication acoustique et les avons testées dans des environnements extérieurs réels à l'aide de robots de surface servant de substituts aux robots sous-marins. Enfin, nous sommes passés aux méthodes multi-robots, en développant des algorithmes de localisation et de détection coopératifs. Nous avons testé nos méthodes multi-robots dans un environnement simulé. La simulation a été calibrée pour être fidèle à la réalité au moyen des expériences extérieures précédentes utilisant un seul robot.

Nous avons commencé par la démonstration d'une mission autonome d'échantillonnage sous-marin. Nous avons montré que le robot pouvait suivre et tracer une région étudiée, et recueillir des mesures spatiales de haute qualité à l'intérieur de celle-ci. Ensuite, pour les robots individuels, nous avons développé un système de communication et de télémétrie acoustique pour fournir des références de position externes à partir de balises acoustiques mobiles disposées à la surface. Nous avons ensuite présenté une méthode pour l'optimisation des paramètres utilisés pour la localisation inertielle, et montré une amélioration de la précision de la localisation.

Dans un système multi-robots, nous avons développé une méthode de localisation coopérative utilisant des mesures relatives de portée uniquement. Nous avons également présenté un algorithme pour le mouvement optimal de balises acoustiques mobiles à la surface. Les deux méthodes ont permis de réduire les erreurs de localisation des robots. Enfin, dans des expériences simulées, nous avons comparé l'impact de la qualité de la localisation et de la communication sur une méthode d'échantillonnage environnemental adaptatif multi-robots.

Mots-clés : Robotique sous-marine, robotique distribuée, localisation, télémétrie acoustique, communication acoustique, localisation coopérative, échantillonnage environnemental adaptatif.

Contents

Acknowledgements **i**

Abstract (English/Français) **iii**

I Introduction and Background **1**

1 Introduction **3**
1.1 Underwater Robotics Applications 4
1.2 Underwater Environmental Sampling 5
1.3 Challenges 6

2 Aquatic Robotics Hardware **9**
2.1 Introduction to Aquatic Robots 9
2.2 Underwater Acoustic Transceivers 11
2.3 Navigation and Positioning Devices 14

3 Thesis Scope and Contributions **15**
3.1 Objectives and Outline 15
3.2 Scientific Contributions 16
3.3 Funding and Collaborator Contributions 17

II Platforms and Experimental Setup **19**

4 Introduction **21**
4.1 Overview 21
4.2 Experimental Setup 22

5 Robot Platforms and System Design **23**
5.1 Vertex Autonomous Underwater Vehicles 23
5.2 Autonomous Surface Vehicles 25
5.3 Hardware Architecture 26
5.4 Software Architecture 30
5.5 Contributions 31

6 Experimental Setup 33
6.1 Testing and Experimental Arenas 33
6.2 Base Station Setup 35
6.3 Simulation and Emulation 37

III Single-Robot Subsystems 39

7 Introduction 41
7.1 Outline 41
7.2 Related Work 43

8 State Estimation, Control and Navigation 47
8.1 State Estimation 47
8.2 Navigation 53
8.3 Control 54
8.4 Conclusion 56

9 Adaptive Sampling Missions 57
9.1 Problem Definition 57
9.2 Adaptive Behaviors 60
9.3 Results from Field Deployments 61
9.4 Conclusion 64

10 Acoustic Transceivers 65
10.1 Acoustic Transceiver Module 65
10.2 Range-Based Localization 67
10.3 Data-in-the-Loop Emulation 70
10.4 Field Experiments with ASVs 71
10.5 Conclusion 74

11 Acoustic Communication and Enhanced Localization 75
11.1 Underwater Acoustic Communication 75
11.2 Implementation and Error Correction 76
11.3 Field Experiments with ASVs 81
11.4 Commercial Hardware 83
11.5 Conclusion 84

12 Parameter Estimation for Inertial Localization 85
12.1 Introduction and Motivation 85
12.2 Preliminaries 87
12.3 Optimization Framework 88
12.4 Results from Field Experiments 90
12.5 Conclusion 96

IV Multi-Robot System **97**

13 Introduction **99**
13.1 Outline 99
13.2 Related Work 100

14 Multi-Robot Simulation **103**
14.1 Basic Elements 103
14.2 Calibration of Robot Odometry 104
14.3 Simulating Acoustic Ranging and Communication 107
14.4 Conclusion 108

15 Cooperative Localization **109**
15.1 Motivation and Background 110
15.2 Covariance Intersection for Cooperative Localization 110
15.3 Peer Selection 114
15.4 Simulation and Field Experiments 116
15.5 Conclusion 120

16 Coordinated Mobile Beacons **121**
16.1 Problem Definition 122
16.2 Methodology 122
16.3 Experiments in Simulation 125
16.4 Conclusion 130

17 Impact on Multi-Robot Adaptive Sampling **131**
17.1 Related Work 131
17.2 Methodology 132
17.3 Experimental Setup 134
17.4 Impact of Communication and Localization Errors 135
17.5 Conclusion 137

V Conclusion and Outlook **139**

18 Conclusion and Outlook **141**
18.1 Summary 142
18.2 Discussion and Future Work 144
18.3 Lessons Learnt 146

Bibliography **149**

Curriculum Vitae **159**

Introduction and Background Part I

1 Introduction

Oceans and inland water bodies cover about 71 % of the surface of our planet. They are rich ecosystems comprising of plant, animal, and microbial life, as well as minerals and materials. However, much of the volume below their surface is yet unexplored. These water bodies hold a wealth of scientific knowledge and information. They also support many lucrative activities, contributing significantly to the world economy. The World Economic Forum estimated that oceans (inland water bodies excluded) contributed US$ 2.5 trillion to the world economy in the year 2020, including marine fisheries, trade, and transport, employing tens of millions of people [1]. Inland water bodies contribute additionally with freshwater fisheries, irrigation, and drinking water. Therefore, the health of oceans, coasts, and freshwater ecosystems is crucial, as is their security. Excessive exploitation of ocean resources, pollution, oil spills, and climate change threaten to alter or even damage the aquatic ecosystems and living organisms within them.

Figure 1.1: A toxic algal bloom induced by phosphorous pollution. Algal blooms kill aquatic life, degrade drinking water quality, and pose other dangers. Image source: PBS.

Monitoring and exploring water bodies is of interest for two broad reasons. Firstly, by measuring various phenomena in water bodies, biologists, and limnologists hope to understand the physical, chemical, and biological processes taking place in aquatic ecosystems. Secondly, by studying changes in various parameters over time, they hope to understand the effect of external causes such as pollution and global warming on aquatic life. This is an important step towards preserving aquatic ecosystems.

Performing large-scale, continuous measurements in water bodies is a complex task, however. The primary reasons behind that are the difficulty of reaching underwater spaces, lack of radio communication, and labor-intensive effort for operating underwater equipment. Remote sensing methods are limited to the surface layers and do not provide information from deep below. We argue that teams of robots working together are perfect candidates for the task of exploring and monitoring water bodies. To begin with, robots can cover large volumes of water autonomously and reduce manual effort. Additionally, multi-robot systems bring several advantages, including accelerated operation, robustness, and fault tolerance.

While underwater robots have existed for a long time, they have typically been expensive and large in size and weight, hindering their wide adoption. It also made them difficult to deploy in shallow, coastal areas and inland lakes, where ships with cranes are generally not available. However, the continued miniaturization of computation and sensing hardware in recent years has enabled the development of several low-cost, lightweight underwater robot platforms. While the small sizes prevent the use of traditional peripherals for localization and communication, they do offer an opportunity for the development of new methods and hardware for effective real-world applications.

In this thesis, we attempted to push the state of the art in the field of underwater robotics and develop algorithms that would facilitate cooperative, multi-robot environmental sampling with miniature underwater robots. The work done in this thesis lies at the intersection of underwater and distributed robotics.

1.1 Underwater Robotics Applications

Underwater robots have a wide variety of applications in addition to environmental monitoring. This section briefly introduces some of the applications. A more comprehensive list of applications of marine robots is presented in [2].

Surveying and Mapping

Underwater robots have long been used for surveying and mapping the sea floor. The robots are typically equipped with side-scan or other kinds of sonars, which allow them to generate images of large areas of the sea bottom. More recently, cameras have also been used to capture such images [3]. These images are then stitched together to generate detailed height-maps of the seabed. Such maps have several uses, which include studying marine life, hydrodynamic modeling for calculating currents, tides and predicting floods [4]. Underwater robots have also been used for mapping under-ice surfaces [5], and exploring coral reefs in order to assess their health [6]. They have further been used for marine archeology, discovering and mapping old shipwrecks [7], and even recovering artifacts [8].

Search and Rescue

Underwater robots have been used in supporting marine search and rescue operations. A notable instance is the use of a swarm of underwater robots in an attempt to locate the wreckage of the Malaysian Airlines Boeing 777 aircraft in the Indian Ocean [9]. In many emergency situations, however, expert human divers are involved in risky operations. With new advancements, underwater robots have the potential to complement or even replace human divers.

Inspection

Robots have been employed for inspection of underwater objects and infrastructure for a long time. While remotely operated underwater robots were initially more common, autonomous robots are increasingly being used. For example, robots are used for inspection of installations in the oil and gas industry [10], in fish farms [11], and for inspecting ship hulls [12].

1.2 Underwater Environmental Sampling

For biologists and limnologists, measuring various physical and chemical quantities in water bodies is crucial to understanding the aquatic ecosystems and the phenomena taking place in them. For instance, extensive measurements in Lake Cadagno, Switzerland helped understand the activity of the sulfur-reducing bacteria in a compact layer within the lake [13]. However, gathering high-quality data at high spatial and temporal resolutions can be difficult.

Traditional sampling methods can be broadly classified into two types: in-situ, where measurements are performed and recorded in place, and in-vitro, where samples are collected for further processing in a laboratory. We consider only in-situ measurements, which can be performed by fixed (anchored or moored) static sensors, providing a temporal series of measurements, or by mobile probes that are deployed at and retrieved from various locations repeatedly for spatial coverage. A picture of both kinds of sensors is shown in Fig. 1.2. Both approaches require significant effort and provide data at limited spatio-temporal resolutions.

Robots, on the other hand, can transport sensing probes to various locations and gather data at unprecedented resolutions. This will allow capturing and studying various phenomena that take place in lakes at smaller time and length scales. The algorithms and methods in this thesis were developed with underwater sampling applications in mind.

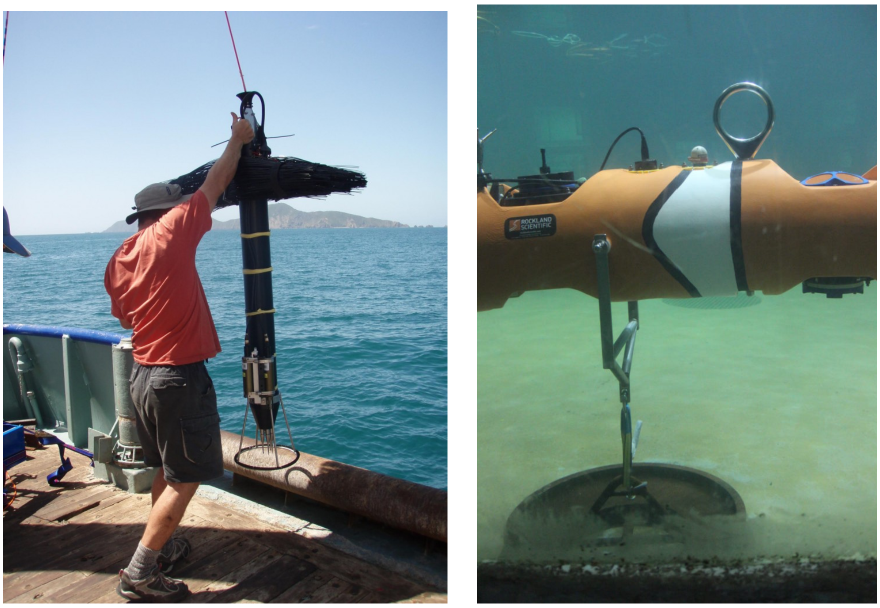


Figure 1.2: Left: VMP-500 profiler equipped with a host of sensing probes. The device is manually deployed at various locations in a lake for gathering vertical profiles. Right: Nemo mooring, also equipped with a host of sensors. This moored device collects measurements at a static location over time. Image source: Rockland Scientific.

1.3 Challenges

Progress in the field of underwater robotics generally lags behind that in surface or aerial robotics for several reasons. Water as a medium poses several technical and practical challenges, which increase the cost and complexity of designing and operating underwater robots. Reducing price and miniaturization of sensing and computing hardware has generally benefited the advancement of robotics. However, some of these developments are not directly applicable to underwater robotics. As a result, robot platforms and peripherals for underwater robotics tend to be more expensive or not as easily available as compared to their aerial and ground equivalents.

Technical challenges primarily stem from the physical property of water that electromagnetic waves with higher frequencies are severely attenuated in the medium. As a result, radio communication modalities commonly used by ground and aerial robots do not work in water. Satellite-based navigation systems also do not work below the water surface. Robots have to rely on low frequency radio or acoustic signal based communication, which have their own range and bandwidth limitations. Further, visually distinct features are sparse away from the lake-bed, and turbidity or sediments hinder visibility for camera-based localization.

Practical challenges are related to the general physical accessibility of underwater environments and the need for impermeable enclosures. All robotic equipment needs to be properly water- and pressure-proofed before being deployed. Once the robot is deployed, it is out of sight, and there is no visual feedback of robot actions. Any robot failure can be expensive, because it can be difficult and cumbersome to retrieve a malfunctioning robot. This necessitates the development of a variety of fail-safe systems followed by thorough testing, which adds cost and complexity to the development cycle.

Further, traditional peripherals for underwater vehicles, such as acoustic transceivers and sonars, are too large (and expensive), and incompatible with small-sized underwater robots. As a result, research teams are often compelled to develop some of these components in-house – as we did with acoustic transceivers. A number of new sensors and peripheral devices for underwater robots have recently become available commercially, during the course of this project, showing that there has been recent progress in the field.

Underwater communication and sensing modalities have unique characteristics. Their application to robotics requires the development of new methods and algorithms. For example, when underwater computer vision is used, it requires special lenses or software to compensate for the refractive properties of the air-water interface. Acoustic signals have a high latency due to the low speed of sound in comparison to that of electromagnetic waves. By the time an acoustic signal reaches its target, it is already several milliseconds old. All underwater communication, low frequency radio or acoustic, is low bandwidth, which makes underwater communication a scarce and expensive resource. Coordination algorithms need to take this into account and minimize the exchange of information, yet achieve the required coordination.

The work done in this thesis primarily addresses communication and localization in underwater multi-robot systems. Given the aforementioned challenges, external information available to individual robots for either coordination or localization is sparse. We attempt to exploit this available information in an efficient manner. To overcome the challenge of limited external information, we also attempt to use computation, which has become inexpensive tanks to recent advancements.

The following two chapters in this part of the manuscript introduce the hardware used in aquatic robots and outline the contributions made in this thesis.

2 Aquatic Robotics Hardware

In this chapter, we will introduce aquatic robot platforms as well as related sensing and communication peripherals. We will also introduce the hardware developed and used in this thesis. We used the broad term ‘aquatic robots’ in order to encompass both, surface and underwater robots.

2.1 Introduction to Aquatic Robots

We classify aquatic robots into two types, Autonomous Underwater Vehicles (AUVs) and Autonomous Surface Vehicles (ASVs). In this thesis, both kinds of robots have been used for different purposes. We have used the AUVs for testing various subsystems under development as well as for actual environmental sampling missions. We have used ASVs as surrogates for AUVs to rapidly test specific subsystems under development, and also to provide navigation support to AUVs by acting as intelligent, moving acoustic beacons.

2.1.1 Autonomous Underwater Vehicles

A number of examples of commercially available, actively propelled AUVs of various sizes are shown in Fig. 2.1. Note that this list is not exhaustive. They are 1-5 m in length and weigh between 10-700 kg. Some of them have a modular design, where additional equipment can be attached, and thus increasing their size. The physical characteristics of these AUVs are compiled in Table 2.1.

For AUVs to be easily deployable for sensing applications in lakes and coastal areas, an important requirement for us was that they should be transportable by 1-2 persons. There are few commercially available AUVs that meet this requirement. Notable ones are the Riptide family of robots, called Unmanned Undersea Vehicles (UUV) [18], [19]. They are around 1 m long and weigh around 10 kg (depending on configuration), are capable of operating at a maximum depth of 300 m, and are customizable for different applications.

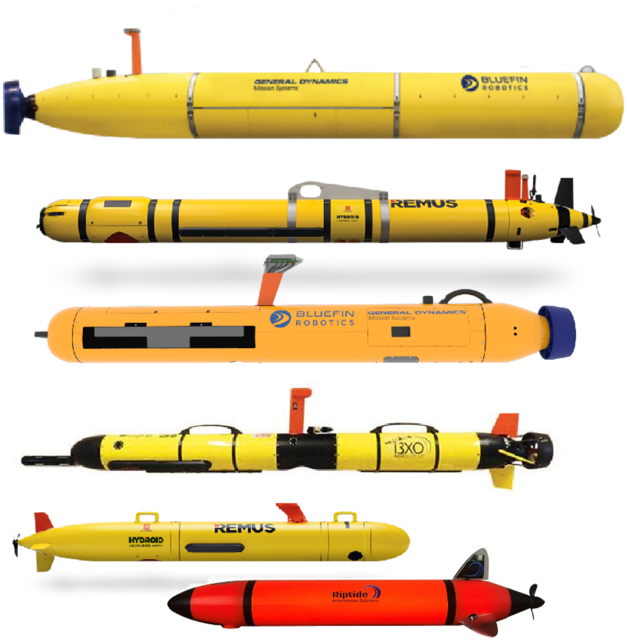


Figure 2.1: Examples of commercially available AUVs. From top: Bluefin-21 [14], REMUS 600 [15], Bluefin-9 [16], YSI EcoMapper [17], REMUS 100 [15] and Riptide UUV [18], [19].

Research prototypes also exist, some of which have not been used in real-world applications. Maehle et al. presented two micro AUV platforms, MONSUN II [20] and SEMBIO [21] (neither are commercially available). While MONSUN and its newer version, MONSUN II are designed for swarm-based environmental monitoring, SEMBIO is designed to be energy efficient and to aid swarm robotics research. They are both around 0.5 m long and are rated to a depth of 10 m, which for many applications might be limiting. HippoCampus is another AUV platform, developed by Solowjow et al. [22] also specifically for swarm robotics research. Their initial application environments are restricted to laboratory tanks with depths of up to 10 m. The MEDUSA AUVs presented in [23] have been used in several experiments at sea. They are about 1 m long and have a depth rating of 50 m.

In this thesis, we used the Vertex AUV [24], shown in Fig. 2.2. This small and lightweight AUV platform was developed in our laboratory and is currently manufactured and commercialized by Hydromea[1]. They are introduced in detail in Chapter 5. At the time of starting this work, the Vertex AUVs were in primitive stages of development. As the work of this thesis progressed, we performed several hardware and software changes and developments in response to practical needs and challenges. A significant effort in this work went towards developing simulation and logistical tools for gradually moving from the lab to harsh outdoor environments.

Name	Dimensions (dia × length)	Weight	Depth rating
Bluefin 21	53 cm × 5 m	750 kg	4500 m
Remus 600	32 cm × 2.7 m+	220 kg+	600 m
Bluefin 9	23 cm × 2.5 m	70 kg	200 m
YSI EcoMapper	15 cm × 2 m	30 kg	100 m
Remus 100	19 cm × 1.7 m	36 kg	100 m
Riptide UUV	12 cm × 1.0 m	10 kg	300 m
Hydromea Vertex	14 cm × 0.7 m	7 kg	300 m

Table 2.1: Physical characteristics of some commercially available AUVs as well as that of Vertex AUV.

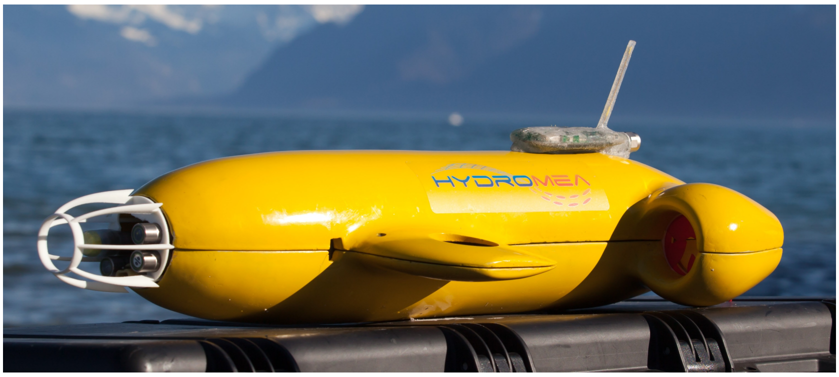


Figure 2.2: Hydromea Vertex AUV. It is 70 cm long and weighs 7 kg.

2.1.2 Autonomous Surface Vehicles

ASVs are much less complex compared to their underwater counterparts. Being above the surface, they can leverage GNSS reception and radio communication. Some commercially available ASVs are shown in Fig. 2.3.

During this thesis, we built a fleet of lightweight ASVs by outfitting small, remote-controlled boats with autopilot and computation hardware (see Fig. 2.4). We used electronics hardware that was similar to that on the AUV, which also allowed us to use the same software. This made it possible to easily substitute the AUVs with ASVs for quick experiments since the ASVs are easier to deploy. The ASVs are introduced in detail in Chapter 5.

2.2 Underwater Acoustic Transceivers

Given the high attenuation of electromagnetic waves, acoustic waves are often used as signal carriers in the underwater medium. Acoustic transceivers are used for both communication as well as localization. For communication, acoustic signals act as the carrier of information.

---

[1] Hydromea S.A., Lausanne, Switzerland. Web: www.hydromea.com

Figure 2.3: Examples of commercially available ASVs. Left: Clearpath Heron [25], Right: C-Worker 4 [26].

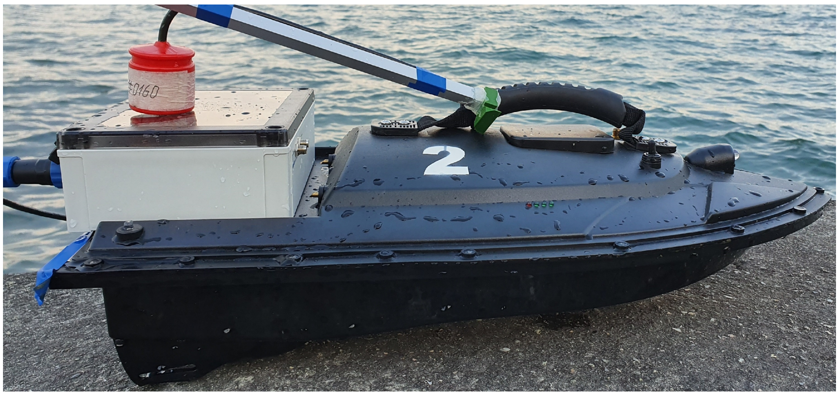


Figure 2.4: ASVs developed and used in this thesis. They weigh around 2.5 kg and are 50 cm long.

Note that compared to radio communication, acoustic communication has low throughput (of the order of few bytes per second) and high latency (due to low propagation speed). For localization, these transceivers either measure time-of-flight or difference in time of arrival of a signal. This is then converted to a geometric quantity, such as distance or direction of arrival, and used to calculate position.

Table 2.2 and Fig. 2.5 show examples of commercially available modems. Almost all of the available modems during the start of this thesis were too large for the Vertex AUV. Compact

Name	Throughput	Range	Dimensions (dia × length)	Weight (dry / wet) kg
Evologics HS	62 kbit/s	300 m	63 × 310 mm	1.2 / 0.3 kg
Teledyne ATM-925	15 kbit/s	4000 m	89 × 850 mm	10.8 / 3.6 kg
Sonardyne Modem 6 Sub-mini	9 kbit/s	1000 m	97 × 420 mm	5.5 / 3.2 kg
Waterlinked M64	64 bit/s	200 m	30 × 112 mm	0.12 kg (dry)
DiveNET Microlink	78 bit/s	1000 m	40 × 45 mm	0.16 kg (dry)

Table 2.2: Examples of commercially available acoustic modems and their characteristics.

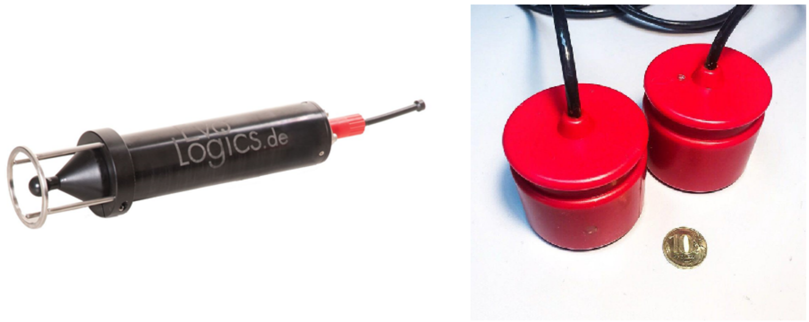


Figure 2.5: Examples of commercially available underwater acoustic modems. Left: Evologics HS, Right: DiveNET Microlink.

Figure 2.6: Examples of commercially available underwater positioning systems. They contain an array of receivers that can measure both distance to and direction of a transmitter. Left to right: Waterlinked Underwater GPS, DiveNET Commander positioning system, Advanced Navigation Subsonus.

sized modems have only recently become available. In addition to developing our own acoustic modems, we integrated the DiveNET Microlink modem [27] into our AUVs and ASVs in the later part of the thesis.

Many of the modems listed above also perform measurement of range or even direction of arrival, which can be used for localization. However, there are some products that are dedicated to localization. They have a multi-channel array of hydrophones for receiving an incoming signal and they measure the time of arrival as well as difference in time of arrival at each hydrophone. Such a receiver array can be installed at a base station to track AUVs, or installed on the AUVs so that they can localize themselves with respect to a fixed base station. Some acoustic localization products are shown in Fig. 2.6.

A number of research prototypes of acoustic transceivers for communication and localization exist [28]–[31]. They have been used by research groups for their own experimental activities. They are not available commercially.

2.2.1 Other Communication Modalities

There are other underwater communication modalities that are used less often in practical robotics applications. They include Very Low Frequency (VLF) radio and optical commu-

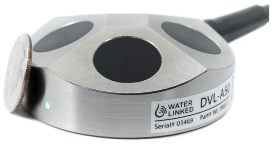


Figure 2.7: DVL 150 Doppler Velocity Logger from Waterlinked. It measures 66 × 25 mm, and weighs 250 g in air.

nication. Attenuation of electromagnetic waves in water increases with carrier frequency, therefore, VLF radio has lower attenuation than standard radio communication channels. However, the size of the antenna required for a VLF radio is a concern for miniature AUVs. Optical communication can provide high throughput but its range is limited to a few meters, especially when the water is not clear. Line of sight and alignment requirements make it cumbersome to use.

2.3 Navigation and Positioning Devices

Other than acoustic transceivers themselves, other sensors that can aid navigation include Doppler Velocity Logger (DVL), side-scan sonar, and pressure sensors. The pressure sensor is the simplest of all, it provides an accurate vertical position based on water pressure measurements. Typical pressure sensors are also compact enough to be easily integrated into miniature AUVs and small submerged devices.

A DVL works by transmitting acoustic beams in different directions towards the seafloor and measuring the frequency shift in the reflected echo. Relative motion between a robot and the seafloor causes the frequency shift in the reflected echo due to the Doppler effect. A number of DVL products are commercially available, but all of them are too large or heavy for miniature AUVs. The only product with a size compatible with the Vertex AUV is the DVL A50. It was launched recently, in the year 2020, and is shown in Fig. 2.7. A common operational requirement of all DVLs is sufficient proximity to the sea floor.

Side-scan sonars use acoustic beams and create an image of the seafloor based on the reflected beams. They have been used to perform localization using an approach similar to Simultaneous Localization and Mapping (SLAM). However, no compact sonar products exist that are compatible with miniature-sized AUVs.

3 Thesis Scope and Contributions

Underwater multi-robot systems are the primary focus of this thesis. The work done in this thesis was inspired and shaped by a real-world need to leverage robots to gather high-quality environmental measurements for scientific purposes. Such an application requires the development of methods to address important problems in underwater robotics. This chapter highlights the objectives of the work done in this thesis and the contributions made to the existing state-of-the-art in underwater robotics. It outlines the structure of this thesis and identifies our specific contributions and related publications.

3.1 Objectives and Outline

The research done in this thesis can be broadly divided into two main and sequential efforts: (i) developing single-robot hardware and software subsystems for underwater navigation and communication, and (ii) developing methods that achieve multi-robot coordination and cooperation under the constraints of underwater environments. Our objectives were to

- Push the boundaries in the state of the art in distributed underwater robotics,
- Make it easy to deploy robots without the need for existing infrastructure, for underwater operations in general and limnological sampling in particular, and
- Integrate information from various sources over long time horizons, given external information and sensing in the underwater domain is scarce.

This thesis is divided into five parts:

Part I: Introduction and Background This part introduces the general landscape of underwater robotic systems, and their application to underwater environmental sampling. It further highlights the main challenges of the domain and contributions made during this thesis.

Part II: Platforms and Experimental Setup In the second part, we present the robotic platforms used in this project and the experimental arenas used for evaluating our methods. We also present software tools developed for facilitating outdoor robot deployments.

Part III: Single-Robot Subsystems This part primarily focuses on subsystems and methods developed for individual robots, in particular for localization and communication. Results from experiments in outdoor environments are presented.

Part IV: Multi-Robot System In the fourth part, we describe the methods developed for cooperation in multi-robot systems. Since experiments are largely performed in simulation, the simulation setup is also introduced in detail.

Part V: Conclusion and Outlook In the final part, we summarize the research conducted and lessons learnt during this thesis. Leveraging the experience gathered over the course, we provide an outlook for possible future research.

3.2 Scientific Contributions

The work done in this thesis resulted in several scientific contributions and related publications. The work is primarily concerned with improving localization and communication in an underwater robotic system, given the constraints and challenges of the underwater environment.

Part III: Single-Robot Subsystems The first set of contributions pertain to the work done in this part, on single-robot systems. The following publication presents results from an outdoor adaptive environmental sampling mission, one of several performed during this thesis. The robot was programmed to track and trace a region of interest, with motors turned off to avoid disturbance to sensitive temperature measurements.

1. A. Quraishi, A. Bahr, F. Schill, and A. Martinoli, “Autonomous Feature Tracing and Adaptive Sampling in Real-World Underwater Environments”, in *IEEE International Conference on Robotics and Automation*, 2018, pp. 5699–5704.

The following publications presented our work on developing acoustic transceivers for communication and range measurements. They were eventually used for localization.

2. A. Quraishi, A. Bahr, F. Schill, and A. Martinoli, “Easily Deployable Underwater Acoustic Navigation System for Multi-Vehicle Environmental Sampling Applications”, in *IEEE International Conference on Robotics and Automation*, 2019, pp. 3464–3470.
3. A. Quraishi, A. Bahr, F. Schill, and A. Martinoli, “A Flexible Navigation Support System for a Team of Underwater Robots”, in *IEEE International Symposium on Multi-Robot and Multi-Agent Systems (MRS)*, 2019, pp. 70–75.

We subsequently worked on using an optimization framework to estimate the kinematic and dynamics parameters of robots. The parameters are used for inertial localization. The work resulted in the following publication:

4. A. Quraishi and A. Martinoli, "Online Kinematic and Dynamic Parameter Estimation for Autonomous Surface and Underwater Vehicles", in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2021. To appear.

Part IV: Multi-Robot System The work in this part pertains to multi-robot systems and cooperation. We developed a method for cooperative localization using pair-wise range-only relative measurements. We further derived a heuristic for peer-selection for performing range measurements. Our work is presented in the following publication:

5. A. Quraishi and A. Martinoli, "Distributed Cooperative Localization with Efficient Pairwise Range Measurements", in *International Symposium on Distributed Autonomous Robotic Systems, 2021*, Springer Proceedings in Advanced Robotics, 2021. To appear.

Then, we worked on path-planning for actively controlled surface acoustic beacons for supporting localization for a group of underwater robots. This work resulted in the following publication:

6. A. Quraishi and A. Martinoli, "Coordinated Path Planning for Surface Acoustic Beacons for Supporting Underwater Localization", in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2021. To appear.

3.3 Funding and Collaborator Contributions

A significant amount of the work of this thesis was carried out within the framework of an interdisciplinary project entitled 'A Flexible Underwater Distributed Robotic System for High-Resolution Sensing of Aquatic Ecosystems'. The project was sponsored by the Sinergia programme of the Swiss National Science Foundation under grant CRSII2_160726/1. The project was a collaboration between three research groups:

- Distributed Intelligent Systems and Algorithms Laboratory at EPFL
- Physics of Aquatic Systems Laboratory (APHYS) at EPFL
- Microbial Ecology group, Department F.-A. Forel for Environmental and Aquatic Sciences at University of Geneva

Correspondingly, this thesis was partially funded under the same grant.

Our former colleagues Dr. Alexander Bahr and Dr. Felix Schill were responsible for the design of the Vertex AUV at our laboratory before the start of this thesis. They were also responsible for the integration of the YSI EXO2 sensing system with the Vertex AUV, and the development of a fast temperature sensor for certain sampling missions. They later founded the company Hydromea for commercializing the AUV. We would like to thank both of them for their help and support with further developments, modifications, and operation of the AUVs, both during their time at our lab and as a part of their new company. We delineate the work done at our lab by them, at Hydromea, and as a part of this thesis in Chapter 5.5.

We would also like to thank our lab engineer Emmanuel Droz for helping us with logistics, engineering, and setting up of a boat as a launch platform for our experiments. The platform is introduced in Chapter 6.

Finally, we would like to thank our colleagues from APHYS and the Microbial Ecology group, in collaboration with whom we performed several underwater sampling missions in Lake Cadagno, Lake Zurich and Lake Greifen. In particular, the missions presented in Chapter 9 were carried out in collaboration with Oscar Sepúlveda of APHYS, headed by Prof. Alfred Johny Wüest.

Platforms and Experimental Setup Part II

4 Introduction

In line with the main objectives, a significant portion of the experiments during this thesis were performed in outdoor, real-world environments, primarily using two different robot platforms – an AUV and an ASV. Both platforms were developed in our lab. Given the complexity in operating robots outdoors, a range of additional experimental and simulation setups were used to test and improve various subsystems. To facilitate systematic and safe outdoor deployments, software tools for managing the operation of the robots were developed. The subsequent chapters in this part introduce the robot platforms, tools, and experimental arenas used during this thesis.

4.1 Overview

We used the Vertex AUV for performing underwater environmental sampling experiments. In addition, we used a fleet of ASVs for two broad purposes. They were used as surrogates for AUVs for experimental testing and evaluation of various methods in real outdoor environments. They were also used as a support system to aid underwater localization. Sections 5.1 and 5.2 in Chapter 5 describe both robot platforms in detail.

Both platforms are composed of several individual software and hardware modules. These modules were designed to be self-contained subsystems compatible with either robot platform (to the extent possible). This allowed us to first test the modules on the ASV (serving as surrogates for AUV), since they are easier to deploy. The ASVs are also easier to retrieve in case of a catastrophic failure caused by a module under test. Sections 5.3 and 5.4 in Chapter 5 provide an insight into the modular architecture of the robots. The specific contributions of this thesis to the robot hardware and software are delineated in Section 5.5.

The small size, while facilitating easier deployment, precludes integration of traditionally available underwater localization and communication peripherals. To overcome this limitation, we attempted to develop our own acoustic transceivers, as well as integrate new commercial hardware that became available during the course of this thesis.

4.2 Experimental Setup

During the initial part of the thesis, single-AUV and ASV subsystems were developed and their successful operation was demonstrated during several outdoor experimental deployments. Outdoor environments are dynamic and are prone to external disturbances, making it difficult to perform repeated experiments under similar conditions. As a result, they take significant time and effort. Therefore, a simulation system is a crucial tool that helps prototype and evaluate new algorithms and methods faster. In the later part, production and delivery issues with additional Vertex AUVs and damages during outdoor experiments forced a shift towards simulation for multi-AUV experiments.

Real Experiments

Experiments and tests were performed in several different arenas, namely a tank or a swimming pool, near a lake shore (launched from the shore), and in the lake (launched from a boat). A tank or a pool offered a structured, controlled environment that was easy to access but was not realistic. On the other hand, the lake offered a realistic environment but posed logistical and practical complexity. In addition, experiments were also hindered by bad weather. The lake shore offered a compromise between the two but was still subject to weather conditions. The experimental arenas are introduced in detail in Chapter 6.

Simulations

A multi-tiered simulation setup, combining various degrees of real and simulated conditions, is briefly introduced in Section 6.3 of Chapter 6. Details about the different simulation setups are presented in relevant chapters. The multi-robot simulation setup is presented later in Chapter 14.

Workflow

A failure during outdoor experiments can be potentially expensive. Failure in ASVs would cause them to be stranded on the surface away from the launch point. AUV failures can be more catastrophic. While they are tuned to be slightly buoyant, they are still submerged while on the surface and hard to locate visually. However, leakage problems or software malfunctions can cause them to travel downwards or sink, making them harder to retrieve. Therefore, before outdoor deployment, any significant addition or modification to a particular module is subject to thorough testing using a combination of hardware-in-the-loop and near-shore tests.

5 Robot Platforms and System Design

Robots of two different types were used in this research – Vertex AUVs and ASVs. The AUVs were meant to be used for regular underwater operations. However, underwater robot operations are complex and need external support (e.g., for localization). To that end, the ASVs served dual purposes. In several experiments, they were used as surrogates for AUVs for quick experiments. ASVs were also used to provide localization support for underwater operations.

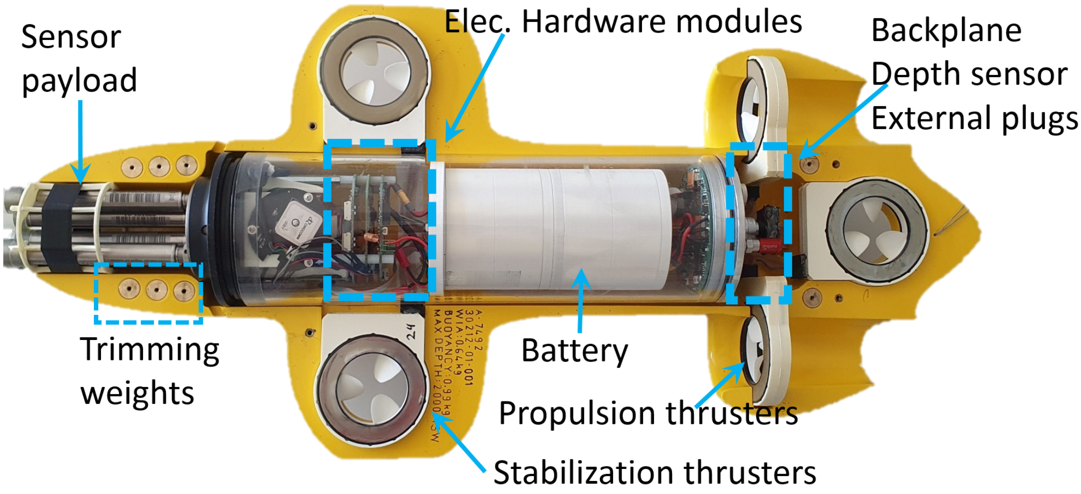
The AUVs and ASVs used similar electronic hardware components. This allowed us to use the same software for both robot platforms and helped keep the operational aspect of this project simpler. While much of the hardware and a portion of the software was acquired along with the Vertex AUVs, a significant amount of hardware and software development was done as a part of this thesis. This chapter introduces the mechanical and electrical hardware, and the software framework of the robots.

5.1 Vertex Autonomous Underwater Vehicles

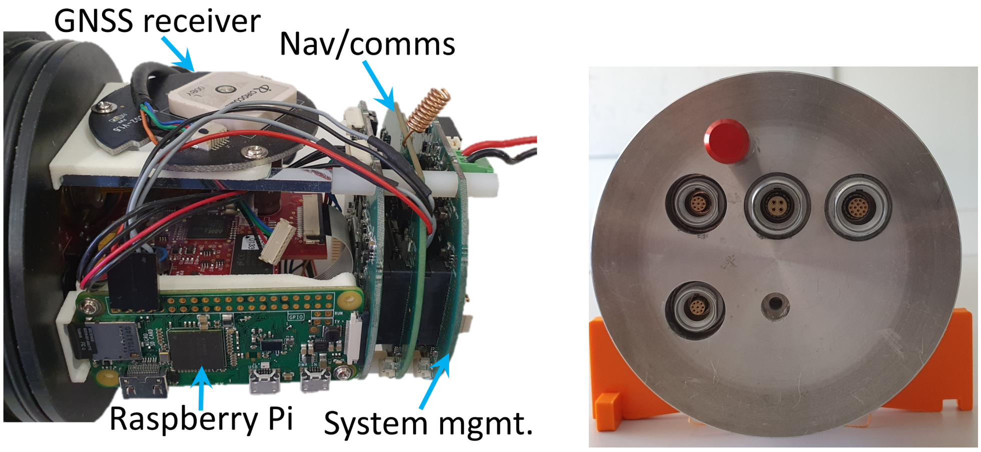
The Vertex AUVs were designed (before the start of this thesis) specifically as a platform for distributed environmental sensing in lakes and coastal areas. They are currently being commercialized by Hydromea S.A. The AUV weighs 7 kg and is about 70 cm in length, which makes it easy for one person to deploy and retrieve them. Despite the small size, the AUV has an autonomy of 6-8 hours under normal operation.

5.1.1 Mechanical Hardware

A picture of the open AUV hull is shown in Fig. 5.1a. The electronics and the battery are enclosed in a water- and pressure-proof interior hull. Connection to external peripherals such as motors and acoustic modems is provided through waterproof plugs (see Fig. 5.1c). This internal hull is further enclosed in a free-flooding, buoyant exterior hull.



(a) Inside AUV hull.



(b) Assembled hardware stack.

(c) Backplane.

Figure 5.1: (a) An open hull showing various components of the Vertex AUV. (b) Zoomed-in view of the electrical hardware stack that is situated in the front of the AUV. (c) Backplane showing the depth sensor (with a red cap) and sockets for plugging in external modules such as acoustic transceiver and thrusters.

Actuation is provided by five thrusters, of which two are forward facing for propulsion and three are vertically oriented for stabilization. The AUV weight is tuned to be slightly positively buoyant as a safety mechanism. It is trimmed to be naturally horizontal about the pitch axis. Slots for adding weights (shown in Fig. 5.1a) provide a way to tune the buoyancy and trim the weight distribution. Attitude about the roll axis is maintained via active control using the two front thrusters. The maximum cruise speed is about 2 m/s.

A suite of sensors for measuring physical and chemical parameters is attached in the front. This placement of the sensors allows measurement with minimal interference from the thrusters behind when the AUV is moving.

5.1.2 Electrical Hardware

The AUV is composed of several modules, each for a different task such as navigation, system management, sensor payload, and computation. Each of these modules is a separate Printed Circuit Board (PCB), with its own processor for computation. A picture of three modules on the assembled electronic stack is shown in Fig. 5.1b. A detailed description of the various modules and the overall hardware architecture is provided in Section 5.3.

5.1.3 Environmental Sensing System

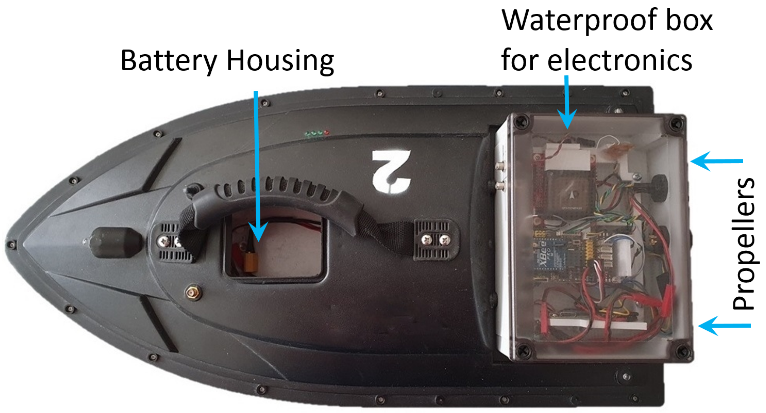
The AUV is equipped with a YSI EXO2 sensing system [38], which is mounted in the front, as shown in Fig. 5.1a. This sensing system is modular itself, and can be configured with up to seven different physical and chemical sensors for in-situ measurements. Example sensor modules include CTD (Conductivity-Temperature-Depth), dissolved oxygen, blue-green algae, chlorophyll concentration, turbidity, and a fast (high sampling rate) temperature sensor. Other modules on the AUV have online access to real-time sensor data, which can be used for decision making. Note that the ASVs are not equipped with a sensing system.

5.2 Autonomous Surface Vehicles

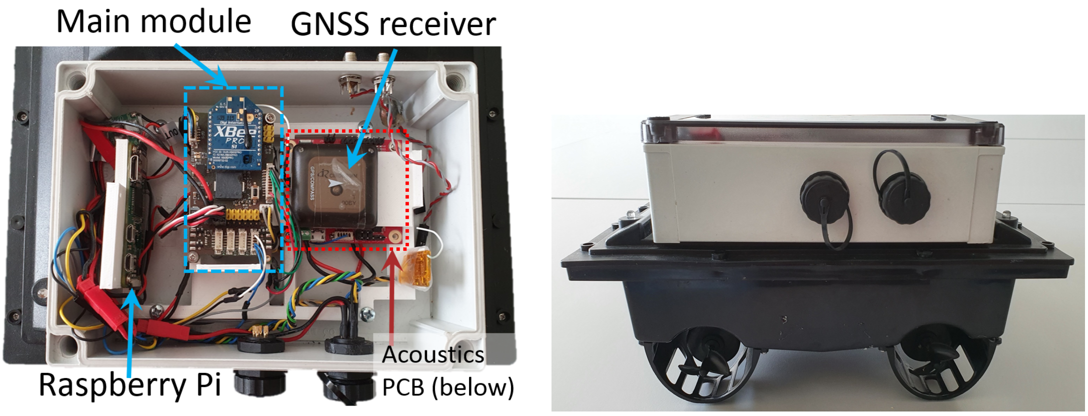
The Autonomous Surface Vehicles, shown in Fig. 5.2, were built in-house as a part of this thesis by outfitting pre-made hulls with our own electronics and making them autonomous. Being on the surface, the ASVs have continuous radio communication and GNSS reception. A separate radio module is used for manual remote-control. They weigh around 2.5 kg (with electronics and battery), and are 50 cm in length. They have a maximum cruise speed of about 2.5 m/s. A 3000 mAh battery provides an autonomy of about 4 hours under normal operation. The boat is buoyant enough for carrying a heavier, higher-capacity battery.

5.2.1 Mechanical Hardware

The ASVs are built using empty boat hulls with a pair of motor-propeller assemblies. The hulls, made by injection molding ABS plastic, were a part of a manual remote-controlled boat (made for deploying fish feed). This mechanical hardware assembly (hulls, propellers, motors) was bought off the shelf. We enclosed our electrical hardware in a waterproof box and mounted it on the top of the boat, as shown in Fig. 5.2b. At the bottom at the rear end, there are two independently driven motors with propellers, providing differential thrust for steering the boat (see Fig. 5.2c). Batteries are housed in the center of the hull, as shown in Fig. 5.2a.



(a) ASV top view.



(b) Housing for electronics.

(c) ASV rear view.

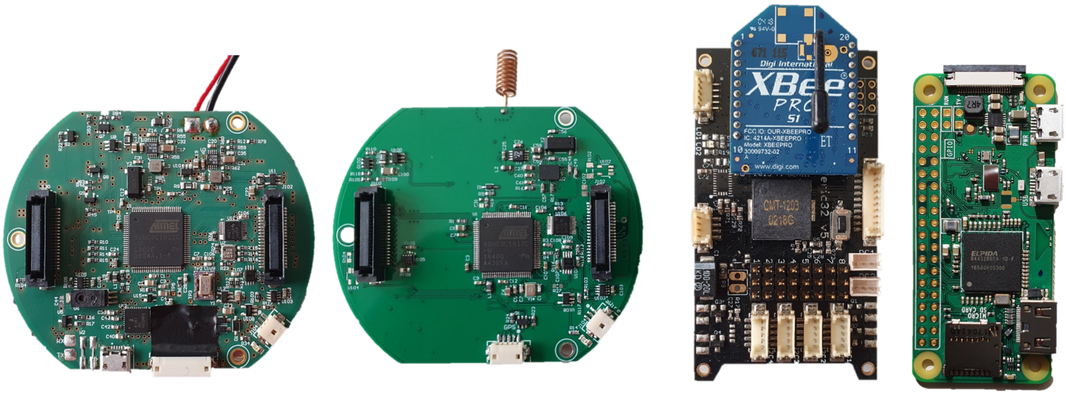
Figure 5.2: (a) Top view of the ASV showing the waterproof housing for electronics and fo battery. (b) Electrical hardware modules. (c) Rear view of the ASV showing the two propellers.

5.2.2 Electrical Hardware

Similar to the AUV, the ASV is also composed of several independent modules for different tasks, with each module being a separate PCB. In Fig. 5.2b, the main module that performs navigation and system management appears in the center. This module also hosts an IMU, a connection to a GNSS receiver, a radio communication module, and has connections to the two propellers. In addition, there is an acoustic transceiver module and a computation module. The ASV electrical hardware is similar to that of the AUV but with fewer components. Further details are presented in Section 5.3.

5.3 Hardware Architecture

The electrical hardware of both the AUV and the ASV consists of several independent modules, each for a separate task. Such a modular architecture offered several advantages, such as compact stacking of small PCBs in the AUV hull and easy swapping of modules for maintenance.



(a) AUV system management (left) and nav/comms (right) (b) ASV main board (c) Computation

Figure 5.3: Some of the modules that make up the AUV and ASV electrical hardware.

In addition, a modular architecture allowed us to replace one or more hardware modules with a simulated module, and perform partial hardware-in-the-loop emulation and testing.

The ASV hardware was designed to have an architecture similar to the AUV. This allowed us to use the same software on both robot platforms. Further, new modules were first tested by plugging them into the ASV for quicker design/test iterations. These mainly included the computation module, the acoustic transceivers (developed jointly with Hydromea), and the commercial acoustic transceiver that we integrated later.

Most modules are equipped with an Atmel AVR-32 series microcontroller. The only exception is the computation module, which is a Raspberry Pi board with an ARM processor. This section discusses the various modules and their organization in detail.

5.3.1 Modules

Most of the AUV hardware modules were developed with the Vertex AUV, while the ASV main module is a modified version of the MAV'RIC autopilot hardware [39]. The hardware (on both AUV and ASV) was modified during this thesis to replace the IMU unit with an XSENS MTi-3 IMU module and provide a connection to the newly added computation module.

AUV System Management The system management module on the AUV, shown in Fig. 5.3a, manages the power supply to all other modules, monitors their operation, detects and reports errors, if any. It has a memory card for logging data incoming from all the modules as well as runtime errors.

AUV Nav/Comms The navigation/communication module on the AUV is shown in Fig. 5.3a. It hosts an IMU, a radio transceiver module, a GNSS receiver and connects to the five thrusters. The navigation and localization algorithms run on this module. It also connects to the computation module.

ASV Main On the ASV, the system management and navigation/communication functions are combined on a single module, called the main module, pictured in Fig. 5.3b. Given that the AUV has fewer subsystems, a dedicated system management module was not necessary.

Computation A Raspberry Pi Zero W computer [40] (see Fig. 5.3c) provides additional computational capacity, given that the AVR-32 series microcontrollers on other modules have a limited one. It is equipped with a 1 GHz single-core processor and a 512 MB RAM. We chose this computer because of its small size, good community support, and availability of wireless Ethernet.

Acoustic Transceiver The acoustic transceiver module contains a microcontroller connected to the receiver and transmitter piezoelectric transducers via appropriate analog/digital conversion and amplification. The microcontroller performs modulation and demodulation of acoustic signals and measures time-of-flight for ranging. The packaging of the module is different on the AUV and ASV, as shown in Fig. 5.4. The software and modulation techniques were developed as a part of this thesis (see Chapters 10, 11).

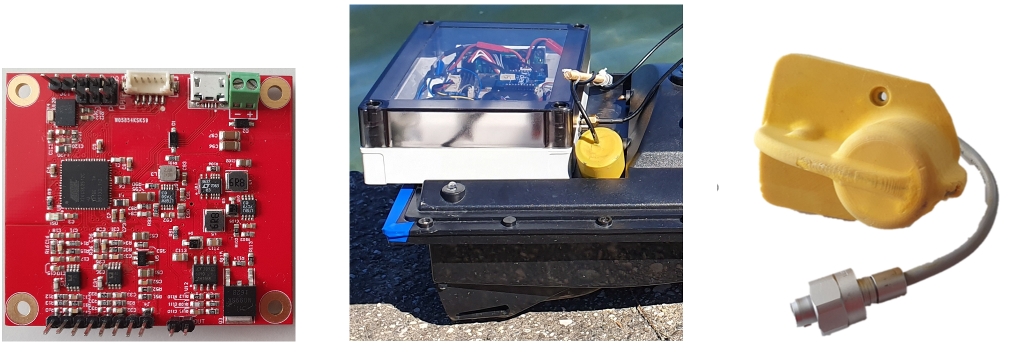
Thrusters/Propellers On the AUV, a dedicated CAN interface carries control commands from the navigation/communication module to the five thrusters (each of which has a separate speed controller). A waterproof external plug is used for the physical connection between the thruster-set and the AUV. On the ASV, each of the two propellers is driven by a brushed DC motor connected to an Electronic Speed Controller (ESC). The ESC takes in commands in the form of a Pulse Width Modulated (PWM) signal generated by the ASV Main module.

AUV Sensing System The sensing system module (only present on the AUV) consists of the YSI EXO sensing system (a separate product) and a PCB with a microcontroller that acts as an interface between the AUV and the YSI EXO. The sensor data are then relayed to the rest of the modules on the AUV in real-time.

AUV Backplane On the AUV, the backplane module is installed at the rear end and contains a depth sensor and a UART interface for an acoustic transceiver. Data from the depth sensor and the acoustic transceiver are relayed to the rest of the system.

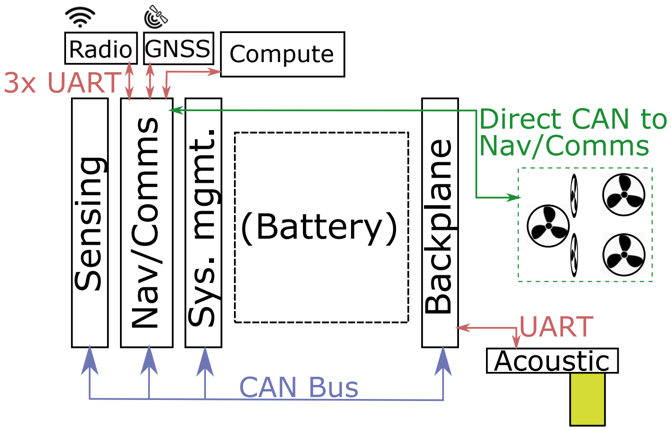
5.3.2 Interface Between Modules

On the AUV, a CAN bus connects the system management, navigation/communication, sensing system and backplane modules, and the thrusters. Data are broadcast, i.e., any data transmitted by any component are received by all other components. The compute module is connected to the navigation/communication module using a UART interface. Acoustic transceivers are connected to the backplane. Both, the navigation/communication and backplane modules act as a bridge between the CAN and their respective UART interfaces.

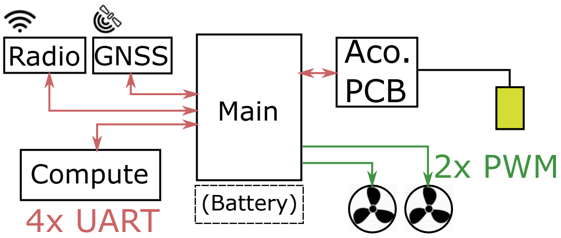


(a) ASV acoustic transceiver PCB (b) ASV transducer (c) Acoustic transceiver for AUV

Figure 5.4: Acoustic transceiver module on the AUV and ASV. (a) On the ASV, the module consists of a separate PCB housed inside a waterproof box, connected to an external transducer shown in (b). (c) The AUV acoustic transceiver has the PCB and transducer in a single waterproof package.



(a) AUV hardware modules and interfaces



(b) ASV hardware modules and interfaces

Figure 5.5: The interfaces between various modules on the AUV and the ASV. The 1-wire connection for the PPS signal from the GNSS receiver to all modules is not shown.

On the ASV, the acoustic transceiver and the compute module are connected to the main board using separate UART interfaces. The ESCs of the two propellers are connected to two PWM outputs.

In addition, the Pulse Per Second (PPS) signal from the GNSS receiver is connected to all the modules for clock synchronization. The schematic in Fig. 5.5 shows the connection between various components on the AUV and the ASV.

5.4 Software Architecture

The AUV and the ASV share the same software, making development easier. The software stack for the robots is separated into two layers. The application layer contains the functional and behavioral code, such as communication, control, navigation, and localization. The Hardware Abstraction Layer (HAL) contains drivers for hardware such as IMU, GNSS receivers, etc., as well as internal microcontroller peripherals such as timers and interrupt handlers. It provides a common set of routines to access the underlying hardware and makes the application layer agnostic to it. The HAL also abstracts away the different communication interfaces (UART or CAN) and provides channels for transmitting and receiving data. This makes it easy to use the same application layer code on different modules and different robots.

Each hardware module introduced in the previous section runs a separate instance of the aforementioned software stack, where the HAL runs drivers for the module-specific hardware, and the application layer runs tasks specific to the module. The software is written using a combination of C and C++. It runs bare-metal (without an intervening operating system) on modules with the AVR32 microcontrollers. It runs as a Linux application on the computation module, which uses the Raspbian operating system in the background. Data are shared between modules within the robot, and between robots, using a publish-subscribe mechanism.

The core of the software was initially derived from the MAV'RIC autopilot [39], but the software code has since diverged significantly from the MAV'RIC project and is highly customized for our robot platforms. While MAV'RIC project consisted of a single module per robot, our robots are composed of several hardware modules. A description of the components making up the application layer software is provided below.

Task Scheduler The central component of the application layer is a task scheduler that is responsible for executing tasks (which are `C++` functions) at specified intervals. The scheduler repeatedly checks for tasks that need to be executed at the current time. It keeps statistics of scheduling violations, which occur when a collection of tasks take too long to finish execution. Some example tasks include querying the IMU, calculating control outputs, updating the state estimator, and processing received data.

Message Marshalling We use MAVLink [41], an external, open-source software library, for packaging data to be stored or communicated. The library provides predefined formats for different kinds of messages (IMU data, GNSS data, robot status information, etc.). It includes Cyclic Redundancy Check (CRC) based error-detection using a 2-byte checksum. Each message is tagged with identifiers for the message type, the transmitting robot, and the internal robot module. These identifiers are used for subscribing to specific messages.

Internal Communication For internal communication between modules, modules publish packaged messages into the communication channel provided by the HAL, where each channel maps to a hardware communication interface (CAN or UART). These data are then received by all the other modules. The receiving modules only process messages that they subscribe to. The system management module logs all internal communication to the memory card.

Telemetry / External Communication For external communication, the navigation/communication module transmits specific messages (also packaged using MAVLink) using the on-board radio transceiver. It also forwards specified messages from the internal communication to the radio. All incoming external messages are forwarded to the internal communication channel, so all other modules receive them. Later in the project, the wireless Ethernet on the computation module was also added as an additional external communication interface.

Message Handler A message handler keeps track of the identifiers of all incoming messages (message type, sending robot, and module ID). It provides a mechanism for specifying 'handling routines' that subscribe to specific incoming messages and execute some code when the messages arrive.

5.5 Contributions

Several components of the Vertex AUV were developed in our lab before this thesis and produced by Hydromea, while those of the ASV were bought off-the-shelf or designed to be similar to the AUV. Others, however, were modified and enhanced as a part of this thesis, sometimes in collaboration with our colleagues or with Hydromea. Table 5.1 shows a module-wise delineation between elements that were acquired and the work done during this thesis.

HW Module	Hardware	Software
System Management	Used as blackbox	Mostly used as blackbox. Developed: added message handling for new messages, software maintenance, and bug-fixes.
Navigation / Communication	Modified in collaboration with lab colleagues: added new radio transceiver, GNSS module, interface to computation module. Modified in collaboration with Hydromea: added new IMU module.	In collaboration with lab colleagues: sensor hardware drivers, attitude control. Developed independently: navigation, state estimation, localization, adaptive sampling, data interface with computation module.
Sensing system	Used as blackbox.	Used as blackbox.
Backplane	Used as blackbox.	Used as blackbox. Added drivers and message forwarding for acoustic transceivers.
Acoustic transceivers	Acquired from Hydromea, minor modifications in collaboration with Hydromea.	Acquired: drivers for producing/sampling acoustic signals, timing. Developed independently: modulation, demodulation, data handling, communication with other modules.
Computation	Reviewed/tested various available options. Acquired Raspberry Pi Zero W.	Developed independently.
ASV main board	MAV'RIC autopilot with minor modifications. Replaced IMU, added interfaces to acoustics module, computation module.	Similar to AUV Sys. mgmt. and nav/comms. Developed independently: same developments as AUV nav/comms hardware module.

Table 5.1: Module-wise contributions towards design and development of the robot platforms. Some hardware modules were used as they were, as blackboxes. Others were modified/developed either in collaboration with colleagues Dr. Alexander Bahr and Dr. Felix Schill, in collaboration with Hydromea, or independently.

6 Experimental Setup

Experiments in realistic conditions are fundamental to developing algorithms and methods ready for real-world deployment. One of the aims of this work is to develop a cooperative underwater multi-robot system for outdoor, real-world applications. However, performing experiments in outdoor environments is challenging. In addition to logistical and weather-related constraints, dynamic outdoor conditions imply that conducting repeatable experiments is nearly impossible. Aquatic environments are also complex, and an accurate simulation of their characteristics is difficult to achieve.

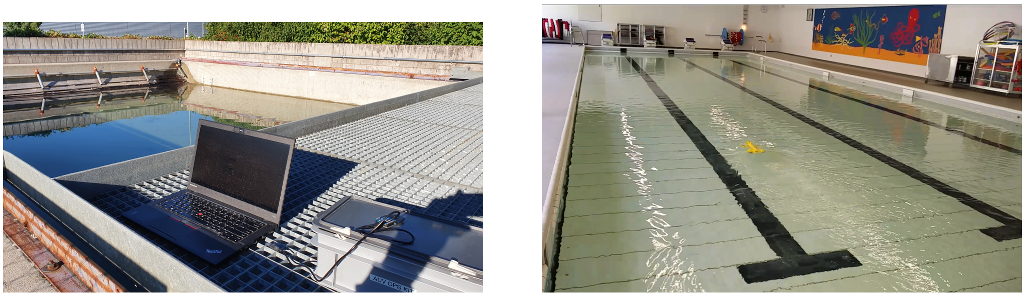
In light of the aforementioned challenges, we took a multi-faceted approach to perform experiments. Firstly, depending on the purpose of particular tests and experiments, we used different experimental arenas, including lakes. Then, to facilitate smooth operation, command and control of the robots, we developed a ground station software tailored to our experiments. Finally, we developed a tiered framework for simulation, where different combinations of simulated phenomena and real-world data are used for evaluation of our algorithms. This chapter outlines our testing and experimental setup.

6.1 Testing and Experimental Arenas

We used several different arenas for either performing tests for a narrow set of robot subsystems or for full-fledged experiments. Each of the arenas offered different trade-offs between logistical ease of performing experiments and closeness to real-world conditions.

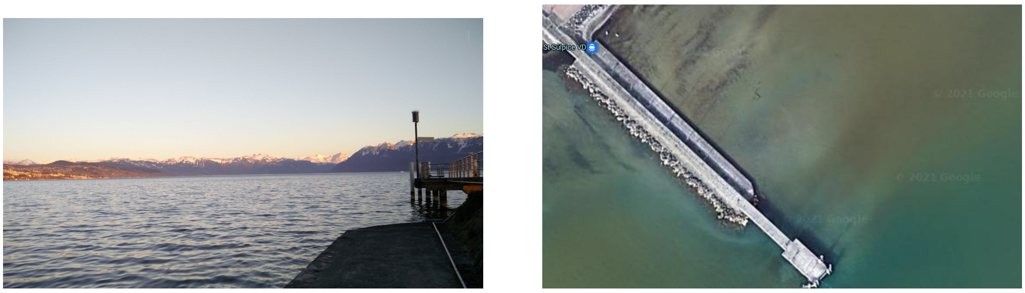
Swimming Pools and Tank

We performed preliminary tests for tuning thrust and attitude controllers of the AUVs in a tank and a swimming pool. We had access to a tank at the Platform of Hydraulic Constructions (PL-LCH) at EPFL, which measures $4 \times 4\ \mathrm{m}^2$ and had a depth of 2 m. In addition, we used public swimming pools in Lausanne. Pictures of these facilities are shown in Fig. 6.1. The



(a) (b)

Figure 6.1: (a) Water tank at the Platform of Hydraulic Constructions (PL-LCH) at EPFL. (b) A swimming pool. Both facilities were used for tuning low-level controllers and functional tests of the AUV.



(a) (b)

Figure 6.2: (a) Boat pier from which robots were launched. (b) A satellite view of the shore area (Courtesy: Google Maps). The depth of the water column in this region is between 2-10 m, and it varies by season.

advantage of these facilities was that robot failures were not a cause for concern, since they could be recovered easily. However, the size of the arena was too small for most experiments. Further, the smooth and hard walls acted as reflective surfaces for acoustic signals, causing echoes and interference.

Lake Shore

Many of our experiments were performed near the shore of Lake Geneva at a boat pier near EPFL, shown in Fig. 6.2. The advantage of this experimental location was that it was easy to access and offered a convenient, fixed launch point for the robots. However, shallow water presented disadvantages. It limited the vertical motion of the AUV, and the conditions were not ideal for acoustic transceivers. Robot malfunctions also posed a risk, since robots were not easily reachable from the launch point, and required swimming or using a boat to retrieve the robots. Note that experimental activity was hindered by windy or rainy weather, by ferries passing by, and by swimmers in the Lake.

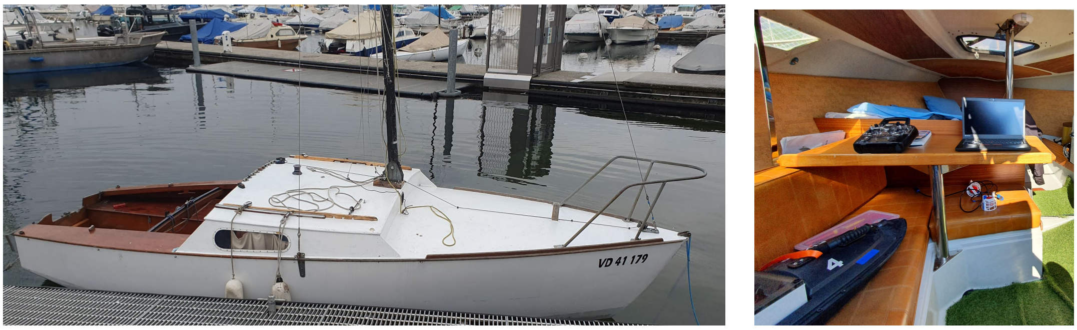
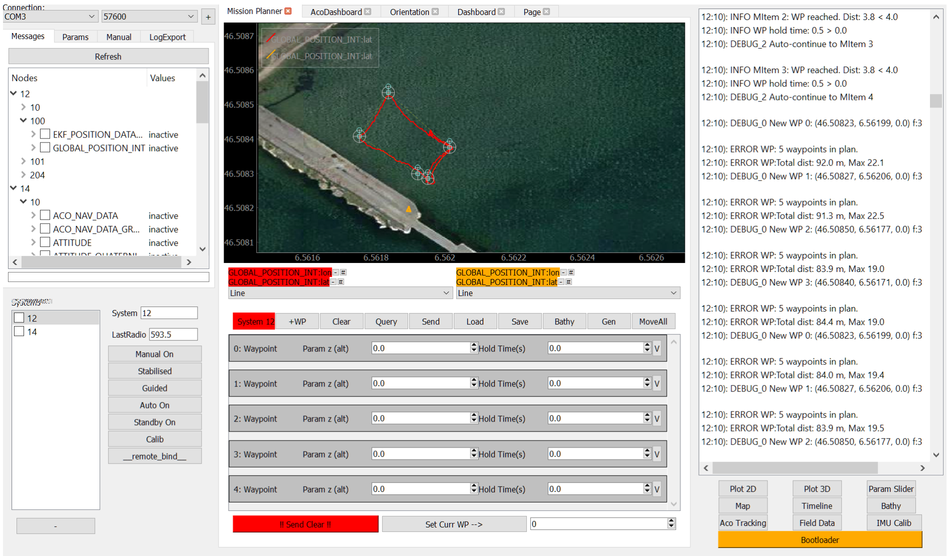

(a) (b)

Figure 6.3: (a) Corsaire Amiguet, the 5 m boat, set up by our colleague Mr. Droz., which was used as a base station for launching robots. (b) Desk space used for preparation and mission planning.



Connection: COM3 57600

System 12

LastRadio 593.5

```
12:10): INFO MItem 2: WP reached. Dist: 3.8 < 4.0
12:10): INFO WP hold time: 0.5 > 0.0
12:10): DEBUG_2 Auto-continue to MItem 3

12:10): INFO MItem 3: WP reached. Dist: 3.8 < 4.0
12:10): INFO WP hold time: 0.5 > 0.0
12:10): DEBUG_2 Auto-continue to MItem 4

12:10): DEBUG_0 New WP 0: (46.50823, 6.56199, 0.0) f:3

12:10): ERROR WP: 5 waypoints in plan.
12:10): ERROR WP:Total dist: 92.0 m, Max 22.1
12:10): DEBUG_0 New WP 1: (46.50827, 6.56206, 0.0) f:3

12:10): ERROR WP: 5 waypoints in plan.
12:10): ERROR WP:Total dist: 91.3 m, Max 22.5
12:10): DEBUG_0 New WP 2: (46.50850, 6.56177, 0.0) f:3

12:10): ERROR WP: 5 waypoints in plan.
12:10): ERROR WP:Total dist: 83.9 m, Max 19.0
12:10): DEBUG_0 New WP 3: (46.50840, 6.56171, 0.0) f:3

12:10): ERROR WP: 5 waypoints in plan.
12:10): ERROR WP:Total dist: 84.4 m, Max 19.0
12:10): DEBUG_0 New WP 0: (46.50823, 6.56199, 0.0) f:3

12:10): ERROR WP: 5 waypoints in plan.
12:10): ERROR WP:Total dist: 84.0 m, Max 19.4
12:10): DEBUG_0 New WP 1: (46.50827, 6.56206, 0.0) f:3

12:10): ERROR WP: 5 waypoints in plan.
12:10): ERROR WP:Total dist: 83.9 m, Max 19.5
12:10): DEBUG_0 New WP 2: (46.50850, 6.56177, 0.0) f:3
```

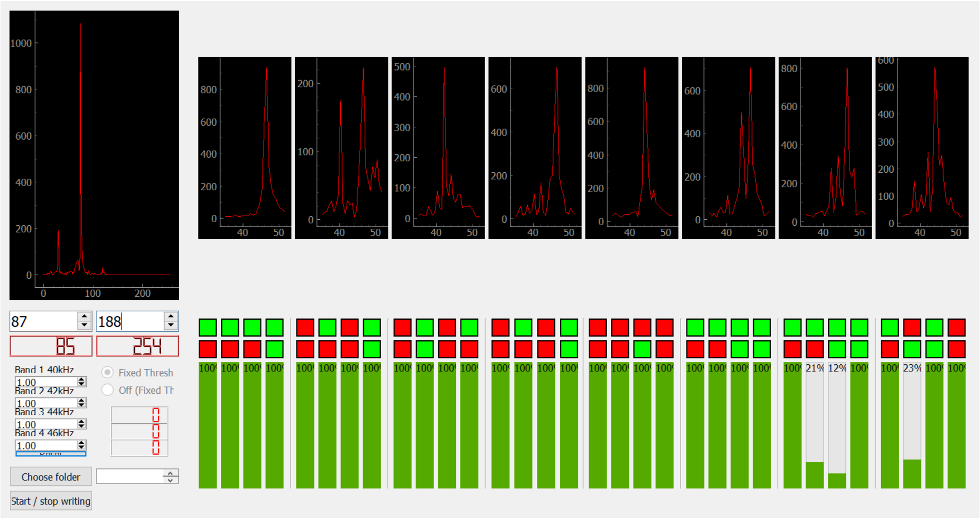
Figure 6.4: Screenshot of the base station software.

Boat

Experiments were also performed by launching robots from a boat, shown in Fig. 6.3. We used a 5 m long boat with a small cabin that provided space for setting up the base station laptop, and a mast that was used to hoist a radio antenna. The boat allowed us to do experiments at any location in the lake and made it easy to recover malfunctioning robots if they were on the surface. However, the logistics were more complex compared to lake-shore experiments, and activity was constrained by bad weather.

6.2 Base Station Setup

The base station consisted of a laptop running base station software, a radio transceiver (for communicating with robots), and an acoustic transceiver (used in some acoustics experi-



87 188

85 254

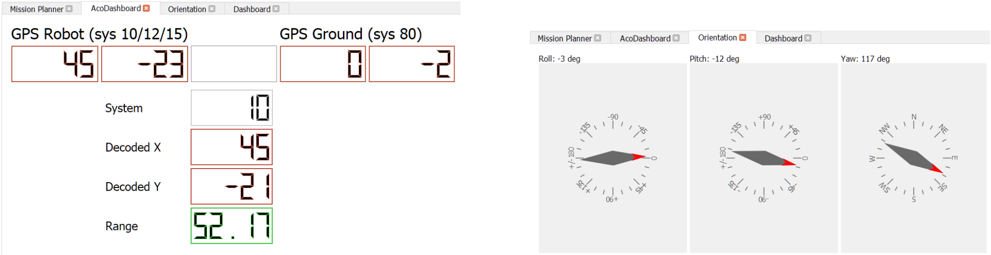
Band 1 40kHz 1.00

Band 2 42kHz 1.00

Band 3 44kHz 1.00

Band 4 46kHz 1.00

(a) Interface for monitoring acoustic communication. It shows the signal strength of individual carrier frequencies and compares the expected and received stream of bits.



GPS Robot (sys 10/12/15) 45 -23

GPS Ground (sys 80) 0 -2

System 10

Decoded X 45

Decoded Y -21

Range 52.17

Roll: -3 deg

Pitch: -12 deg

Yaw: 117 deg

(b) Interface for monitoring acoustic localization, displaying range, and communicated beacon position.

(c) Interface for displaying attitude estimator output, showing roll, pitch, and yaw angles.

Figure 6.5: Interfaces used for monitoring specific subsystems of the AUV.

ments). The base station was used for command-and-control and monitoring of the robots using the telemetry data broadcast by them.

While there were a number of open-source software applications for robot telemetry and control, they were usually designed for single, aerial robot operations. We did not find alternatives tailored to a team of aquatic robots, and therefore, developed our own software. It was written in Python and used the MAVLink library for message packaging (the same library used by the robots). A screenshot of the software is shown in Fig. 6.4.

The software is used for uploading mission plans (list of waypoints and actions), modifying parameters on-the-fly (cruise speed, PID parameters for various controllers, etc.), and monitoring telemetry information (position, battery level, robot health, etc.). Note that most monitoring features work only when robots are on the surface. When the robots are submerged, only a limited amount of information can be sent via the acoustic channel.

The software also includes specialized interfaces for specific aspects of the robot subsystems, so that all the important information related to the ongoing experiment is prominently dis-

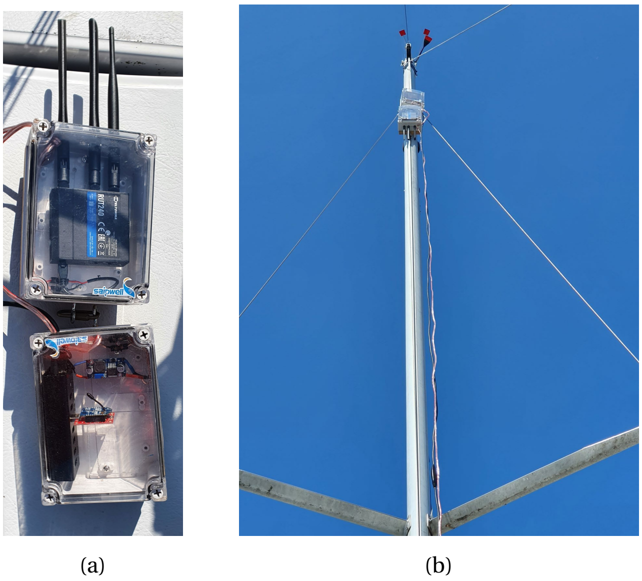

(a) (b)

Figure 6.6: (a) Base station radio receiver, comprising of an XBee receiver and a wireless ethernet hub. (b) The radio setup hoisted on the sail mast of the boat.

played. These help better understand the functioning of those subsystems and help quickly identify errors and malfunctions. Fig. 6.5 shows some such interfaces. The interface used for monitoring acoustic signal modulation and demodulation is shown in Fig. 6.5a. It highlights erroneous bits and helps understand if certain carrier frequencies are more prone to bit-errors. Fig. 6.5b shows the interface for acoustic ranging experiments, where the positions of the robots and the measured range is displayed. Fig. 6.5c shows the interface used for monitoring attitude estimation.

The radio transceiver actually comprises two radio interfaces – an XBee transceiver as well as a wireless Ethernet hub. This is installed in a waterproof housing shown in Fig. 6.6. On the boat, this housing is hoisted on the sail-mast. That offered a better line-of-site with ASVs, and hence, better communication throughput over a longer range.

6.3 Simulation and Emulation

Given the complexity and effort involved in outdoor, multi-robot experiments, simulations are essential for testing and evaluating algorithms under a variety of conditions. Over the course of the thesis, depending on what aspect of the robot is to be evaluated, we set up different tiers of evaluation, combining varying levels of reality and simulated conditions. To begin with, we identify four different parts of the application layer of the robot software, namely

- functional logic code (includes telemetry broadcast, waypoint handling, data management etc.),
- estimation algorithms,

- high-level control and coordination algorithms, and
- low-level control algorithms.

Simulating low-level control algorithms would require a simulator with a high degree of accuracy. They are tested and evaluated in the tank or swimming pool, as described in Section 6.1. A brief introduction to the various simulation/emulation tiers is provided below. They are explained in detail in relevant chapters, when such simulation was used for evaluating newly developed methods.

For emulation, we execute the relevant software or algorithms on the real system or with realistic data inputs. For simulation, we use a robotics simulation software to model real-world environments and evaluate the performance of our algorithms within them.

Hardware-in-the-Loop Emulation HIL emulation is used to test functional logic code on a particular module. It is performed by isolating a hardware module, feeding it synthetic data (instead of data from sensors or other modules), and executing the software. The purpose is to quickly identify software bugs such as data leaks, memory overflows, and to verify whether all code executes on robot hardware respecting real-time constraints. Note that this is purely a test for software and computational budget and not an overall evaluation of any robot application.

Data-in-the-Loop Emulation DIL emulation is used to test and evaluate the estimation algorithms. The application layer code is executed, and sensor measurements are replaced by data from a recorded log file from a real experiment. Then, additional simulated measurements, noise, errors, or measurement failures are injected into the log file. This allows us to evaluate the system under a variety of conditions, while still preserving characteristics of real-world environments. Since DIL is performed on a (generally more powerful) desktop computer, the execution is quick. This facilitates faster iterations between algorithm development and testing. DIL emulation is discussed in detail in Chapter 10.3.

High-Fidelity Simulation We used Webots [42], a high-fidelity, physics-based robotics simulation software for simulating multi-robot experiments. This simulation is used for evaluating high-level control and coordination algorithms. Webots is not inherently an aquatic robotics simulator but only provides a basic simulation of objects in an aquatic medium. We therefore calibrate the simulation by using data from real-world experiments. Chapter 14 describes the setup of this simulation tier in detail and explains the calibration to real-world conditions.

Single-Robot Subsystems Part III

7 Introduction

UNDERWATER robotics is characterized by scarcity of external information owing to the physical properties of water. This makes both, navigation and communication, a challenge. The subsequent chapters in this part of the thesis focus on developing navigation and communication systems for individual robots before moving on to multi-robot systems. In these chapters, we attempt to exploit available external information for improving localization accuracy.

7.1 Outline

We begin this part with **Chapter 8**, which provides a detailed description of the state estimation and basic control of individual robots. We employ the commonly used Extended Kalman Filter (EKF) for state estimation, coupled with a model of dynamics of the robot. The robots are programmed to travel towards specified waypoints in succession. No results are immediately presented within this chapter, however, the inertial trajectory estimated during robot missions are shown with the relevant results in the subsequent chapters. This chapter combines content from several publications listed below.

Chapter 9 presents data gathered during an adaptive environmental sensing mission performed in Lake Cadagno, a Swiss mountain lake. We demonstrated the ability of the robot to track and trace a thin stratified layer rich in photosynthetic bacteria. This mission was performed before the acoustic system for our AUVs was developed, and as a result, the estimated position of the AUV was based purely on inertial measurements. The experience from such missions in this and several other lakes shaped the work on localization and multi-robot coordination in the rest of the thesis. This chapter is based on the following publication:

1. A. Quraishi, A. Bahr, F. Schill, and A. Martinoli, "Autonomous Feature Tracing and Adaptive Sampling in Real-World Underwater Environments", in *IEEE International Conference on Robotics and Automation*, 2018, pp. 5699–5704.

Co-authors Dr. Alexander Bahr and Dr. Felix Schill contributed towards the development of the AUV, integration of the environmental sensing system, and development of the fast temperature sensor module. They also participated in organizing the sampling missions.

In **Chapter 10**, we introduce a miniature acoustic ranging system developed in this work. The system uses absolute time information from GNSS for initial clock synchronization, which allows measurement of range from one-way transmissions. Static beacons at known locations are then able to provide localization support to multiple robots.

Later, in **Chapter 11**, we enhance the system by embedding data in acoustic signals. The beacons could then transmit their own positions and no longer needed to be static. In this chapter, we also introduce a small-sized, commercially available underwater acoustic transceiver.

In both chapters, results from the experimental evaluation of the system are presented. Experiments were performed with ASVs near the shore of Lake Geneva. The two chapters are based respectively on the following publications:

2. A. Quraishi, A. Bahr, F. Schill, and A. Martinoli, "Easily Deployable Underwater Acoustic Navigation System for Multi-Vehicle Environmental Sampling Applications", in *IEEE International Conference on Robotics and Automation*, 2019, pp. 3464–3470.
3. A. Quraishi, A. Bahr, F. Schill, and A. Martinoli, "A Flexible Navigation Support System for a Team of Underwater Robots", in *IEEE International Symposium on Multi-Robot and Multi-Agent Systems (MRS)*, 2019, pp. 70–75.

Co-authors Dr. Alexander Bahr and Dr. Felix Schill contributed towards the development of the acoustic transceiver hardware and firmware and the clock synchronization system.

Finally, in **Chapter 12**, we present a framework for estimating kinematic and dynamic model parameters used for inertial navigation. Accurate values of these parameters result in better trajectory estimation. Our framework is also able to correct errors in the past trajectory by using recently updated parameter estimates. By doing so, we are able to provide improved geo-references for past as well as future spatial measurements. This has an impact on adaptive sampling methods, which use geotagged measurements for building local spatial distributions and choose future sampling points. We present the results of an evaluation of the presented method with experiments in a lake. This chapter is based on the following publication:

4. A. Quraishi and A. Martinoli, "Online Kinematic and Dynamic Parameter Estimation for Autonomous Surface and Underwater Vehicles", in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2021. To appear.

7.2 Related Work

This section reviews the existing work related to our contributions covered in this part of the manuscript.

7.2.1 Underwater Robotic Sensing

Although the prior literature on adaptive sampling with underwater robots is less rich than that with ground or aerial equivalents, a number of works that describe such methods with underwater robots do exist. Zhang et al. [43] employed an AUV to detect a coastal upwelling front separating a stratified and an upwelling water column and programmed the AUV to travel between the two regions in order to track the horizontal position of the front in one dimension over time. They use differences in temperature at two fixed depths during 'yo-yo' AUV trajectories as a feature to distinguish regions on either side of the boundary. Petillo et. al. in [44] proposed a method for 3D front tracking and demonstrated it with simulations. In [45], a drifter is used to track the location of an interesting feature such as an algal patch in flowing water. An AUV then executes a pre-planned survey pattern relative to the drifter position. An isotherm tracking method is demonstrated in [46], where the AUV performs a saw-tooth trajectory within a temperature envelope instead of a depth envelope. In [47], the AUV adjusts its buoyancy to control its depth and stay within a temperature bracket and track an isotherm. It does not use its propellers for motion during this time but drifts with the flow of water. In [48], the robots process their past measurements online and adapt their future path along which new measurements will be recorded. In similar vein, two surface robots are used in [49], where one robot performs in-situ measurements and suggests potential sampling locations. The second robot then gathers water samples at these locations for laboratory analysis.

In our work, our main goal is to locate and trace a stratified layer that is rich in photosynthetic bacteria. We use chlorophyll and turbidity measurements to detect the layer. Then, similar to the work in [47], we attempt to drift through the layer and collect in-situ measurements without disturbance from propellers.

7.2.2 Localization with Acoustic Signals

The most common approach to underwater acoustic localization is to measure a geometric quantity (range or direction) with respect to a beacon at a known location [50]. This is then used to deduce robot positions, for instance, by trilateration.

One main categorization of acoustic localization methods is based on how range is calculated. In [51], Miller et. al. use round-trip time for measuring range to beacons at known positions, combined with a Kalman filter for fusion. However, an approach based on round-trip time requires all entities (including robots) to actively transmit a signal. On the other hand, synchronized clocks, used for example in [52], [53] eliminate the need for two-way signal exchange for ranging. By requiring only beacons (and not robots) to transmit a signal for range measurements, they make the system scalable in the number of robots. We have adopted a similar approach for range measurements based on one-way-travel-time.

Yet another important characterization is based on what geometric quantity is used for localization. In addition to (or instead of) range measurements, the angle of arrival of a signal can also be deduced using time-difference-of-arrival of an acoustic signal at a collection of geometrically displaced receivers. This set of receivers is called an Ultra Short Base Line (USBL) receiver array. Rypkema et. al. demonstrate such an approach in [52]. They combine one-way ranging with bearing measurements. Bearing measurements are also used in [54]. However, a multi-receiver setup adds complexity, both mechanically as well as for signal processing. On the Vertex AUV, there is limited space for adding a USBL array and the additional cabling that will accompany it.

If acoustic communication is available, the beacons can transmit their own position. Munafò et al. in [55] used acoustic communication for exchanging local timing information between static acoustic beacons and AUVs. Their work relaxes the need for synchronized clocks, which are difficult to implement when permanently installing static beacons below the surface. In our work in Chapter 11, communication relaxes the need for beacons to be static.

A few works in literature address related problems. In [56], the authors consider the case where the beacon locations are not known (but they are known to be fixed). They use an approach similar to Simultaneous Localization and Mapping (SLAM), where the beacons and the AUV are simultaneously localized. Becker et al. in [54] propose a similar approach using bearing-only measurements from a single beacon with known depth but unknown position. Since we were unable to install any equipment in the lake, we mounted our beacons on ASVs, which were mobile. Then, our problem was to either keep them static (by actively holding a specified position), or to continuously broadcast beacon position.

A number of underwater localization methods that do not rely on acoustic beacon systems do exist. They include matching measured geophysical properties to apriori maps [57], and SLAM-like approaches using a SONAR [58] or vision [59]. Others use environmental information (such as ocean currents) measured using a Doppler Velocity Log (DVL) or an Acoustic Doppler Current Profiler (ADCP) to aid inertial navigation [60] [61]. They either need apriori information about the environment or require expensive and large equipment such as a SONAR, which cannot be integrated within a miniature AUV. Further, visual SLAM-based methods work only in clear, shallow water or close to the bottom.

7.2.3 Model Parameter Estimation

If the model describing a robot's kinematics or dynamics is used either for estimation or for control, estimating the true value of the parameters describing the model is important. Indeed, when using MEMS-based miniature inertial sensors which have low accuracy, fusion with a model of the robot enhances the accuracy of inertial localization [62]. Directly measuring or estimating some of these parameters can be difficult, time-consuming, or require special equipment. Therefore, some approaches in literature attempt to estimate these parameters by exploiting data from robot operations.

Kelly et. al. in [63] estimate geometric and camera parameters using camera images and IMU measurements. They add the parameters to be estimated as state variables to an Unscented Kalman Filter (UKF) framework. A similar approach is presented in [64]. An approach based on error-minimization for sensor calibration is presented in [65], which is applied offline to recorded datasets. Authors in [66] take an information-theoretic approach to assess the usefulness of robot operation data before using them for re-calibration. The latter two contributions perform parameter identification offline. All of the above works leverage camera images, which are used to deduce relative pose between successive measurements.

Approaches such as [67]–[69] attempt to estimate dynamic or inertia parameters, usually to be fed into a model-predictive control module. In [67], the authors rely on an unspecified absolute pose sensor to estimate parameters using an Extended Kalman Filter (EKF). In [69], a motion capture system is used to observe absolute robot poses. Burri et. al. in [70] use a Maximum Likelihood approach for parameter identification, which is performed offline. They use a camera, IMU, and an external tracking system.

Online methods such as [63], [67] often estimate parameters as additional state variables in an EKF framework. A drawback of such approaches is that the parameters are rarely directly observed, especially if the measurements are sparse. To cope with this partial observability, ad-hoc workarounds such as inclusion of intermediate state variables are necessary.

A contrasting feature of our work in aquatic environments is the sparsity of external measurements pertaining to robot pose. To deal with this sparsity, we leverage data recorded over a relatively wide time window. In essence, we exploit the fact that memory and computation are less scarce than external positioning information in underwater environments.

8 State Estimation, Control and Navigation

AUTONOMOUS mobile robots performing a task in any environment must be able to infer their present state and decide on future actions. This chapter focuses on subsystems of a single robot pertaining to state estimation and control. These are implemented as application layer tasks on the navigation/communication module.

A major part of this chapter is dedicated to the state estimation framework, which uses an EKF for estimating position and other state variables. It presents a detailed description of the process and measurement models. This is followed by a description of the basic control systems for attitude and depth.

8.1 State Estimation

We employ an EKF based state estimation framework for localization, which fuses sensor measurements with a model of the dynamics of the robots. The dynamics model takes into account propeller thrust, inertia of the robots, as well as effects of buoyancy and viscous drag. In the absence of external positioning information, dead-reckoning is performed using the model, aided by the IMU measurements (for acceleration and attitude), a compass (for heading), and a depth sensor. The EKF is complemented by GNSS position fixes when the AUV is on the surface, and acoustic measurements when they become available. The measurement updates from the various sources are not synchronized. Therefore, we use independent measurement models for each, making our EKF framework modular.

The state estimation framework was initially implemented on the navigation/communication module. The choice of a Kalman filter offered several advantages, particularly low computational complexity and simplicity of implementation. We explored various flavors in the Kalman filter family, such as Unscented KF and Sigma-Point KF, but converged on the EKF. This was due to the simplicity in implementation of the EKF, and the other flavors not offering a significant improvement in estimator performance despite increase in computational cost.

Given that the embedded system had limited computation capability, we performed uncertainty propagation at a slower rate than state update. Later in the project, the state estimation framework was moved to the computational module, which has higher computational power. However, the existing EKF framework was retained, except for now performing uncertainty propagation at full rate.

8.1.1 Coordinate Systems

In our calculations, we use two different coordinate systems – a ground-anchored, local level frame with an NED (North-East-Down) axes convention, and a robot-body-anchored moving frame whose axis definition is shown in Fig. 8.1. We denote the frame of reference of a vector by a subscript 'g' and 'b' for ground-frame and robot-body-frame, respectively. Rotations are represented using the quaternion notation.

8.1.2 State Formulation

The state variable comprising of the position and velocity in ground frame and acceleration in the body frame is formulated as

$$\mathbf{X} \triangleq \left[\vec{x}_{\mathrm{g}}^{\mathrm{T}}, \vec{v}_{\mathrm{g}}^{\mathrm{T}}, \vec{a}_{\mathrm{b}}^{\mathrm{T}}\right]^{\mathrm{T}} \tag{8.1}$$

We do not include the variables representing attitude since they are not predicted by the model in this work. Integrated attitude is directly provided by the Xsens IMU.

Note that the IMU internally uses the acceleration due to gravity as a reference for correcting drift in attitude along the roll and pitch axes. Similarly, the drift in yaw angle is corrected using the magnetometer. When not using the magnetometer, the yaw angle is unreferenced and assigned an initial value of zero. Then, a correction θ_e needs to be applied, comprising of the initial yaw angle as well as drift in yaw estimate over time. The corrected attitude is computed as

$$\mathbf{q} = \mathbf{q}(\theta_e)\mathbf{q}_{\mathrm{raw}}, \tag{8.2}$$

where $\mathbf{q}(\theta_e)$ is the quaternion corresponding to θ_e. We address the computation of this yaw correction later in Chapter 12.

8.1.3 Dynamics Model

In between acoustic or GNSS updates, a dynamics model and IMU measurements are fused using an EKF framework for inertial localization. We use a simplified dynamics model that leverages motor commands to compute thrust, and takes into account mass, inertia, and hydrodynamic drag forces. Fig. 8.1 shows the forces acting on the robots.

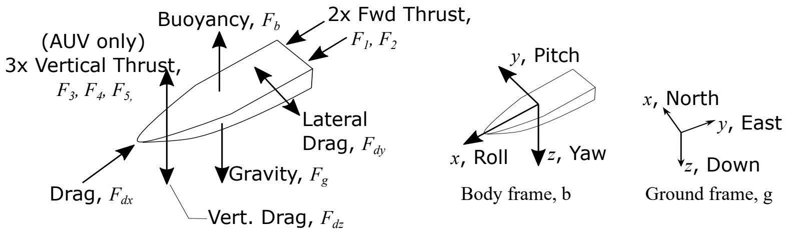


Figure 8.1: Left: Forces on the robots (AUV or ASV). Right: Body frame and ground frame (NED) coordinate systems.

Linear Dynamics

First, motor commands, $[u_1, u_2, ...u_i]$ are sent to the motors in terms of the desired rotational speed of each propeller. We compute the thrust produced by each propeller i as

$$F_i = ku_i^2, \tag{8.3}$$

where k is the thrust constant of the propellers, and $i \in [1,2]$ for the ASV, $i \in [1,5]$ for the AUV. All propellers on a particular robot are identical. The net thrust force is

$$\vec{F}_{\text{thrust}} = \sum F_i \hat{u}_i, \tag{8.4}$$

Where $\hat{u}_i$ is unit vector along the direction of propeller i. We calculate the forces due to viscous drag on the robot body in the robot frame $\vec{F}_{\text{drag}}$ as

$$\vec{F}_{\text{drag}} = \begin{bmatrix} -\text{sign}(v_{rx}) \cdot k_x v_{rx}^2 \\ -\text{sign}(v_{ry}) \cdot k_y v_{ry}^2 \\ -\text{sign}(v_{rz}) \cdot k_z v_{rz}^2 \end{bmatrix}, \tag{8.5}$$

where k_x, k_y, k_z are the viscous drag constants along the body axes and $\vec{v}_{r,\text{b}} = \left[v_{rx}, v_{ry}, v_{rz}\right]^{\text{T}}$ is the relative velocity between the robot and water stream, in the body frame. Essentially, in each direction, the drag force is proportional to the square of this relative velocity and acts in a direction opposite to it. $\vec{v}_{r,\text{b}}$ is computed by appropriate rotation of the said relative velocity in the ground frame as

$$\vec{v}_{r,\text{b}} = \mathbf{q}^{-1}\left(\vec{v}_{\text{g}} - \vec{s}_{\text{g}}\right)\mathbf{q}, \tag{8.6}$$

where $\vec{v}_{\text{g}}$ is the estimated velocity of the robot and $\vec{s}_{\text{g}} = \left[s_x, s_y, s_z\right]$ is the water stream velocity, both in the ground frame. $\mathbf{q}$ is the quaternion representing robot attitude, and the operation $\mathbf{q}^{-1}(\vec{v})\,\mathbf{q}$ rotates the vector $\vec{v}$, transforming it from ground frame to body frame representation.

We assume that the drag constants k_y and k_z are equal, and we refer to it with k_{yz}. This is warranted since the AUV geometry along those axes is largely similar, and ASV has no vertical motion. We also assume $s_z = 0$.

The AUVs are deliberately trimmed to be slightly positively buoyant, and ASVs are floating devices. We assume that the weight of the robot nearly cancels out buoyancy, and hence, both terms are ignored. The net force is then a function of five quantities.

$$\vec{F}_{\text{net,b}}(k, k_x, k_{yz}, s_x, s_y) = \left(\vec{F}_{\text{thrust}} + \vec{F}_{\text{drag}}\right). \tag{8.7}$$

Rotational Dynamics

The shape of the AUV is streamlined only along its longitudinal (x) axis. As a result, rotation about all three axes is highly damped. Further, the AUV is trimmed in a way that the centers of mass and buoyancy coincide. As such, the rotational motion of the AUV is slow enough that the update rate of the IMU (100 Hz) is sufficient to compute orientation. Therefore, we do not consider rotational forces in the dynamics model, instead relying only on the orientation computed internally and reported by the Xsens IMU module. A correction may be applied to the yaw orientation, as explained in Chapter 12.

8.1.4 Measurement Models

We use independent measurement models for each sensor. This is necessary for the incorporation of sensor measurements that arrive at irregular intervals into the state estimate. It also makes it easy to integrate new types of sensors into the EKF framework. For each sensor, the measurement model consists of a function $\hat{z} = g(\mathbf{X})$ that relates the state variable and the sensor measurement. We also derive the corresponding Jacobian of $g(.)$, which is used by EKF for uncertainty propagation.

IMU, GNSS, Depth

The measurement models for IMU, GNSS and depth are straightforward since the measured variables are also a part of the state. For the IMU measurement, we have

$$g_{\text{IMU}}(\mathbf{X}) = \left[\vec{a}_b^{\text{T}}\right]^{\text{T}}, \tag{8.8}$$

and the corresponding Jacobian,

$$G_{\text{IMU}} = \begin{bmatrix} \mathbf{0}_{3\times 6} & \mathbf{I}_{3\times 3} \end{bmatrix}, \tag{8.9}$$

where $\mathbf{0}_{m\times n}$ is the zero matrix of dimension $m \times n$, and $\mathbf{I}_n$ is an $n \times n$ identity matrix.

Similarly, for GNSS position (which is in 2D), we have

$$\begin{aligned} g_{\text{GNSS}}(\mathbf{X}) &= \left[\vec{x}_{\text{g},x}, \vec{x}_{\text{g},y}\right], & (8.10)\\ G_{\text{GNSS}} &= [\mathbf{I}_{2\times 2}, \mathbf{0}_{2\times 7}], & (8.11) \end{aligned}$$

where $\vec{x}_{\text{g},x}, \vec{x}_{\text{g},y}$ are the x and y coordinates of the estimated position (in the ground frame).

For depth, we have

$$\begin{aligned} g_{\text{depth}}(\mathbf{X}) &= \left[\vec{x}_{\text{g},z}\right], && (8.12)\\ G_{\text{depth}} &= \left[0,0,1,\mathbf{0}_{1\times 6}\right]. && (8.13) \end{aligned}$$

Acoustic Range

Let us consider an acoustic range measurement from a beacon at position $\vec{x}_{\text{g,B}}$. Then, we have

$$g_{\text{range}}(\mathbf{X}) = \|\vec{x}_{\text{g}} - \vec{x}_{\text{g,B}}\|. \tag{8.14}$$

The corresponding Jacobian is

$$G_{\text{range}} = \left[\vec{x}_{\text{g}} - \vec{x}_{\text{g,B}}, \mathbf{0}_{1\times 6}\right] \cdot \frac{1}{\|\vec{x}_{\text{g}} - \vec{x}_{\text{g,B}}\|}. \tag{8.15}$$

The sources of error in acoustic range measurements are multi-path reflections, echoes, and signal noise. Some of these sources result in errors that can not be accurately modeled with a Gaussian distribution. Therefore, performing an acoustic range update within an EKF framework is not appropriate on a theoretical level because it inherently assumes a Gaussian error model. Despite that, it works well in practice and offers a number of advantages such as low computational cost and simple implementation, which makes it suitable for a real-time system. In addition, we will show later (in Section 14.3, Fig. 14.5) that after outlier-rejection, the error distribution approximately does follow a Gaussian distribution.

Note that range updates are performed in 2D, given that the depth sensor provides a fairly accurate measurement of the vertical dimension. To do so, the slant range is projected on the horizontal plane knowing the difference in depths of the beacon and the robot. This projected range measurement is used for a 2D position update.

8.1.5 EKF

The dynamics and the measurement models are integrated into an EKF framework as follows.

Process Update

We use the process model $f(.)$ to compute the future state as

$$\hat{\mathbf{X}}^{t+dt} = f\left(\mathbf{X}^{t}, dt\right), \tag{8.16}$$

where t denotes time. The individual state components are computed as

$$\begin{aligned} \hat{x}_{\mathrm{g}}^{t+dt} &= \vec{x}_{\mathrm{g}}^{t} + \vec{v}_{\mathrm{g}}^{t} \cdot dt, & (8.17)\\ \hat{v}_{\mathrm{g}}^{t+dt} &= \vec{v}_{\mathrm{g}}^{t} + \mathbf{q}\left(\vec{a}_{\mathrm{b}}^{t}\right)\mathbf{q}^{-1} \cdot dt, & (8.18)\\ \hat{a}_{\mathrm{b}}^{t+dt} &= (1/m) \cdot \vec{F}_{\mathrm{net,\,b}}, & (8.19) \end{aligned}$$

where $\mathbf{q}$ is the attitude quaternion introduced earlier in Eq. (8.2), used here for the inverse of the operation performed in Eq. (8.6). The acceleration is calculated using the dynamics model in Eq. (8.7).

The new covariance is calculated as

$$\hat{\Sigma}^{t+dt} = F\Sigma^{t}F^{\mathrm{T}} + J_{\theta}\Sigma_{\theta}J_{\theta}^{\mathrm{T}} + \mathscr{R}, \tag{8.20}$$

where the motion noise covariance, $\mathscr{R}$, is constant and determined empirically. The Jacobian of the process model, F is calculated as

$$F = \begin{bmatrix} \mathrm{I}_3 & \mathrm{I}_3 \cdot dt & 0_{3\times3} \\ 0_{3\times3} & \mathrm{I}_3 & R(\mathbf{q})dt \\ 0_{3\times3} & \frac{-2}{m} \cdot \mathrm{diag}\left(\mathrm{sgn}(v_{rx}) \cdot k_x \cdot v_{rx}, \ldots_y, \ldots_z\right) R(\mathbf{q}^{-1}) & 0_{3\times3} \end{bmatrix}, \tag{8.21}$$

where $R(\mathbf{q})$ is the rotation matrix corresponding to quaternion $\mathbf{q}$. Note that $\vec{F}_{\mathrm{net,\,b}}$, used in the calculation of acceleration in Eq. (8.19), depends on both velocity and attitude (since velocity in body frame is obtained using $\vec{v}_{\mathrm{g}}$ and $\mathbf{q}$, see Eqs. (8.5) – (8.7)). This dependency shows up as the non-zero sub-matrices in the third row of the Jacobian matrix in Eq. (8.21).

The middle term in Eq. (8.20) corresponds to propagation in uncertainty in attitude. Σ_{θ} is a diagonal matrix representing uncertainty in attitude along roll, pitch, and yaw axes, obtained from IMU characteristics. J_{θ} is the Jacobian matrix computed as the derivative of the state variable with respect to the three angles, $\vec{\theta} \triangleq [\theta_{\mathrm{roll}}, \theta_{\mathrm{pitch}}, \theta_{\mathrm{yaw}}]$.

$$J_{\theta} = \frac{\partial \hat{\mathbf{X}}^{t+dt}}{\partial \mathbf{q}} \frac{\partial \mathbf{q}}{\partial \vec{\theta}}. \tag{8.22}$$

Measurements

Measurement updates are performed individually for IMU, GNSS or acoustic information, using the measurement models derived earlier and standard EKF update equations.

$$\begin{aligned} K_i &= \hat{\Sigma}G^{\mathrm{T}}\left[G\hat{\Sigma}G^{\mathrm{T}} + Q_i\right]^{-1}, & (8.23)\\ \mathbf{X} &= \hat{\mathbf{X}} + K_i\left(z_i - g(\hat{\mathbf{X}})\right), & (8.24)\\ \Sigma &= (\mathrm{I}_n - K_i G)\,\hat{\Sigma}, & (8.25) \end{aligned}$$

where z_i is the ith measurement, K_i is called the Kalman gain computed during this measurement, $g(.)$ is the measurement model of a particular sensor, G is the corresponding Jacobian, Q_i is the covariance representing the measurement noise of the sensor and I_n is the identity matrix of appropriate dimension.

Parameters

Various parameters used in the dynamics model in Section 8.1.3 were initially either calculated mathematically using certain assumptions or tuned by hand. A formal approach to tuning these parameters is discussed in Chapter 12.

Initialization

The initial position is obtained from GNSS before launching the robots. The initial velocity and acceleration can be assumed to be zero. The initial heading is either obtained from the magnetometer, or estimated separately when the magnetometer is not used. The gyro biases are estimated by leaving the robot stationary for a few seconds and calculating the average of the gyro measurements. This is taken care of automatically by the Xsens IMU.

8.2 Navigation

Autonomous missions consist of a list of actions to be performed by the robot. The list of actions is uploaded to the robot before a mission using the planning interface in the base station software. The actions may directly specify a waypoint to visit or execute an adaptive sampling algorithm to calculate the next sampling point. In either case, a goal position $\vec{W}_\mathrm{g}$ is provided (in ground frame). The desired cruise maximum speed s_max and an acceptance radius are also set using the base station software. The robot controllers then navigate the robot towards the goal point, within a circle of the specified acceptance radius. The minimum value of the acceptance radius is set to 3.0 m.

The relative position to the goal is calculated as

$$\vec{r}_\mathrm{g} = \vec{W}_\mathrm{g} - \vec{x}_\mathrm{g}, \tag{8.26}$$

where $\vec{x}_\mathrm{g}$ is the position of the robot estimated by the EKF. The relative position in the body frame is calculated as

$$\vec{r}_\mathrm{b} = \mathbf{q}^{-1}\left(\vec{r}_\mathrm{g}\right)\mathbf{q}. \tag{8.27}$$

The cruise speed for navigating to the goal is set as

$$s = \min\left(s_\mathrm{max}, K_s \|\vec{r}_\mathrm{g}\|\right), \tag{8.28}$$

where the constant K_s is chosen such that the desired speed gradually drops to zero as the robot nears the goal.

8.3 Control

The relative goal positions in body and ground frame, $\vec{r}_{\mathrm{b}}, \vec{r}_{\mathrm{g}}$ serve as the primary input to the controllers.

Attitude Control

The desired attitude is such that the robot points towards the goal. The set points for roll, pitch, and yaw angles is calculated as

$$\begin{aligned}
\theta_r &= \tan^{-1}\left(\frac{\vec{r}_{\mathrm{b},y}}{\|\vec{r}_{\mathrm{b}}\|}\right), && (8.29)\\
\theta_p &= \phi_{\mathrm{d}} - \tan^{-1}\left(\frac{\vec{r}_{\mathrm{g},z}}{\|\left[\vec{r}_{\mathrm{g},x}, \vec{r}_{\mathrm{g},y}\right]\|}\right), && (8.30)\\
\theta_y &= \tan^{-1}\left(\frac{\vec{r}_{\mathrm{g},y}}{\vec{r}_{\mathrm{g},x}}\right), && (8.31)
\end{aligned}$$

where $\theta_r, \theta_p, \theta_y$ are the roll, pitch and yaw angles, and $\vec{r}_{\mathrm{g},x}$ is the x coordinate of the vector $\vec{r}$ in the specified frame. ϕ_{d} is the pitch angle adjustment output of the depth controller, explained later in this section. Note that the body frame relative vector is used in the calculation of roll angle. These set points are fed to a PID controller. The feedback to the controller is provided by the estimated attitude and the rate gyroscope measurement.

When there is no goal position, the input to the controller is zero. The controller output in this case actively maintains roll and pitch angle at zero (i.e., keeps the AUV upright). On the ASV, only the yaw angle is controlled.

Depth Control

The AUV is slightly more buoyant than its weight, which implies that there is a constant net force that tends to displace the body in the upward direction. Therefore, in order to follow a planned trajectory or to hold a fixed position, the depth must be actively regulated (see Fig. 8.2). The required depth, d_{cmd} is computed by the navigation module. For hovering at a certain position, d_{cmd} is the depth of that position. When navigating between two waypoints, from $\vec{W}_1$ to $\vec{W}_2$, d_{cmd} is computed in a way that the trajectory between the waypoints is a straight line. That is, d_{cmd} varies linearly between the z coordinates of the waypoints, W_{1z} and W_{2z}, with the distance covered by the AUV in the horizontal plane. If the current position of the AUV is $\vec{x}$,

$$d_{\mathrm{cmd}} = W_{1z} + (W_{2z} - W_{1z})\frac{\mathrm{D}_{xy}(\vec{x}, \vec{W}_1)}{\mathrm{D}_{xy}(\vec{W}_2, \vec{W}_1)}, \qquad (8.32)$$

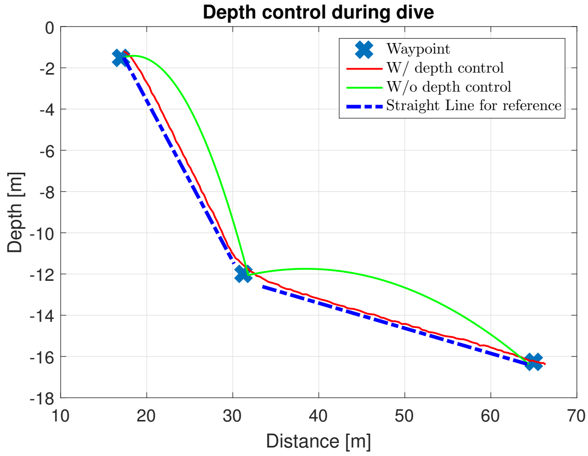


Figure 8.2: AUV trajectory with and without depth control. Without active depth control, the AUV can reach the programmed waypoints, but the trajectory will not be a straight line. This plot is from a simulation.

where $\mathrm{D}_{xy}(\vec{v}_1, \vec{v}_2)$ is the distance between vectors $\vec{v}_1$ and $\vec{v}_2$ projected on the horizontal plane. Then, the depth error is

$$d_{err} = d_{\text{cmd}} - d_{\text{actual}}. \tag{8.33}$$

Note that all vectors here are in the ground frame. When stationary or at low speeds, the vertical thrusters are used to regulate depth. This consumes a relatively large amount of energy. When moving forwards with a speed higher than a threshold, it is more efficient to adjust the pitch angle downward to regulate depth, using the AUV hull and the winglets to generate a vertical force. We have a smooth transition between the two controllers around the threshold speed s_0, which was set to 0.5 m/s, half of the maximum allowed cruise speed. The inputs to the PID controller for pitch, ϕ_{d}, and heave command to the vertical thrusters, H_{d} are computed as

$$\phi_{\text{d}} = f_{mix} \cdot K_{\phi d} \cdot d_{err}, \tag{8.34}$$

$$H_{\text{d}} = (1 - f_{mix}) \cdot K_{Hd} \cdot d_{err}, \tag{8.35}$$

where $K_{\phi d}$ ans K_{Hd} are constant parameters and the mixing factor f_{mix} is computed as a function of the AUV estimated velocity $\vec{v}_{\text{g}}$ and threshold speed s_o as

$$f_{mix} = \frac{1}{2} + \frac{1}{2} \cdot \frac{K_{mix} \cdot (\|\vec{v}_{\text{g}}\| - s_o)}{\sqrt{(1 + K_{mix}^2 (\|\vec{v}_{\text{g}}\| - s_o)^2)}}. \tag{8.36}$$

$f_{mix} \in [0, 1]$ and $f_{mix} = 0.5$ when the AUV speed $\|\vec{v}\|$ is equal to s_o. The parameter K_{mix} controls the smoothness of the controller transition.

Thrust and Heave Control

We use a simple P controller for thrust (forward thrusters, for both AUV and ASV) and heave (vertical thrusters, only on AUV). The thrust set point, T is calculated as

$$T = \begin{cases} K_T \cdot \vec{r}_{\mathrm{b},x} \cdot \left(s - \|\vec{v}_{\mathrm{g}}\|\right), & \vec{r}_{\mathrm{b},x} \geq 0 \text{ and } \|\vec{r}_{\mathrm{b}}\| \geq 4.0 \\ 0, & \text{otherwise} \end{cases}, \tag{8.37}$$

where s is the desired cruise speed calculated earlier in Eq. (8.28). $\vec{r}_{\mathrm{b}}$ and $\vec{r}_{\mathrm{b},x}$ were introduced in Eqs. (8.29)–(8.31).

The heave set point, H is only set by the heave output of the depth controller.

$$H = K_H \cdot H_{\mathrm{d}}. \tag{8.38}$$

In the two equations above, K_T and K_H are constant parameters for the P controller.

8.4 Conclusion

This chapter provides a mathematical description of the estimation and control modules of the robots used in this thesis. In the next chapter, an adaptive sampling mission is described, where the AUV operated using the modules presented in this chapter. The trajectories of the robots estimated with this estimation framework are presented along with relevant results in the subsequent chapters. Subsequently, we developed an acoustic ranging system to provide external position references for correcting the drift in the inertial trajectory estimate. Finally, in Chapter 12, the last chapter in this part, we refer back to the state estimation framework presented here and show how the values of fixed parameters used here in Eq. (8.7) are obtained from operational data of the robots.

9 Adaptive Sampling Missions

ROBOTIC sampling in underwater environments is the primary target application of the work done in this thesis. Robots can do more than simply transporting a sensing probe to different locations for gathering spatial measurements. By performing in-situ measurements and processing them on-board, robots can adapt their trajectory to track and trace features, or target specific regions of interest for gathering data, a paradigm referred to as Adaptive Sampling.

The previous chapter introduced the basic subsystems of a single AUV that enabled autonomous operation. In this chapter, we describe an environmental sampling mission where the AUV autonomously locates a zone of interest and adapts its trajectory to stay within it. Further, it executes specific behaviors to accommodate special sensing requirements necessary to enhance the quality of the data collected. Data gathered from field deployments are presented. The problems and challenges identified during this deployment shaped the work in the subsequent chapters of the thesis.

The work described in this chapter was done in collaboration with limnologists (see Section 3.3). We attempted to map physical, chemical and biological quantities in lakes at high spatial resolutions (of the order of 1 cm vertically and 10 cm horizontally). We performed robotic data gathering missions in several lakes in Switzerland during the summer in 2016–2018. The objective was to understand the interplay between small-scale physical processes and biological activity in water bodies.

9.1 Problem Definition

The primary goal of the work in this chapter is to gather data from within a particular zone of interest in Lake Cadagno, Switzerland. The lake exhibits a distinct, seasonal phenomenon where certain bacterial activity is concentrated in a thin stratified layer. Our goal was to track the boundary of this layer and adaptively adjust the trajectory of the AUV to capture spatial data within it.

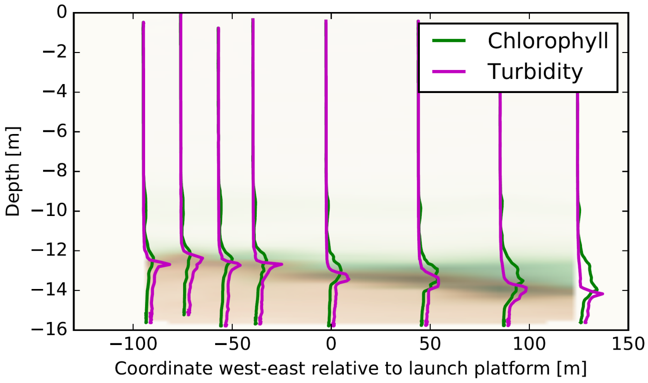


Figure 9.1: A diagram of eight vertical profiles along an east-west transect during an afternoon in September 2016, showing the stratified bacterial layer (dark brown) below a layer rich in chlorophyll (green) in Lake Cadagno. The layer has varying thickness and depth and is characterized by a well-mixed thermocline and high turbidity.

9.1.1 Bacterial Layer

Lake Cadagno is a meromictic lake with a stratified thermocline layer at a depth of about 15 m, which contains diverse species of anaerobic sulfur-reducing photosynthetic bacteria [71]. The layer has a thickness of 0.25-2 m, and exhibits a sharp increase in chlorophyll concentration and turbidity within it, as shown in Fig. 9.1. These bacteria can accumulate and increase the density of the local volume of fluid, which sinks and mixes with the surrounding water, while upward swimming bacteria help maintain water motion. This phenomenon, called bio-convective mixing, generates small-scale turbulent eddies within the stratified bacterial layer [72]. These eddies can be characterized by observing the spatial spectra of temperature measurements. It is important to understand the activity of these sulfur bacteria and mixing processes in order to study their effect on the lake ecosystem. Sepúlveda et. al. in [13] performed an intensive 48 hour survey using a combination of vertical profilers and moored sensors to characterize the behavior of the bacteria.

Such stratified phenomena occur in other lakes as well. A relevant example is that of Lake Zurich, where the cyanobacteria *Planktothrix rubescens* forms a high-concentration layer [73], and also exhibits similar turbulent mixing [74]. We have also performed measurement missions in this lake.

9.1.2 Environmental Sensing

The AUV was equipped with turbidity, chlorophyll, and CTD (conductivity-temperature-depth) probes, which are a part of the YSI EXO2 system. In addition, a fast temperature sensor module (developed in our laboratory by Dr. Alexander Bahr) was added, which can measure at a sampling rate of up to 400 Hz. It is enclosed in a package compact enough to be integrated into the AUV (see Fig. 9.2). While not calibrated to be accurate, this sensor has a resolution of 20 μK, which makes it suitable for capturing microstructure- and turbulence-

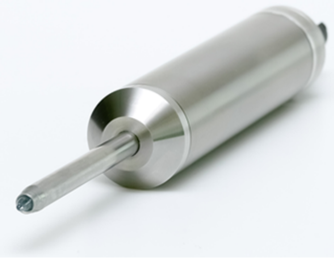


Figure 9.2: The fast temperature sensing module. It is 20 cm long and 2.5 cm in diameter. It has a sampling frequency of up to 400 Hz, and a resolution of 20 μK.

induced temperature fluctuations. The YSI temperature sensor (on the CTD probe) is used as a reference. It has lower resolution and sampling rate, but accuracy of 10 mK. The fast temperature module also includes a 9-axis inertial measurement unit for measuring the orientation and vibrations of the temperature sensor. This module was compared to existing fast temperature sensors from Rockland Scientific [75] with measurements in a flow channel and in an outdoor environment, in collaboration with limnology researchers, and was found to have similar performance.

9.1.3 Sampling Constraints

The length scales of the turbulent eddies within the bacterial layer are similar to the thickness of the layer, i.e., they can be as small as 0.25 m. The fast temperature sensor on the AUV measures at 400 Hz, and at AUV cruise speed of approximately 1 m/s, the temperature measurements would be taken every 2.5 mm, which is adequate for capturing the turbulent eddies (which are an order of magnitude smaller). Given that, the AUV can greatly complement traditional underwater sampling methods by gathering data at high lateral resolutions. Nonetheless, there are a number of practical challenges in sampling with an AUV.

The temperature fluctuations due to the small-scale turbulence within the bacterial layer are extremely weak, in the sub-millikelvin range. The fast temperature sensor on the AUV is capable of resolving those fluctuations. However, vibrations induced by the AUV propellers can also cause interference in the temperature measurements due to the extremely high sensitivity of the sensor, which can potentially corrupt the turbulence measurements. To mitigate this effect, we implemented a behavior that turns off all propellers for a short duration at a suitable time and lets the AUV drift forward due to its inertia to capture spatial temperature measurements without motor vibrations. Further, the thickness and depth of the bacterial layer vary over the lake and with time of the day. Data from traditional sensing methods can provide localized information about the layer depth, but this was found to be inadequate to guarantee that preplanned, horizontal AUV trajectories would always trace the layer. Therefore, we process sensor measurements in real-time on-board the AUV to adaptively adjust the trajectory and trace the layer.

9.2 Adaptive Behaviors

All measurement data are sent from the sensing module to other AUV modules and logged onto the memory card. This allows us to process the data on-board and use them for real-time control. Furthermore, this also allows us to post-process sensor data in conjunction with AUV operational data to evaluate and implement adaptive behaviors.

In order to sample within the bacterial layer and capture the turbulent eddies caused by bacterial activity, the AUV needs to

- identify, locate and trace the bacterial layer, and
- record fast temperature measurements without motor vibrations.

The following sections describe how these two objectives are achieved.

9.2.1 Identifying and Tracing the Layer

We observed a sharp increase in turbidity and chlorophyll at a depth of 12-14 m in the vertical profiles shown in Fig. 9.1. Measurements carried out using fully planned AUV missions with 'yo-yo' trajectories showed a similar phenomenon, as shown later in Fig. 9.4. This observation is consistent with the results reported in [71] stating that the bacterial layer is characterized by higher concentrations of chlorophyll and a turbidity maximum.

Using the data from the fully planned missions, we set a simple threshold on turbidity measurements to identify the bacterial layer. Using the estimated position from the state estimator, the local upper and lower layer boundaries at the current position would be updated. Then, by regulating its depth in real-time, the AUV would attempt to stay within the layer for the rest of the mission. To achieve this in practice, we implemented a new type of waypoint, called *adaptive waypoint* where the horizontal positions are pre-specified by the user but the depth is set automatically during the mission based on turbidity measurements. The turbidity was monitored continuously to update the layer boundary estimates in case the AUV left the layer. Data gathered in such an adaptive mission are shown in Fig. 9.6a.

9.2.2 Mitigating Motor-Induced Vibrations

Tracing the layer with adaptive sampling is sufficient for gathering spatial variation in all quantities except fast temperature. Since temperature fluctuations due to the small-scale turbulence are very weak, the temperature measurements can be easily contaminated due to the vibrations induced by the AUV propellers. To mitigate this effect, we implemented another new type of waypoint, which we call *burst waypoint*. The AUV aimed for this waypoint at the maximum possible cruise speed, and on reaching it, it turned off all motors for a specified duration of time, usually 20 s. It drifted forward due to inertia and continued to sample data without the motor vibrations. The next step was to combine the adaptive and burst features

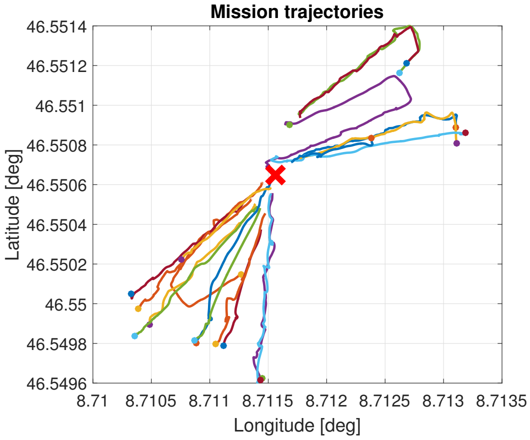


Figure 9.3: AUV trajectories at Lake Cadagno from 19 separate missions, projected on the horizontal plane. The red cross marks the launch point of the missions.

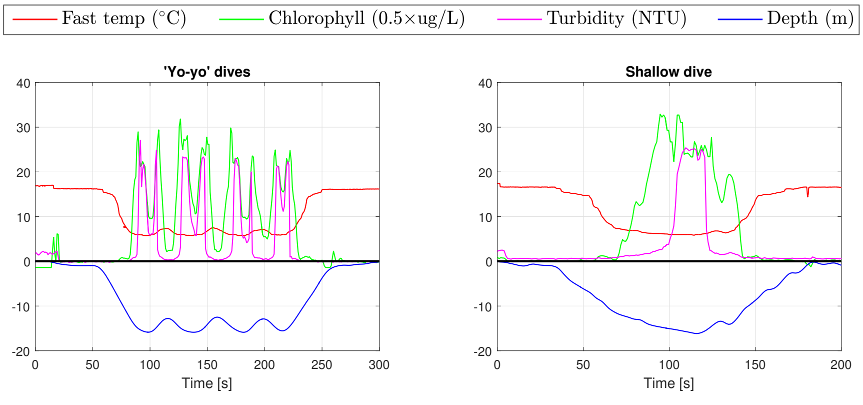
to sample data within the layer with motors off. This was achieved by setting the depth of the burst waypoint automatically by monitoring the turbidity measurements in real-time, as explained earlier. This allowed the AUV to drift horizontally within the layer with its motors off, thus capturing the lateral variation in temperature required for studying the small-scale turbulent eddies. The data gathered from such 'burst mode' missions are shown in Fig. 9.6b.

9.3 Results from Field Deployments

Several AUV missions were performed over three days in Lake Cadagno in August 2018. Data were gathered along several trajectories, which were planned in collaboration with limnologists. Fig. 9.3 shows a plot of the mission trajectories. The horizontal distance to be covered in each of those missions was about 100 m, and the maximum cruise speed of the AUV was set to 1 m/s. The initial missions were fully pre-planned, and the subsequent ones had an adaptive component to autonomously detect and trace the bacterial layer. During all the missions, the AUV was operating completely autonomously.

9.3.1 Fully Planned Missions

To start with, we performed several pre-planned 'yo-yo' trajectories between a specified depth envelope as shown in Fig. 9.4. The depth parameters were chosen to be conservative enough to capture the bacterial layer during the dives. The sensor data from one of these missions are shown in Fig. 9.4a. The twin peaks in turbidity and chlorophyll during each yo-yo dive segment show that the AUV crossed through the bacterial layer in the downward and upward directions of the dive. However, in these missions, the lateral coverage within the layer is limited. We performed several such trajectories to capture spatial data over an area of the lake. We then planned a shallow dive mission shown in Fig. 9.4b, where the AUV descends vertically over a long horizontal distance. This was done in an attempt to traverse through the layer over



(a)

(b)

Figure 9.4: Pre-programmed missions to measure through the bacterial layer. Plots (a) and (b) show the depth of the AUV over time and the corresponding sensor measurements. (a) shows a yo-yo trajectory mission between a depth envelope of 12 m and 16 m. The twin peaks in chlorophyll and turbidity show the crossing of the AUV through the bacterial layer. (b) shows a shallow dive from 14 m to 16 m over a horizontal distance of 15 m. Only a short segment of the trajectory was within the bacterial layer.

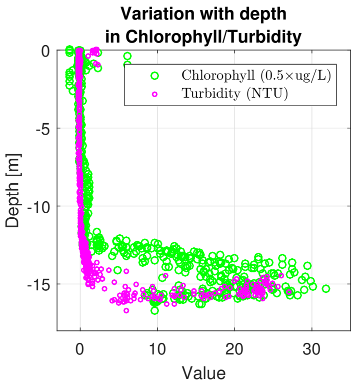


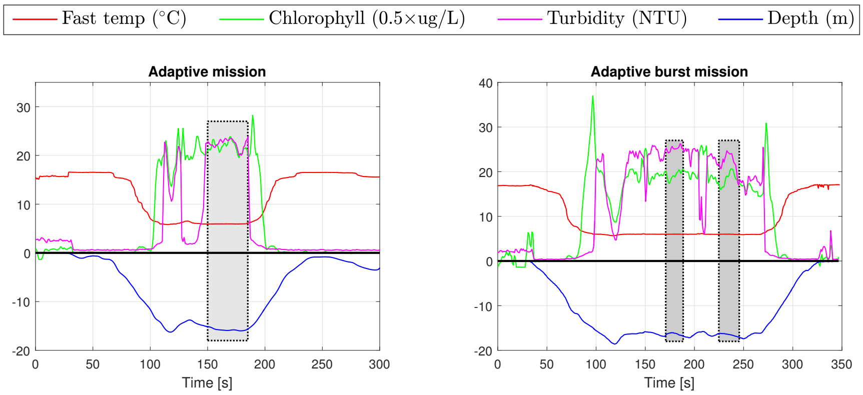
Figure 9.5: Variation in chlorophyll and turbidity with depth during a fully planned mission. There is a sharp rise in these quantities within the bacterial layer. These measurements are used to set the threshold for detecting the bacterial layer.

a longer horizontal distance. Yet, given that the layer thickness and depth is not constant, in many of the missions, only a short segment of the trajectory actually passed through the layer.

Fig. 9.5 shows a plot of chlorophyll concentration and turbidity against depth over one of the yo-yo trajectories. For the adaptive missions, we set the threshold for detecting the layer to half of this measured maximum turbidity value.

9.3.2 Adaptive Missions

During the adaptive missions, the first two waypoints were specified with depth parameters conservative enough to encompass the layer, which allowed the AUV to detect the local layer boundaries. We then used the adaptive waypoint to specify the desired horizontal position.



(a) Adaptive mission.

(b) Adaptive burst mission.

Figure 9.6: (a) Sensor data from an adaptive mission. The shaded region marks the segment where the trajectory is adaptively adjusted. A high value of turbidity shows that the robot traversed within the layer. (b) Sensor data from a mission with burst mode waypoints. The shaded regions indicate the burst segments which lasted for 20 s, during which the robot drifted forward within the layer due to inertia.

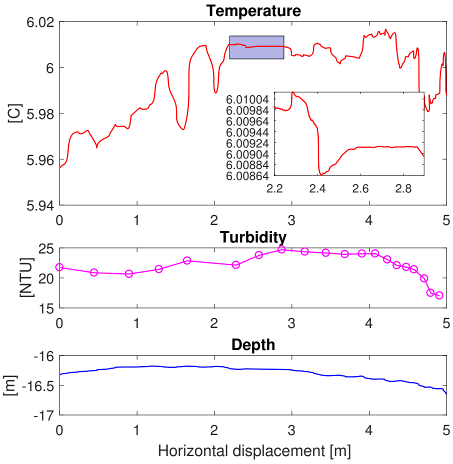


Figure 9.7: Sensor data from a burst segment. A zoomed-in view of the shaded section of the temperature plot is shown.

The AUV adapted the depth automatically to stay within the layer. Data gathered during this mission are shown in Fig. 9.6a. Leaving the layer would have resulted in a change in turbidity measurements, triggering an update in the estimated local layer boundary, and an adjustment in the AUV depth. However, the turbidity measurements in the gathered data show that the AUV was within the bacterial layer throughout the mission.

9.3.3 Burst Waypoint Missions

We used the burst waypoint to sample data within the layer with motors off. Similar to the previous mission, the AUV detected the layer boundaries and adjusted its depth to stay within the layer. In the data from an adaptive burst mission shown in Fig. 9.6b, there are two burst segments, each with a duration of 20 s, when the AUV was drifting with motors turned off. An expanded view of one of the burst segments is shown in Fig. 9.7. The horizontal displacement of the AUV (estimated by the on-board EKF) during this time is about 6 m. A high value of turbidity indicates that the robot stayed within and traced the bacterial layer. A zoomed-in view of the temperature plot shows the minute fluctuations in temperature. The bio-induced eddies can be characterized by studying the spatial spectra of this temperature signal.

9.4 Conclusion

We implemented autonomous feature tracing and adaptive sampling behaviors for gathering data in real-world environments and demonstrated the capability of the Vertex AUV to execute them. We collaborated with limnologists at the Physics of Aquatic Systems Laboratory at EPFL to gather data from Lake Cadagno in Switzerland. The AUV could successfully trace the compact bacterial layer in the lake to sample various quantities at a high lateral resolution. Further, in a separate mission, the AUV could turn off the motors during the mission within the layer and drift forward due to inertia. This was done to avoid motor-induced vibrations from corrupting the fast temperature sensor measurements, thereby enhancing the quality of temperature data. High lateral resolution data gathered within the bacterial layer would complement measurements taken using standard limnological methods leveraging vertical profilers. This would allow studying various biological and physical processes in an additional dimension. In summary, the main contributions of the work in this chapter are (i) using the on-board state estimation and control system presented in the previous chapter for executing autonomous robotic sensing missions, (ii) processing sensor data on the AUV in real-time to locate and stay within a zone of interest, and (iii) implementing and executing specific behaviors that enhance the quality of the data gathered.

During this work, the AUV had no external position reference (except for the depth sensor) for underwater localization and essentially performed inertial navigation. The use of GNSS to obtain position while the robot is on the surface was hindered for two reasons. Firstly, the lake was surrounded by mountains, which occluded a large part of the sky. Secondly, even when the robot is near the surface, there is a thin film of water just above the GNSS receiver antenna installed inside the robot, which distorted the received signal and severely hampered position accuracy. Given the small size, installing the GNSS receiver outside the robot hull was not feasible.

A system that continuously provides external position references is absolutely essential for accurate position estimation. In the subsequent chapters, we develop an acoustic ranging and communication system to aid underwater localization. We then also attempt to tune the parameters of the model used for state estimation, in order to improve inertial positioning accuracy. Further in the thesis, we develop cooperative methods for a team of robots to improve their individual localization accuracy.

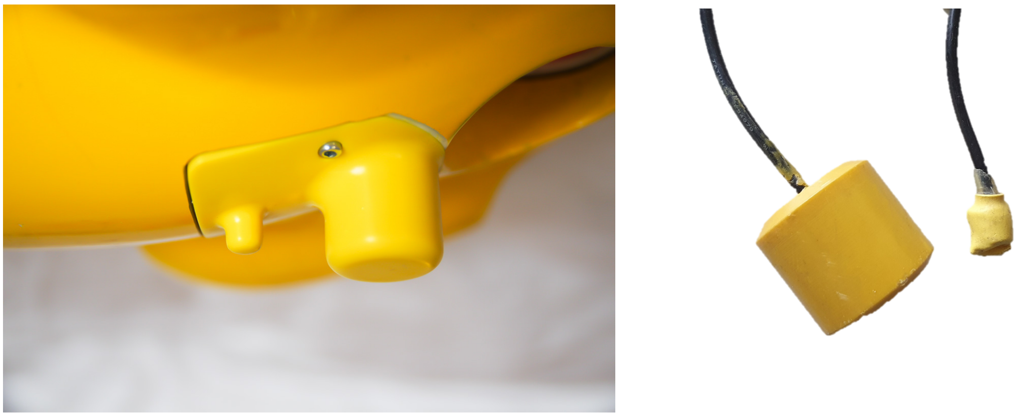
10 Acoustic Transceivers

Electromagnetic waves quickly attenuate in water, which renders satellite-based positioning and radio communication unusable for AUVs. Further, vision-based methods are not reliable due to turbidity or lack of proximity to physical features in natural water bodies. Highly accurate inertial sensors may offer acceptable positioning accuracy, but are expensive and often not compatible with small-sized underwater robots. A common approach to underwater localization and communication, therefore, is to employ acoustic signals [50]. The time of arrival of an acoustic signal is then used to compute the range (based on time-of-flight) to a transmitting beacon, whose position must be known. A series of such geometric measurements are used to deduce the AUV position.

The system developed in our work uses absolute time information from GNSS for initial clock synchronization, and measures one-way-travel-time to obtain range measurements. The beacons are the only transmitters of acoustic signals, and robots are passive receivers, which makes the system scalable in the number of robots. It is easily deployable and does not rely on any installed infrastructure in the environment. This chapter introduces the acoustic transceiver hardware and software. Note that the hardware was acquired from Hydromea, while the software and integration with the robots was done as a part of this thesis (in addition to minor hardware modifications). Results from experiments with ASVs in a real-world environment are presented at the end.

10.1 Acoustic Transceiver Module

The acoustic transducer module consists of electronics mounted on a PCB and two separate piezoelectric transducers, one for transmitting acoustic signals and another for receiving them. To recap, on the ASV, the acoustic module PCB is installed in the box along with other electronics as shown in Fig. 5.4. The two transducers are connected with two separate coaxial cables and are suspended in water. A ground station acoustic transceiver has a similar setup, without the ASV main or computation modules. On the AUV, the PCB and both the transducers are packed into a compact, waterproof package and mounted at the rear-bottom of the AUV.



(a) AUV acoustic system

(b) Transducers used on ASVs

Figure 10.1: (a) Acoustic transceiver module used on the AUV. (b) Transmitter and receiver transducers used on the ASV and the ground station. They are connected to the PCB with two separate cables. In both (a) and (b), the larger transducer is the transmitter ans the smaller one is the receiver.

The acoustic module on both ASV and AUV is shown in Fig. 10.1.

Transducers

The module contains two piezoelectric transducers. The transmitter is a cylindrical piezoelectric transducer with a diameter of 3 cm and a length of 2.5 cm with resonance at a frequency between 39-49 kHz. The receiver is a smaller transducer of diameter 1 cm, and an almost uniform frequency response. Both transducers are omnidirectional perpendicular to the axis of the transducer.

Electronics

The module contains an AVR-32 series microcontroller with hardware peripherals for Digital Signal Processing (DSP) and is programmed to generate and process acoustic signals. The two transducers are connected to the Analog-to-Digital and Digital-to-Analog Converters (ADC / DAC) of the microcontroller via appropriate amplification stages. Note that the electrical signal generated by the receiving transducer is very weak (~ 1 mV), and hence, is prone to external electromagnetic interference. The acoustic transceiver module connects to the rest of the robot via a UART interface.

Signal Processing

For signal transmission, the DAC supports a maximum sample frequency (of the input digital signal) of 1 MHz. However, given the limited on-board memory of the microcontroller (which must be shared with other tasks), there is a tradeoff between signal length and sample frequency. We used digital signals with sample frequency of 250 kHz. For received signals, the

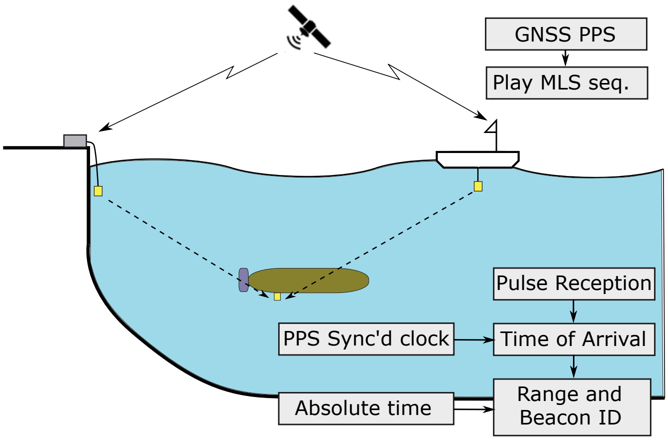


Figure 10.2: One beacon is deployed from the shore and another mounted on an ASV. The two surface beacons transmit alternatively following a fixed schedule based on absolute time, and use the GNSS PPS to trigger playback of an acoustic pulse. The AUV uses absolute time to identify the current transmitter and computes time of flight.

ADC on the microcontroller is capable of sampling at up to 160 kHz, but we used a sampling frequency of 100 kHz. Here also there is the tradeoff between the sampling rate and the length (in time) of the received signal that can be stored in memory for further processing. The AVR-32 microcontroller provides a set of routines for performing common signal processing functions (such as FFT, convolution, etc.) using the dedicated hardware DSP unit.

10.2 Range-Based Localization

The localization system consists of two or more beacons deployed at known, static positions. They can either be deployed from the shore or mounted on an ASV programmed to hold a specific position, as depicted in Fig. 10.2. Note that anchored buoys are not perfectly static and can sway. The beacon positions are transmitted to the AUVs before they are submerged in water. The AUVs then obtain range measurements to the beacons for position estimation. The robots and the beacons use the same acoustic transceiver modules. The difference is that the beacons only use the transmitting transducer, and the robots use only the receiver.

Given that the AUV is equipped with a depth sensor, position estimation reduces to a two-dimensional problem. Further, since the initial position is known and incremental inertial updates are available, two beacons are sufficient to fully estimate the AUV position. We can tune the accuracy of the system by altering the relative location of the two (or more) beacons.

10.2.1 Clock and Time Synchronization

For computing range from one-way travel time and for scheduling transmissions from multiple beacons, we need all the beacons and the AUV to have synchronized clocks. We do that at two levels. First, we use the timing pulse emitted per second (PPS) by the Ublox M8N GNSS receiver module, which has an accuracy of the order of 10 ns, to tune the *speed* of the clock

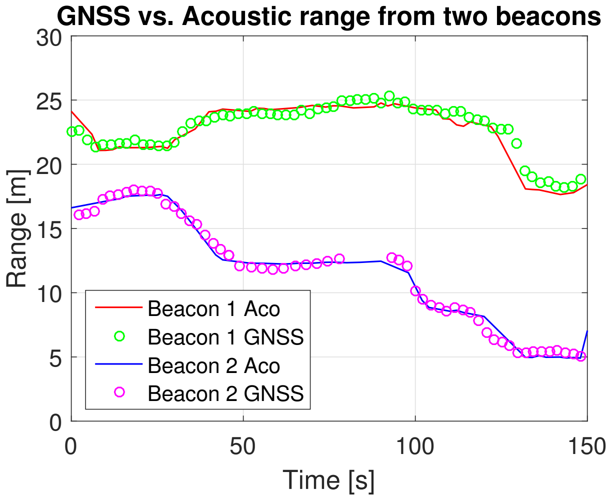


Figure 10.3: Range measured using one-way travel time by a receiver to two beacons that transmit according to a prearranged schedule. Range computed by GNSS positions is shown for comparison. The receiver associates each range measurement to the correct transmitter beacon using absolute time information from GNSS.

and eliminate the clock drift. This helps in accurate measurement of time of arrival of the ranging pulse, as explained in the next section. Second, we use the absolute time information received from the GNSS module to set the wall-clock time. The absolute time solution has an error of the order of 10 ms. To mitigate that, we first set the wall-clock time to the GNSS time solution, and then round it off to the next full second at the PPS trigger. With a known transmission schedule and synchronized clocks, each device knows the identity of the current and previous transmitter.

The AUV clock is synchronized while it is on the surface, but the clock drifts when underwater. For this reason, the AUV is equipped with a crystal oscillator of higher accuracy and temperature compensation. It has a clock drift of 0.3 ppm, which translates to a range measurement error of less than 1 m over 1 hour. This error is insignificant compared to the accumulated error in purely inertial positioning. Nonetheless, this clock drift can be reset with GNSS reception during the periodic resurfacing of the AUV.

10.2.2 Transmission Scheduling and One-Way Travel Time Ranging

Our system is based on one-way travel time for ranging, where only the surface beacons transmit and the AUVs are passive receivers. This has the advantage of scalability since any number of AUVs can receive the ranging pulse transmitted by a beacon. We use the PPS signal, which is fed to an interrupt pin on the microcontroller, to trigger playback of a pre-recorded Maximum-Length Sequence (MLS, a pseudorandom binary sequence) pulse through the transmitting transducer.

On the receiving side, a threshold on the received signal amplitude is used to detect a ranging pulse. The receiver then measures the time between its own PPS trigger and the acoustic pulse

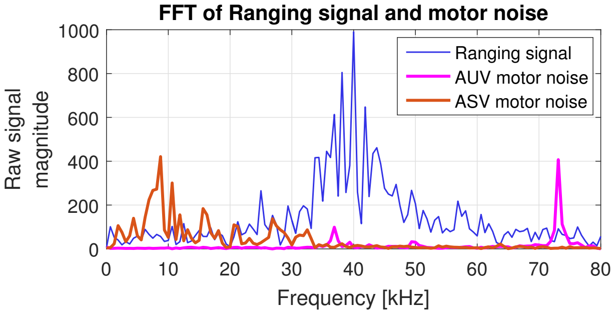


Figure 10.4: FFT of the ranging signal and the motor-induced noise of AUV and ASVs. The ranging signal is a pre-recorded MLS sequence that is sent through a piezo, which has resonance in the 35-45 kHz range.

reception to compute the time of flight.

Only one of the surface beacons can transmit at a time in order to avoid interference. Therefore, we need to schedule their transmissions. We use the absolute time information from GNSS to specify a transmission schedule, which is known to all devices prior to a mission. Each transmitter takes turns to transmit a signal such that there is one transmission overall per second. With two beacons, each beacon gets a transmission slot once in two seconds, during which they use the PPS signal to trigger their transmission. This approach is known as Time Division Multiple Access (TDMA). Fig. 10.3 shows that the receiver is able to correctly identify the transmitting beacon for each range measurement.

10.2.3 Position Updates with Acoustic Range

On receiving a range measurement r, the posterior position and covariance is calculated using the regular Kalman update equations, as introduced in Chapter 8, Section 8.1.4. We recall that following from Eqs. (8.14) – (8.15), the Jacobian of the corresponding measurement model G_{range} is used to compute the Kalman gain, K_{gain} as

$$K_{\text{gain}} = \hat{\Sigma} G_{\text{range}}^{\text{T}} \left[G_{\text{range}} \hat{\Sigma} G_{\text{range}}^{\text{T}} + \sigma_r \right]^{-1}. \tag{10.1}$$

We point out that the bracketed expression is a scalar and hence there is no matrix inversion involved, which makes this update computationally inexpensive.

10.2.4 Signal Filtering and Outlier Rejection

The mechanical noise of the motors is also picked up by the acoustic receiver, which generates spurious range measurements. A Fourier transform of the motor noise shows the presence of frequencies beyond the 70 kHz range for the AUV, and sub 30 kHz range for ASVs (see Fig. 10.4). While we use an MLS signal as the ranging pulse which contains a wide range of frequencies, the transmitters have a resonance in the 35-45 kHz range. Therefore, the ranging pulse is band-limited. A band-pass filter with lower and upper cut-off frequencies of 30 kHz

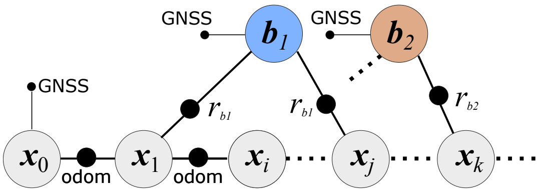


Figure 10.5: Factor graph for smoothing the trajectory estimated by EKF, which fuses odometry and acoustic range measurements. x_n are vehicle positions, connected in a chain by odometry factors. b_1 and b_2 are beacon positions, which are connected to vehicle positions by range measurement factors.

and 50 kHz respectively effectively removes the motor-induced disturbance.

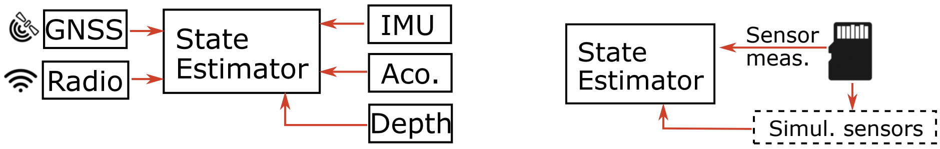
Echoes and reflected signal arrivals result in outlier range measurements. These errors are exacerbated especially when operating in shallow water bodies or close to the shore, as is the case for the experiments presented in this chapter. We perform outlier rejection using the kinematic model of the AUV and the surface beacons, combined with the past range measurements. Specifically, given that the velocities of the AUV and the beacons are bounded, inconsistent range measurements with a large error can easily be identified as outliers and rejected. However, occasional outliers with small errors may be difficult or even impossible to eliminate.

10.2.5 Post-Processing

While the EKF is computationally efficient for on-board fusion of acoustic range measurements, each update is performed by marginalizing out the previous state estimates and measurements. This sometimes results in discontinuities in the estimated trajectory when the inertial estimation error is large or when the range measurements are erroneous. However, in post-processing, we can use the odometry information (control inputs and inertial measurements) logged by the vehicle together with the set of all range measurements to optimize over the whole trajectory. To do so, we represent the vehicle positions, odometry updates, and range measurements as a factor graph, as shown schematically in Fig. 10.5. In the factor graph formulation, consecutive vehicle positions are related through *odometry* factors, while external measurements are represented by their respective *measurement* factors. Essentially, a factor graph represents the joint distribution $P(\vec{x}_1, \vec{x}_2, \dots \vec{x}_n | z_1, z_2, \dots)$, where $\vec{x}_i$ are vehicle positions and z_i are any measurements. Our goal is to estimate the collection of vehicle positions over the entire trajectory. We assume that the first position of the vehicle is known from GNSS reception on the surface, and hence is added as a prior constraint. We then perform a Maximum A-Posteriori (MAP) inference over the factor graph using the GTSAM library [76].

10.3 Data-in-the-Loop Emulation

In Data-in-the-Loop (DIL) emulation, we combine various simulated phenomena with data recorded from outdoor field experiments and use that to evaluate estimation algorithms.



(a) Real-world execution.

(b) DIL emulation.

Figure 10.6: In Data-in-the-Loop emulation, the state estimator is executed using recorded and simulated sensor data instead of data from actual sensing hardware.

Ground truth is provided by recorded GNSS positions. This is done by executing the estimation code offline, and replacing the hardware abstraction layer with appropriate routines that feed in data from a recorded log file instead of the sensing hardware. This is shown schematically in Fig. 10.6. Essentially, this setup allows us to evaluate the performance of estimation algorithms under a variety of simulated scenarios while preserving the characteristics of real-world conditions.

In this chapter, we plugged in simulated range measurements to the log files recorded during outdoor experiments. This allowed us to experiment with various placements for an additional beacon and simulate various effects such as temporary sensor failures and noisy or outlier measurements.

10.4 Field Experiments with ASVs

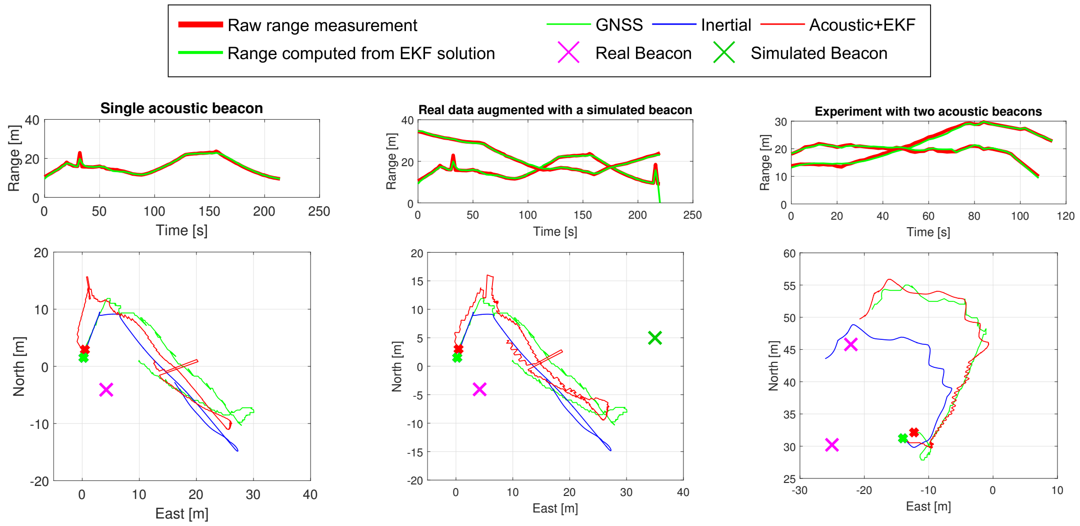
We performed several experiments with one and two surface beacons in a shallow area of Lake Geneva, close to a boat pier. The first beacon was deployed from the shore, and the second was mounted on an ASV and deployed off-shore. The ASV was programmed to hold a static position.

In order to be able to use the GNSS position as ground-truth for validation of the proposed system, we mounted the acoustic receiver on a second surface vehicle. This ASV served as a proxy for the AUV and did not use the GNSS position for localization. It was simultaneously performing inertial position estimation, as well as fusing acoustic range measurements into its inertial estimate using an EKF. The inertial estimates and the EKF output were both logged.

10.4.1 Acoustic Navigation

To begin with, we deployed a single transmitting beacon emitting a ranging pulse once every 2 s. The estimated trajectory of the robot is shown in Fig. 10.7a. The quality of the position estimate in this case will depend on the position and direction of motion of the vehicle relative to the beacon.

Fig. 10.7b shows an instance of a data-in-the-loop emulation with an added virtual beacon. As expected, adding an additional beacon or increasing the relative separation between the beacons improves the position estimate. To evaluate the accuracy of the estimate, we compute



(a)

(b)

(c)

Figure 10.7: Trajectories estimated by EKF by fusing inertial estimates and acoustic range measurements. In the top row, raw range measurements (after outlier rejection) are compared with range computed from the EKF solution. Plot (a) shows an experiment with a single beacon. In (b), a virtual beacon with simulated range measurements is added to the data logged during the mission in (a). This helps in experimenting with beacon placement and studying its impact on navigation accuracy. (c) shows an experiment with two beacons, the range measurements from which are used by a receiving vehicle to perform online real-time position estimation. Ground truth trajectory obtained from GNSS is shown in all the three plots.

the Root Mean Square Error (RMSE) in the estimated trajectory using GNSS as ground truth. The errors for one and two beacons and various relative beacon placements are summarized in Table 10.1.

Finally, we deployed two real beacons and repeated the experiment, with the beacons transmitting their ranging pulse every alternate second. Fig. 10.7c shows the estimated trajectory from this experiment.

10.4.2 Outliers and Range Errors

Fig. 10.8 shows a plot of raw range measurements and the measurements that were accepted for fusion with the EKF into the position estimate. In practice, outliers with a large error are easily rejected since they have a low likelihood. However, outliers with smaller errors have a larger potential to influence the state estimate. While range measurements help bound the absolute error in the robot trajectory, outlier range measurements result in discontinuities and jumps in the estimated trajectory. There are a number of methods to address these in post-processing.

Method / Baseline	Trajectory RMSE [m]
Inertial (Fig. 10.7a and 10.7b)	4.26
Single beacon (Fig. 10.7a)	2.90
Real+virtual / 30 m (Fig. 10.7b)	2.16
Real+virtual / 45 m (plot not shown)	2.00
Real+virtual / 60 m (plot not shown)	1.87
Inertial (Fig. 10.7c)	6.90
Two beacons / 16 m (Fig. 10.7c)	2.10

Table 10.1: RMSE over the trajectory for purely inertial localization and localization with acoustic range measurements. For the case where two beacons are used, the inter-beacon distance is mentioned.

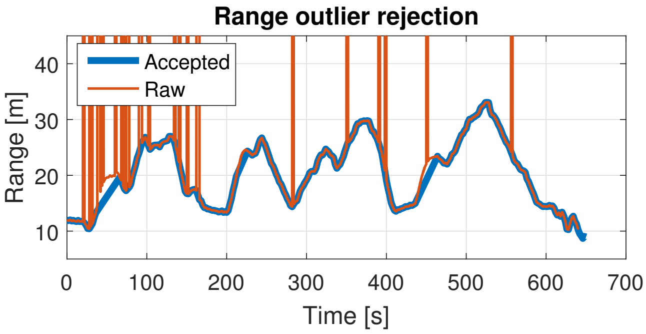


Figure 10.8: Raw range measurements with outliers or spurious range measurements (caused by motor noise) and range measurements that are accepted for fusion using EKF.

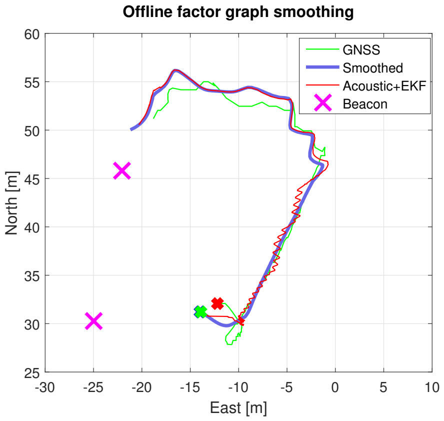


Figure 10.9: Factor graph smoothing is applied offline in post-processing to the trajectory estimated by EKF.

10.4.3 Factor Graph Smoothing

Offline factor graph smoothing is performed on the estimated trajectory using the GTSAM library, as explained in Section 10.2.5, and the result is shown in Fig. 10.9. This results in a smooth and consistent trajectory estimate, and therefore provides consistent geo-referencing for data gathered during sampling missions.

10.5 Conclusion

This chapter presented an acoustic localization system for aiding AUV operations. Our goal was to provide the AUVs with an external positioning reference so as to bound the error in the estimated position. This is necessary to follow a pre-planned trajectory as well as to correctly geo-reference environmental measurements. Our system is easy to deploy and does not require any equipment to be installed in the environment. This was important to us because we were not allowed to leave equipment permanently installed in public places. This feature also facilitates quick measurement missions in new environments. Further, our system is scalable in the number of vehicles since the vehicles are passive receivers of acoustic ranging signals.

We demonstrated our system with experiments in a real-world environment. The experiments presented were conducted in shallow water close to the shore of Lake Geneva, which is a challenging environment for acoustic-based localization methods. We used kinematic constraints of the robots to reject outlier range measurements.

A major drawback of the system is that the beacons need to be static since their positions are transmitted to AUVs before they submerge. In the next chapter, we will embed data in acoustic signals to communicate beacon positions. This will eliminate the need for the beacons to be static. Further, in the subsequent chapter, we will see why incorrect parameters of the model used for inertial navigation result in large jumps in trajectory after range measurements (this would also be true for GNSS or any external positioning references). We will show how this can be avoided by estimating those parameters online.

11 Acoustic Communication and Enhanced Localization

ACOUSTIC signals, as we saw in the previous chapter, are commonly used for signaling and communication in the underwater domain. This chapter describes our approach to embedding data in acoustic signals for underwater communication. Essentially, we implemented our own acoustic modem using the transceiver hardware. At the time of this work, there were no commercially available equivalents within our size and weight constraints. However, later in the project, some new products became available and were acquired for our work.

Having communication relaxes the need for the acoustic beacons to be static since they can transmit their own position along with a ranging pulse. The positions of beacon-carrying ASVs can now be reconfigured while the AUV is in operation, increasing the flexibility of the system. This is a capability we will exploit later in the project to improve localization accuracy. In this chapter, a brief introduction to underwater acoustic communication is provided, followed by an explanation of our approach and results from field experiments.

11.1 Underwater Acoustic Communication

Underwater acoustic communication is a challenging problem, due to the peculiar characteristics of the medium as well as the carrier waves. Higher frequency acoustic waves attenuate faster, and lower frequency acoustic waves limit the data throughput. Further, acoustic waves travel at a lower speed and therefore have a high propagation delay.

In [77], the author notes that local phenomena in water bodies such as turbulence and temperature gradients cause strong amplitude and phase fluctuations in acoustic transmissions. Then, the fluctuations in the received signal are dependent on the frequency and on locations of the transmitter and receiver. In addition, the received signal is also contaminated by time-varying interference between several reflected propagation paths. Such effects usually increase with distance and cause interference between consecutive data packets, as highlighted by Stojanovic et. al. in [78]. The use of multiple carrier frequencies provides significant

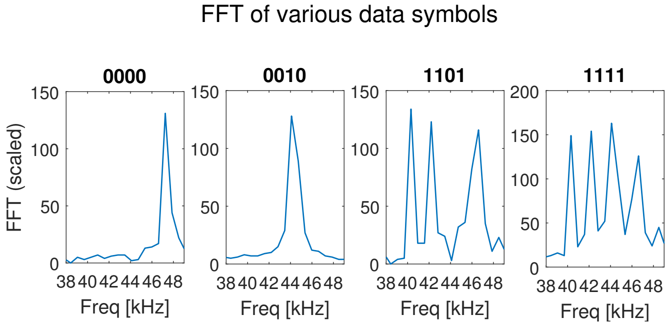


Figure 11.1: FFT of various signals that encode different binary numbers. Each bit is represented by presence of a different frequency. When all bits are zero, a fifth frequency is added to the symbol instead of transmitting a silent signal. The symbols here represent the numbers 0 (binary 0000), 2 (binary 0010), 13 (1101) and 15(1111).

improvement in bandwidth and offers robustness against frequency-selective channel effects. However, this approach is susceptible to Doppler shifts on moving devices, which need to be estimated and compensated. These measures end up increasing receiver complexity [79].

In our system, we do use multiple carriers to enhance bandwidth but have a conservative spacing between the carrier frequencies to minimize Doppler distortion. Further, we transmit short acoustic pulses (around 10 ms long) and have long guard times between data packets to avoid interference between packets. In effect, we trade off bandwidth for decoder simplicity.

11.2 Implementation and Error Correction

We used the same acoustic transceiver module introduced earlier for implementing acoustic communication. On the transmitter side, we generated modulated signals with data bits. On the receiver side, we processed the signal (demodulation) to recover the data bits. We performed initial tests using hardware-in-the-loop emulation as well as by deploying transceivers in Lake Geneva. This allowed us to quickly test a range of different parameters for modulation and demodulation.

The transmitting hardware has a resonance between 39-49 kHz due to a combination of the physical characteristics of the piezo and the electrical characteristics of the transmitter amplifier. This gives us a bandwidth of 10 kHz in the 44 kHz range. For acquiring the received signal data, the sampling rate of the ADC was set to 160 kHz, and the signal processing was performed using the hardware DSP unit.

11.2.1 Data Modulation and Demodulation

We employ the Frequency-Division Multiplexing (FDM) approach for modulating data into an acoustic signal. We divide a data packet into individual symbols, where each symbol is an acoustic pulse of a duration of 1.5 ms. Each symbol is a sum of sinusoids of up to four carrier frequencies and encodes four bits, as shown in Fig. 11.1. The presence of a frequency indicates

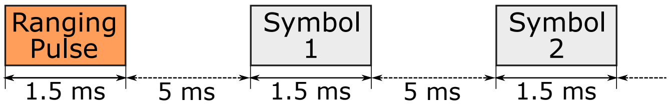


Figure 11.2: The chain of individual signals (ranging pulse and data symbols) transmitted by each beacon. Each signal is 1.5 ms in duration and is followed by a pause of 5 ms before the next signal is transmitted. The transmission is triggered by the PPS signal from the GNSS receiver.

that the corresponding bit is 1, and an absence denotes 0. Therefore, the signal encoding the bits $[b_0, b_1, b_2, b_3]$ is mathematically represented as

$$s(t) = \begin{cases} \sum_n b_n \cdot A \cdot \sin\left(2\pi f_n t\right), & \text{at least one bit is non-zero.} \\ A \cdot \sin\left(2\pi f_0 t\right), & \text{all bits are zero.} \end{cases} \tag{11.1}$$

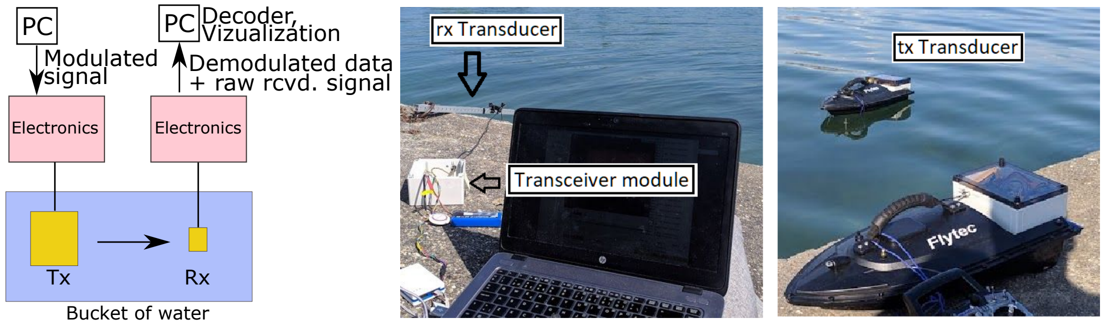
Here, f_n is the nth carrier frequency, and A is the amplitude of the signal. We use four carrier frequencies in the range 40-46 kHz, spaced by 2 kHz. When all bits are zero, the signal $s(t)$ is replaced by a single frequency sinusoid of frequency $f_0 = 47$ kHz, to avoid having a 'silent' symbol. With four carrier frequencies, we have sixteen unique symbols (representing numbers from 0-15). These symbols are precomputed and stored for efficient transmission.

In addition to offering robustness against frequency-dependent channel distortions as explained in [79], this FDM approach has the additional advantage of simple demodulator implementation. On the receiver side, demodulation is carried out by computing a Fast Fourier Transform (FFT) of the received signal, which is efficient to do using the hardware DSP unit of the microcontroller. The values of individual bits are then deduced by identifying the frequency components that are present in the FFT. Also, a conservative spacing of 2 kHz avoids Doppler-shift-induced distortions for relative velocities that the robots are capable of.

11.2.2 Ranging Pulse and Beacon Position Broadcast

We use an absolute time-based approach described earlier in Chapter 10, Section 10.2.2 to schedule transmissions from multiple beacons. During their slot, transmission is triggered by the PPS signal from the GNSS receiver. First, a ranging pulse is transmitted which is used for time-of-flight based range computation, followed by the data packet consisting of a series of symbols. In the time domain, the ranging pulse and each symbol have a duration of 1.5 ms. They are each separated by 5 ms. These parameters were chosen to be conservative, and can be lowered if needed. The transmission scheme is illustrated in Fig. 11.2.

Since all transmitters are deployed at the surface, we only encode two dimensions of the position. Further, working in a local level frame with origin at the launch point allows us to represent position in meters with smaller numbers, and hence encode them with a smaller number of bits. We round the coordinates to the nearest integer and encode them as a signed 8-bit integer, Although this appears to limit the positions (and hence, the operational area)



(a) (b)

Figure 11.3: (a) HIL emulation setup and (b) outdoor, lake test setup for testing acoustic communication. tx=transmitter (mounted on ASV), rx=receiver (deployed from shore).

from −127 to +127 m in each dimension, the number of bits can be scaled up easily by using additional carriers per symbol or additional symbols per transmission. A total of 32 data bits are transmitted, which include 16 bits of position information and an additional 16 bits to mitigate bit errors (explained later in Section 11.2.5).

The receivers constantly look for a ranging signal, and on detecting it, record the time stamp, deduce time-of-flight and compute the range. Next, the data reception routine is triggered which acquires the individual symbols in the data packet based on the known time intervals shown in Fig. 11.2. The demodulator then uses the series of individual symbols to produce a bit stream.

11.2.3 Test Setup

We performed initial tests for acoustic communication using HIL emulation, where the acoustic modules were used only for operating the transducers, which were immersed in a water container, as shown in Fig. 11.3a. Instead of the robots, the modules were connected to a desktop computer, where the signal processing was performed. Subsequently, we tested acoustic communication in the lake between an ASV and a transceiver deployed from the ground, connected to a computer, as shown in Fig. 11.3b. A known set of data were transmitted from the ASV to detect errors on the receiving side. The receiver sent raw and processed signals to the computer over a serial link.

For quick visualization of the functioning of the decoder, a new user interface was added to the ground station software, shown earlier in Fig. 6.5a in Chapter 6. It displayed the FFT of the received signal in real-time, as well as indicated which bits were correctly or erroneously decoded. It allowed us to quickly test various modulator parameters and understand the causes and sources of errors. We evaluated performance in terms of bit-error rates and frequency of undetected packets by varying the duration of each symbol and the guard-time in between symbols. In general, we found better performance with higher values of both the timing parameters, until the performance plateaued. Note that for the experiments in this

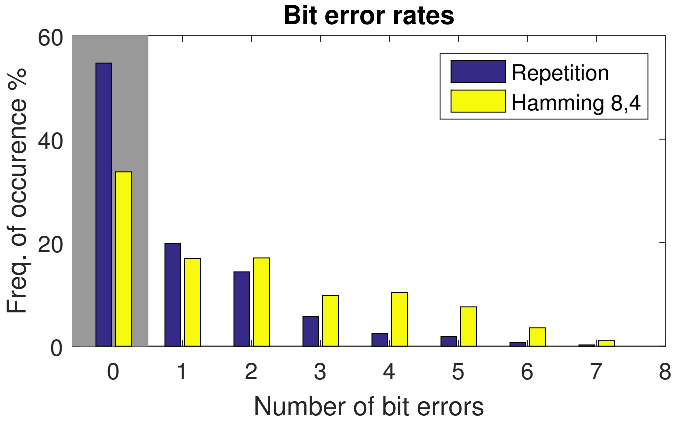


Figure 11.4: Bit error rate with two approaches of encoding, Hamming 8,4 error correction code and a one-time repetition. One-time repetition performs significantly better, as shown by the occurrence of zero bit errors (shaded region) in both approaches.

chapter, our throughput requirement was very limited, allowing us to use conservative timing parameters.

11.2.4 Data Throughput

Each symbol in our approach carries 4 bits of data. We transmit a total of eight symbols, i.e., 32 bits (or 8 symbols) for broadcasting position. Fig. 11.2 shows that the duration of the signal with 32 bits of data (including guard times) is 52 ms. Therefore, per transmission slot, we transmit 32 bits of data in 52 ms.

The current configuration offers room for improving the data throughput further. In theory, additional symbols can be chained to the transmission to communicate more data. The limitation is that we need guard times between full acoustic transmissions to allow echoes to fade out. This is especially important in shallow water. While we have not empirically determined the minimum guard time required, the current configuration is extremely conservative and has room for adding more data. Secondly, it is possible to improve the overall throughput by adjusting the symbol and inter-symbol pause durations. However, smaller symbol durations may make communication more susceptible to external disturbances. The time taken to process each symbol in our hardware is about 3 ms, which is slightly smaller than the length of the inter-symbol pause. But computation capability imposes a limit on this parameter.

11.2.5 Mitigating Bit Errors

Given that the underwater acoustic channel is highly susceptible to communication errors, measures to detect and correct them are necessary. We attempted to do so with two separate approaches. First, we implemented the Hamming (8,4) error-correcting code [80], which encodes 4 bits of data into a packet of 8 bits by adding 4 parity bits. It can detect up to two bit

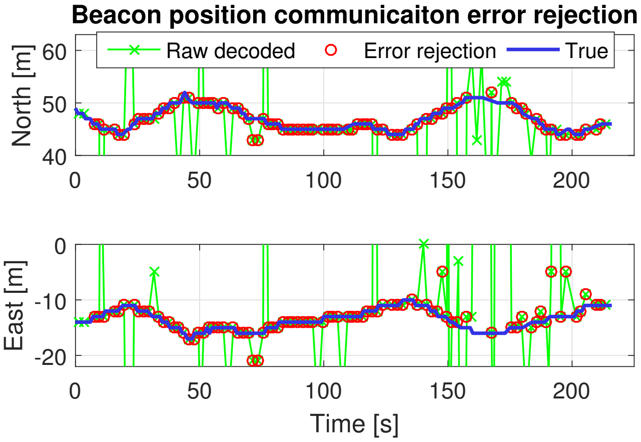


Figure 11.5: True beacon positions (obtained via radio communication) are shown along with raw decoded positions obtained via acoustic communication and the result of applying error detection based on kinematic constraints. Rejecting erroneous beacon positions improves the overall quality of navigation by minimizing inconsistent range updates.

errors or correct a single bit error. Second, we implemented a repetition based-method, where each data bit is transmitted twice. The repetition is done in such a way that two different frequency carriers are used to represent the same data bit. Error detection is done by checking if a decoded bit and its repetition are in agreement. This approach performed better than the Hamming approach because of the characteristics of the channel and the transducer.

A comparison of bit error rates after detection and rejection by both methods is shown in Fig. 11.4. The Hamming approach works well with symmetric binary errors, i.e, bits 1 and 0 are equally likely to be misrepresented. This is not true for the OFDM approach used by us, where it is unlikely that a bit 0 will be corrupted into bit 1 (which essentially translates to a frequency component being detected when it is not present at all), while the opposite is more likely.

11.2.6 Rejecting Range Outliers and Erroneous Beacon Positions

We use the approach discussed in Section 10.2.4 for range outlier rejection. However, localization errors can also be caused by erroneous beacon positions. Despite measures for mitigating bit errors in acoustic communication, these errors are unavoidable. These errors corrupt the beacon positions obtained through acoustic communication and result in inconsistent localization updates.

Since beacon positions are subject to kinematic constraints relative to the AUV, some erroneous positions can be filtered and rejected. We use the corresponding range measurement and the estimated position of the AUV to check the consistency of the received beacon positions. Fig. 11.5 shows a comparison of true beacon positions, raw decoded positions, and the set of accepted positions after the filtering routine. About 24 % of the positions are rejected and the impact of communication errors on the navigation solution is reduced. Note that for this comparison, the error detection explained in Section 11.2.5 was not used.

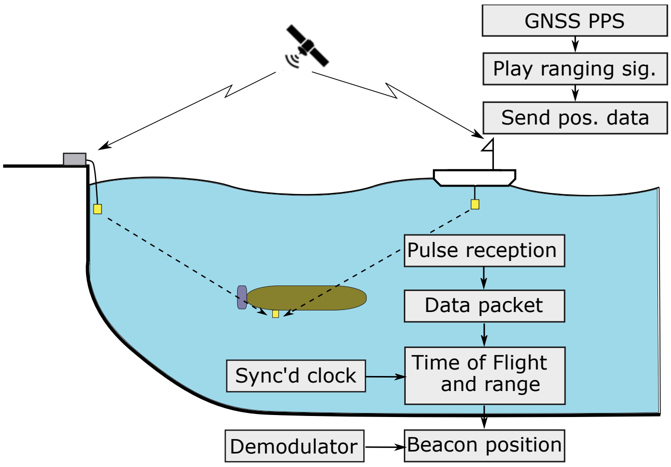


Figure 11.6: One beacon is deployed from the shore and another is mounted on an ASV. Both beacons transmit an acoustic signal according to a fixed schedule. The acoustic signal comprises of a ranging pulse followed by a data packet that encodes their position. With synchronized clocks, the AUVs can measure one-way travel time of the ranging pulse and compute range. It then decodes the data packet to obtain the beacon position and perform a position update.

11.3 Field Experiments with ASVs

Experiments to evaluate localization accuracy were performed in a shallow region of Lake Geneva. The water depth in the experimental area was between 1-4 m, and experiments were performed in low to medium wind conditions (< 5 m/s). The shallow water depth exacerbates the challenges of acoustic ranging and communication to a small but significant extent.

We deployed one beacon mounted on an ASV and another from the shore. The shore-deployed beacon was used for logistical simplicity, and can also be deployed from an additional ASV. While it was stationary, this information was not used for navigation. It was treated like a moving beacon and its position was obtained via acoustic communication. A schematic of the deployment and the sequence of events from ranging pulse transmission to reception are also shown in Fig. 11.6. The setup was similar to that in the previous chapter.

We used a second ASV with an acoustic receiver as a surrogate for the AUV. Beacon positions received over the radio and GNSS positions are used as ground truth. To analyze the efficacy of our method, we first performed a number of experiments and recorded the navigation output as well as all sensor and communication data. Then, we performed data-in-the-loop emulation by augmenting data from real experiments and recomputing the navigation solution offline for comparison. In particular, we replaced the beacon positions obtained via acoustic communication with true beacon positions. This was done to provide the baseline performance for navigation. We present results from simulations as well as experiments gradually increasing in complexity and compare the accuracy of the estimated trajectory.

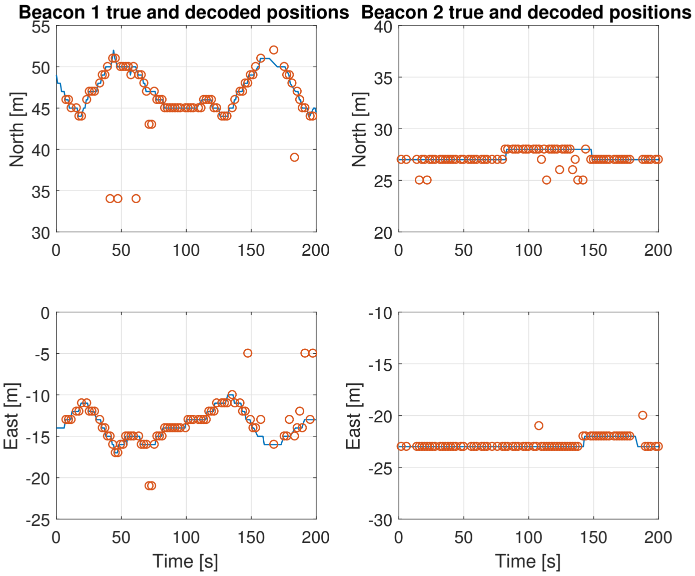


Figure 11.7: Beacon positions as decoded by the receiving ASV (red circles) compared with true positions of the beacons (blue lines). While some errors in decoded positions are incorrectly accepted by the filter, the beacon positions are largely accurately tracked.

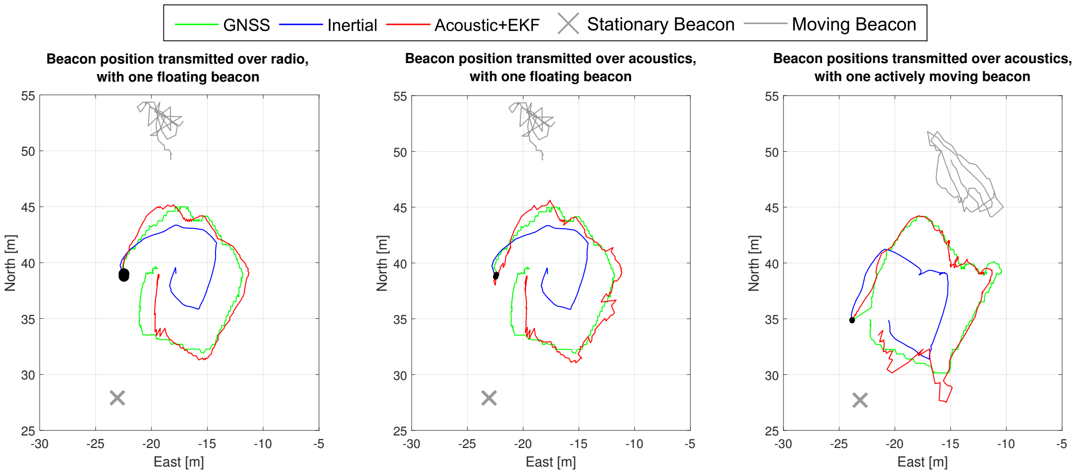
11.3.1 Tracking Beacon Positions

We deployed a transmitting beacon mounted on an ASV that was mobile, while keeping the receiving ASV stationary (but not actively holding position). The transmitting ASV obtained its position from the GNSS receiver, transformed it to the local-level NED frame, and sent its position via radio as well as acoustic communication. The receiving ASV received the acoustic signal and stored the decoded position information. It also performed rejection and filtering of erroneous positions as explained in Section 11.2.6. Fig. 11.7 compares the true beacon positions and those obtained by the receiving ASV over acoustic communication. The receiving ASV is able to accurately deduce the position of the beacon.

11.3.2 Localization

We present results of acoustic navigation from three different experiments with increasing complexity. To begin with, the beacon ASV was not programmed to move but was also not actively holding position either. To demonstrate the effect of error-prone acoustic communication on the navigation solution, we first performed a DIL emulation using true beacon positions (obtained over radio). The estimated trajectory from this simulation experiment is shown in Fig. 11.8a. The estimated trajectory from the corresponding real experiment (from which data were recorded for DIL emulation) is shown in Fig. 11.8b. In this case, the positions of both beacons were sent over acoustic communication.

Finally, the beacon ASV was programmed to travel back and forth between two waypoints. This was done to study the accuracy of the navigation solution when the beacons move. The



(a) (b) (c)

Figure 11.8: Trajectories estimated by EKF by fusing inertial estimates with acoustic range measurements and beacon positions. Plot (a) shows a simulation experiment where one beacon is stationary and the other is mounted on a passively floating ASV. Beacon positions were transmitted over radio. Plot (b) shows a real experiment that has the same setup as (a) but beacon positions are transmitted via acoustic communication. Real trajectory data from this mission are used to perform the simulation in (a). Plot (c) shows a real experiment where one of the beacons is stationary as before while the other is actively moving between two waypoints, and beacon positions are transmitted over acoustic communication. Ground truth trajectory (from GNSS) is shown in all plots.

Method / Baseline	Trajectory RMSE [m]
Inertial (Figs. 11.8a and 11.8b)	3.55
Floating beacon, radio communication (Fig. 11.8a)	1.17
Floating beacon, acoustic communication (Fig. 11.8b)	1.21
Inertial (Fig. 11.8c)	3.52
One moving beacon, acoustic communication (Fig. 11.8c)	1.66

Table 11.1: RMSE over the trajectory for purely inertial estimates and those estimated with acoustic ranging.

estimated trajectory from this experiment is shown in Fig. 11.8c. As expected, the estimated trajectory in this case has the highest error. The RMSE between the estimated and true (GNSS) trajectory is tabulated in Table 11.1.

11.4 Commercial Hardware

Due to difficulties in the production of the transceiver hardware, we were unable to equip all robots with acoustic communication and to have spare hardware to mitigate operational damages. Therefore, we decided to use commercial communication and localization systems, which became available recently. We acquired the DiveNET Microlink acoustic transceiver [27], which is capable of acoustic ranging as well as communication, similar to our transceivers. Their size and weight are also similar. However, the main difference is that the Microlink uses

Characteristic	Our system	DiveNET Microlink
Ranging accuracy	~ 1.5 m	2-3 m
Ranging method	One-way TT	Two-way TT
Time to measure range	Propagation time	1.25 s + 2 × Propagation time
Data throughput	32 bits in 52 ms	78 bits/s
Data error rate	~ 40 %	< 8 % (empirical)
Time to send data	Propagation time	1.0 s + Propagation time

Table 11.2: Comparison between ours and DiveNET Microlink acoustic transceivers.

two-way-travel-time (TWTT) to perform range measurements, which will affect the scalability of the system presented in this chapter. While the data throughput of the Microlink is lower than what was achieved in this chapter, the error rate is significantly better. The differences are shown in Table 11.2.

The subsequent work on acoustic localization in this thesis will be based on TWTT. The DiveNet Microlink transceiver will be introduced in detail, along with its error characteristics in Chapter 14, Section 14.3. Experiments with the system are performed in Chapter 15. We also acquired a tracking system based on acoustic range and bearing, which will be introduced and used in the next chapter, Chapter 12.

11.5 Conclusion

This chapter builds on the multi-beacon acoustic positioning system developed for the AUVs and adds communication capability to the acoustic transceivers. Communication relaxes the requirement that the beacons be static and makes the system more flexible. The beacon-carrying ASVs can now move if needed. In the next part of the thesis, we will exploit this capability to actively position the beacon-carrying ASVs. They will be positioned such that they provide improved quality of localization aid to a group of AUVs.

We retained the scalability feature of the older system, since beacons are still the only active transmitters, and AUVs are passive receivers (much like GNSS satellites and receivers). However, in more complex applications where the AUVs also need to transmit some data, the system would no longer have unlimited scalability.

A common problem with most acoustic localization systems is that the duration between measurement updates is long, of the order of a few seconds. In TWTT-based systems (such as the DiveNET Microlink), the update interval increases with increasing number of AUVs. Then, it is important that in between acoustic updates, robots can perform accurate inertial localization. Towards that end, the next chapter addresses the tuning of parameters of the dynamic model of the robots used for inertial localization.

12 Parameter Estimation for Inertial Localization

INERTIAL localization with good accuracy is crucial for underwater robots since external positioning references are scarce. The measurement intervals of acoustic positioning systems are often several seconds long. In between these updates, the robots need to rely on inertial localization. This chapter presents a framework for estimating kinematic and dynamic model parameters used for inertial localization. It can run online while the robot is in operation, as well as offline using recorded log files, with either choice providing different advantages. With optimal parameter estimates, robots can tolerate longer intervals in external positioning updates for a specified acceptable level of estimation error.

The proposed approach only uses external position information from the sparsely available GNSS or acoustic measurements. When a new measurement is available, an optimization framework is used to update the parameters such that the sum of update errors within a specified time window is minimized. Each time the parameters are updated, possible errors in past trajectory are also corrected by tracing a desired length of the history and recomputing the state estimates. The following sections explain the motivation behind the estimation framework, followed by a detailed description of the method. Results from field experiments demonstrating improvement in trajectory estimation accuracy are presented.

12.1 Introduction and Motivation

Inertial sensors of high accuracy may be sufficient to perform inertial localization in the absence of external positioning references. However, sensing modules compatible with miniature AUVs (e.g., MEMS-based devices) do not offer the required accuracy. Fusing these measurements with a dynamic model of the robot enhances the accuracy of the position estimates [62]. This requires knowledge of a wide range of parameters that make up such a model. Measuring or estimating some of the parameters (such as hydrodynamic constants) can be difficult, time-consuming, or even require special equipment. Further, some parameters such as mass and actuator properties may differ from one robot to another. They may also change, for example when new equipment or sensors are added to the robot.

The motivations behind this work are three-fold. Firstly, we seek to improve the accuracy of inertial localization in between successive external position updates. Secondly, we aim to correct errors in the past trajectory by recomputing it using the newly obtained parameter estimates. This is important for adaptive sampling methods such as [81], which rely on accurately geo-referenced past measurements to decide future sampling locations. Finally, we aim to calibrate the parameters for several robots. Identical robots may have slightly different parameters owing to different on-board equipment and differing wear and tear over time.

We used GNSS or an acoustic range and bearing system for obtaining absolute positions, which are eventually used for estimating the parameters. Both systems are less accurate and have low update rates (~ 1 s for GNSS available only on surface, a few seconds for acoustics) compared to the systems used in the state-of-the-art introduced in Chapter 7.2.3. Note also that the systems we use only measure position, not full pose. Suitable tracking systems for accurate pose measurements at a high frequency for outdoor aquatic environments are unavailable. Indoor or controlled environments such as pools are too small to perform adequate motion.

Our work takes an approach similar to other error-minimization-based methods. However, to deal with sparse position measurements, a timestamped history of all robot control and measurement updates is maintained. We use the sum of Kalman update errors (product of Kalman gain and innovation) over a window of time as the error metric to be minimized. When new external position information is available, error minimization is performed over a specified window of the measurement history that includes several position measurements.

Our approach offers several advantages. The error metric accounts for uncertainty in measurements and state estimates. This is crucial for acoustic positioning devices, since the uncertainty in their measurements is directional and dependent on relative positions of acoustic beacons and robots, as explained later. In addition, this formulation makes it easy to integrate our method into any kind of state estimation framework with minimal modification to it. In this chapter, we plugged the parameter estimation procedure into an existing EKF framework. Finally, our approach allows for correction of possible errors in past trajectory by tracing a desired length of the history and recomputing past estimates using updated parameters.

The proposed method can run both online or offline, each providing different benefits. Online execution uses on-board information to estimate parameters on-the-fly, but the computational budget is limited. Such procedure is well-suited for parameters that can change quickly such as drift in yaw angle. Offline execution, on the other hand, is not constrained by computational cost and can perform estimation using a larger amount of data. Such procedure is well-suited for parameters that are likely to remain constant or change rarely, such as the physical properties of the robot.

Variable	Description
θ_e	Offset between actual and estimated heading
s_x	Water stream velocity x (ground frame)
s_y	Water stream velocity y (ground frame)
$k' = k/m$	Ratio of thrust constant to mass
$k'_x = k_x/m$	Ratio of drag constant to mass
$k'_{yz} = k_{yz}/m$	Ratio of drag constant to mass

Table 12.1: List of variables to be estimated.

12.2 Preliminaries

Inertial position estimation uses several kinematic and dynamic parameters. The dynamics model of the robots was presented in Chapter 8.1.3. This section lists the parameters we aim to estimate and uses the EKF measurement updates to define an error metric.

12.2.1 Parameters to Estimate

We consider the following set of parameters, **P**, to be estimated.

$$\mathbf{P} \triangleq \left[\theta_e, s_x, s_y, k', k'_x, k'_{yz}\right]. \tag{12.1}$$

Table 12.1 provides a description of the parameters. The last three variables correspond to the dynamic model parameters in Eq. (8.7) and always appear as a ratio with mass. Therefore, the ratio itself is estimated.

The parameters can be categorized into two types – *fixed parameters* do not change often, such as the vehicle model parameters, while *variable parameters* may change depending on the situation during operation, such as the heading offset (due to yaw drift of the IMU) and water stream velocities.

12.2.2 Error Model

We use the Kalman measurement update equation (see Eq. (8.24)) to obtain the error associated with a measurement. The ‘update error’ e_i is defined as the product of the Kalman gain and innovation.

$$e_i = K_i \left(z_i - g(\hat{\mathbf{X}})\right), \tag{12.2}$$

where z_i is the ith measurement, K_i is the Kalman gain computed during this measurement as per Eq. (8.23), and $g(.)$ is the measurement model of a particular sensor. The key advantage of this metric is that the uncertainties in the measurement and the state estimate are accounted for. Due to the adaptive weighting in the Kalman gain in Eq. (8.23), for a measurement with a large uncertainty, the resulting error will be lower. We will explain later that for an acoustic range and bearing system, the uncertainty in the obtained position depends on the range. The sum of errors corresponding to a set of measurements is used in the optimization framework.

12.2.3 Initialization

The initial position is obtained from GNSS before launching the robots. The initial velocity and acceleration are assumed to be zero (acceleration due to gravity is accounted for). The initial heading, if unknown, will result in a large θ_e, which will be estimated as a parameter eventually. The gyro biases are internally estimated by the XSENS IMU module by leaving the robot stationary for a few seconds.

12.3 Optimization Framework

The optimization framework seeks to obtain a set of parameters that minimize the sum of update errors defined in Eq. (12.2) over a specified window of time. To do so, it maintains a history of all measurements and evaluates the error over the corresponding subset.

12.3.1 History Management

We maintain a timestamped, chronological history of information pertaining to robot state updates, as illustrated in Fig. 12.1. This includes all the measurements (IMU, acoustics, GNSS etc.) as well as motor commands output by the controller. Each of these entities is called an *update node*, denoted by u_i^t, where i denotes the index of the node, and t the timestamp. Note that multiple update nodes may have the same timestamp, for example, when there are concurrent measurements from two sensors. We also record the initial state of the robot at $t = 0$. Additionally, we record the state at points of time when absolute position measurements (GNSS or acoustics) are received. These are called *anchor states.*

Starting from an appropriate anchor state, we can use the chain of update nodes for successive EKF updates to compute the estimated robot state, $\hat{\mathbf{X}}$, at any point of time in history. We will also need the values of parameters in $\mathbf{P}$.

$$\hat{\mathbf{X}}^{t_2} = \mathscr{F}_{t_1}^{t_2}\left(\mathbf{X}^{t_1}, \mathscr{U}, \mathbf{P}\right), \tag{12.3}$$

where $\mathscr{F}_{t_1}^{t_2}$ is some function that performs this computation from t_1 to t_2, and $\mathscr{U} := \{u_i^t \mid t \in [t_1, t_2]\}$ is the sequence of all the update nodes between the two points of time. In Fig. 12.1, the computation of state at t is illustrated. Note that we treat motor commands as a type of measurement, and define the corresponding update error to be zero.

12.3.2 Optimization

A series of EKF updates are performed in Eq. (12.3). The net error in the calculation of the estimated state is defined as a function of the parameters, $\mathbf{P}$ as

$$E_{[t_1, t_2]}(\mathbf{P}) = \sum_{\mathscr{U}} \|e_i\|^2, \tag{12.4}$$

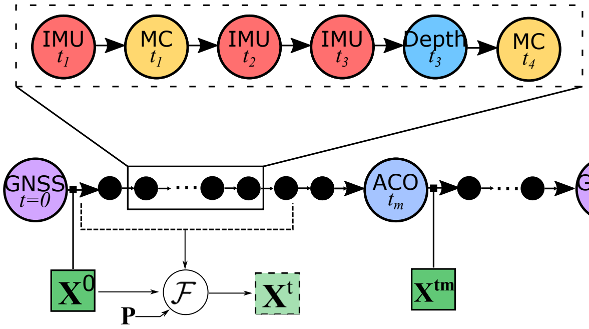


Figure 12.1: Timestamped history consisting of update nodes representing state transition information. Each colored circle corresponds to an update node, for e.g., IMU measurements, Motor Commands (MC), etc. Anchor states, shown in dark green squares, are recorded after an acoustic or a GNSS position update. The function $\mathscr{F}$ computes the state at t using the previous anchor state and the intermediate sequence of update nodes.

where $\mathscr{U}$ is the set of nodes within the time window from t_1 and t_2, and e_i is the update error corresponding to node i, as computed in Eq. (12.2). We seek to find values of parameters **P** that result in the most accurate inertial positioning. To do so, we need the inertial estimates to be in agreement with absolute position information. We achieve this by minimizing the measurement update error in Eq. (12.2) over a time window $[t_1, t_2]$, spanning several GNSS or acoustic measurements. Formally, we seek to solve the following optimization problem:

$$\mathbf{P}^* = \underset{\mathbf{P}}{\arg\min}\ E_{[t_1,t_2]}(\mathbf{P}). \tag{12.5}$$

For a valid solution, at least two absolute position measurements are required within the time window.

12.3.3 Past Trajectory Correction

Once optimal parameter estimates are obtained for a particular window $[t_1, t_2]$, the past states are recomputed again with the newly obtained parameters, starting from an appropriate anchor state. Thus, the past trajectory of the robot is also corrected.

12.3.4 Window Size, Online and Offline Execution

The computational complexity of the optimization problem in Eq. (12.5) increases with the size of **P** and the length of the time window $[t_1, t_2]$. A larger optimization window encompasses more data and in turn increases the probability that it contains additional motion and measurements to render all parameters sufficiently observable. Further, it also results in better estimation of parameters that are likely to be static or change slowly. On the other hand, for parameters that may change quickly (such as those that depend on the local situation, e.g., stream velocity), longer time windows result in slow or no convergence. For such parameters, smaller time windows are more nimble.

Given the limited online computational budget, online estimation is performed only for variable parameters with small time windows. Offline estimation is preferred for fixed parameters, which are unlikely to change during normal operation of the robot. We gather GNSS data by running the robot on the surface and perform the estimation of fixed parameters offline. These parameters are then saved and used as fixed values during normal operation.

12.3.5 Suboptimal or Locally Optimal Estimates

Convexity of the optimization problem with respect to all the chosen parameters is not guaranteed. Therefore, it is possible that the obtained solution may be a local optimum. However, as long as the obtained parameters result in a lower inertial positioning error, a local optimum is an acceptable solution. A solution causing large errors will be updated in the subsequent optimization steps.

In practice, windows containing insufficient motion (such as at the beginning or if the robot was at rest for some time) result in large changes to parameters between successive optimization steps, before the parameters eventually converge. To avoid this, parameter bounds need to be chosen carefully, for example, corresponding to physical limitations. Further, we use the standard deviation between the last $n = 3$ parameter estimates as a measure of non-convergence.

12.4 Results from Field Experiments

Experiments and data gathering missions were performed in Lake Geneva with three ASVs and an AUV, on days with mild-to-no wind. Absolute position measurements are provided either by GNSS or by an acoustic range and bearing system. The GNSS positions are available at around 1 Hz, depending on the quality of the received signal. Acoustic range and bearing measurements are available approximately every 4.5 s. The true and estimated (inertial) positions were recorded. Errors in inertial position estimates were due to wind/water flow, unknown initial heading, heading drift, and other external factors.

The ASVs are identical but are in varying states of wear after use. In particular, the brushed motors are prone to wear and rusting after use in water. They also have slightly different mechanical attachments for external peripherals such as acoustic transceivers. Robot trajectories were pre-defined and specified as a sequence of waypoints to be visited. A waypoint is considered to have been visited if the robot reaches within a radius of 3 m.

12.4.1 Offline Parameter Estimation

We first recorded control inputs, IMU measurements, and GNSS positions from the operation of robots on the surface on several trajectories. A plot of a subset of robot trajectories is shown in Fig. 12.2. We used the recorded data to perform offline estimation of fixed parameters,

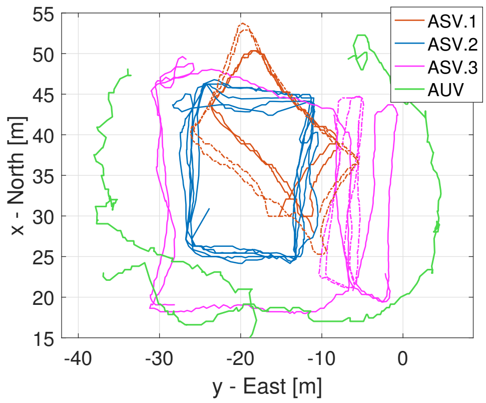


Figure 12.2: Trajectories of robots in Lake Geneva from which data are used for parameter estimation. Colors represent different robots. Experiments were performed with three ASVs and one AUV on several different days and two different locations near the shore of Lake Geneva.

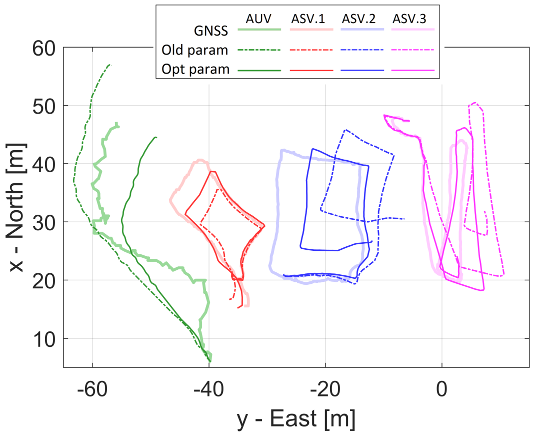


Figure 12.3: Comparison of trajectories estimated with old non-optimal as well as optimal parameters. The trajectory estimation and parameter optimization was done offline by using real robot data. The RMSE in both kinds of estimated trajectories is shown in Table 12.2. Note: the trajectories have been offset in the plot to avoid overlap for clarity.

$\left[k', k'_x, k'_{yz}\right]$ for each robot separately. The optimization window was set to encompass the entire length of a combination of several trajectories, for a particular robot, therefore using all the measurement nodes from several mission datasets. We found slight differences in model parameters across robots, as expected. Results show a significant reduction in error in the inertial trajectory computed with optimal parameters.

To demonstrate the effectiveness of our approach, we tested the new parameters on a separate testing dataset. We recomputed the inertial trajectory of the robot without using any GNSS measurements (except for initial positions) with old as well as new parameters. Fig. 12.3 shows a comparison of the estimated trajectories. Table 12.2 shows the RMSE in the trajectories computed with old and optimal parameters, as well as the values of the parameters themselves. Optimal parameters result in a significant reduction in error.

	k'	k'_x	k'_{yz}	RMSE of estimated traj.	
ASV.Old	8.0	2.0	0.3	Old param	Opt. param
ASV.1	8.1	1.71	0.155	7.62 m	3.57 m
ASV.2	10.33	1.85	0.15	3.04 m	1.66 m
ASV.3	9.26	1.82	0.14	5.72 m	1.96 m
AUV	4.42	0.55	0.07	12.95 m	6.53 m

Table 12.2: Left: Values of estimated parameters for different robots. The old, pre-optimization parameters are also shown. Right: RMSE of inertial trajectory estimated with old parameters as well as newly obtained optimal parameters. New parameters result in lower RMSE.

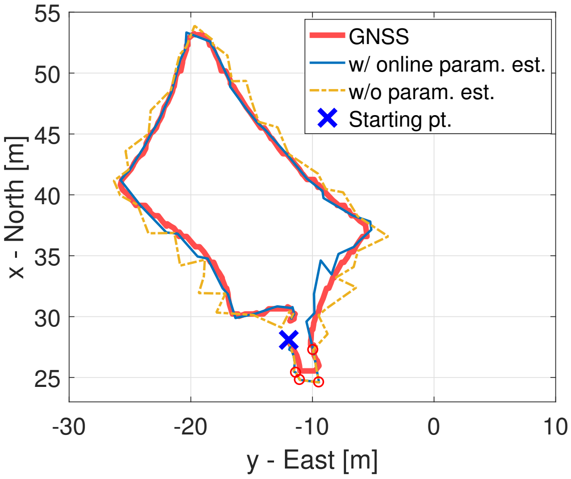


Figure 12.4: Online estimation of variable parameters. The trajectory estimates with and without parameter optimization are identical initially, shown by red circles. After a few optimization steps, optimal values of variable parameters result in a more accurate trajectory estimate.

12.4.2 Online Parameter Estimation

By using the optimal values of fixed parameters obtained in the previous step, we ran the estimation procedure online to estimate the variable parameters, $[\theta_e, s_x, s_y]$. We were able to improve the accuracy of estimation of the robot trajectory between GNSS updates. To demonstrate this clearly and to partially emulate the use of acoustic positioning devices, we restricted access to GNSS information to once in 8 s, a lower update rate than that of an acoustic system. GNSS was continually recorded for ground truth separately, however. We compared the position estimation with and without parameter estimation, using restricted GNSS updates in both cases.

Fig. 12.4 shows a plot of the estimated trajectory with and without online estimation of variable parameters. We deliberately chose a trajectory with a high initial heading error ($\sim 30°$). The optimization window was set to include the latest three GNSS updates (roughly 24 s long). The optimization was performed after each new GNSS update (roughly every 8 s). The error in

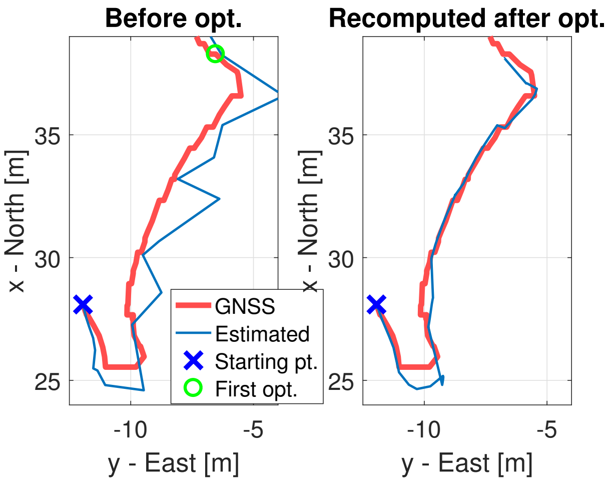


Figure 12.5: Correction of past trajectory after an optimization step. Left: The section of the trajectory where the first online parameter estimation is performed at the point marked by the green circle. Right: Past trajectory recomputed with the newly estimated optimal parameters.

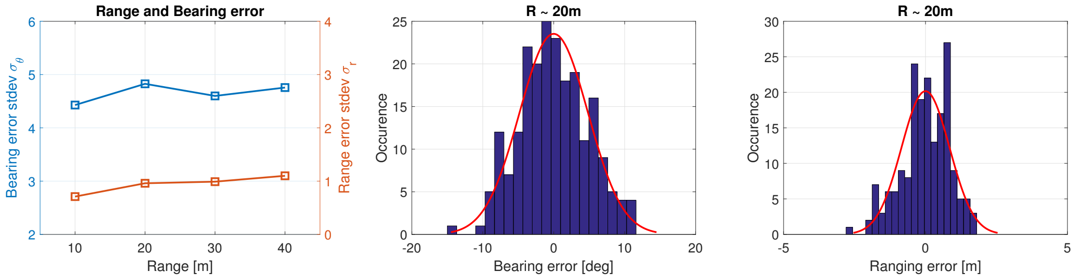
the estimate is nearly identical initially, both with and without online parameter estimation. However, after the first three optimization steps, optimal parameters result in a significantly better trajectory estimate.

12.4.3 Past Trajectory Correction

To demonstrate correction of past trajectory, we delayed the optimization process until after a few GNSS updates. At this point, we performed the optimization and used the newly estimated variable parameters ($[\theta_e, s_x, s_y]$) to trace the recorded measurement history and recompute the past trajectory. We used the same scenario as the previous section – GNSS updates once in 8 s, optimization window of 24 s. Fig. 12.5 shows the section of the trajectory after which the first optimization is performed, and the newly corrected past trajectory.

12.4.4 Online Estimation Using Acoustic Range and Bearing

Finally, we performed online parameter estimation using acoustic range and bearing measurements. We operated the robots on the surface and recorded GNSS positions for ground truth. We used the DiveNET Commander acoustic positioning system [82], which consists of a base station with a USBL receiver array, and a single channel transceiver mounted on the robots. The base station was deployed by attaching it to a rigid, submerged mount near the shore. Position measurements are initiated by the base station, which uses a query-response cycle to measure range (from two-way time-of-flight) and bearing (from angle of arrival). It then transmits this information back to the robot. Therefore, each measurement requires three transmissions in total, and takes 4.5 s, including idle time to avoid interference. With N robots served by the base station, the update interval for each robot would be $4.5N$ s.

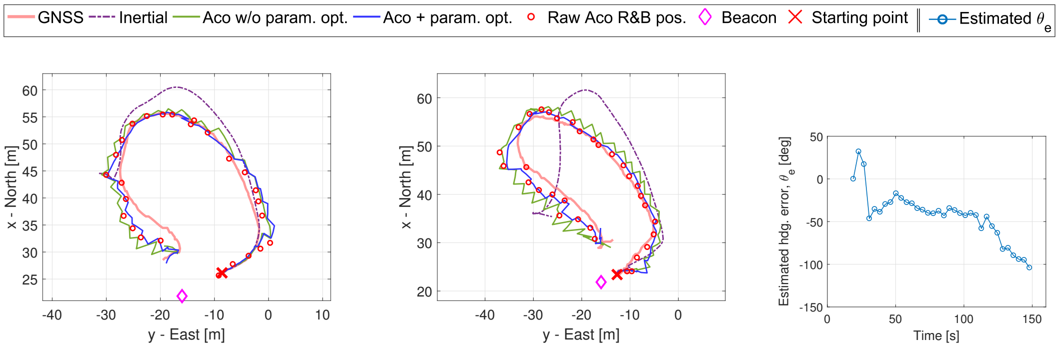


(a) Std. dev. of error.

(b) Bearing error dist. at 20 m

(c) Range error dist. at 20 m

Figure 12.6: Characteristics of error in range and bearing measurements. (a) Standard deviation of error in range and bearing at various distances (range). (b,c): Distribution in error in range and bearing measurements at 20 m range, with a Gaussian fit, after outlier rejection.



(a) Trajectory 1

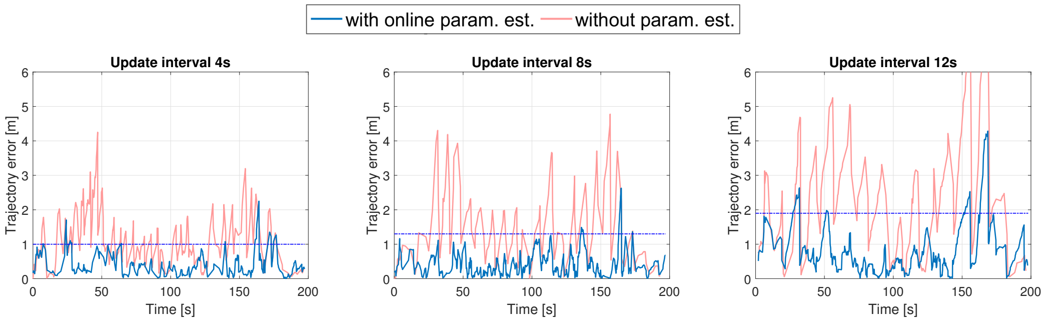
(b) Trajectory 2

(c) Heading error θ_e in (b)

Figure 12.7: (a,b) Two trajectories estimated with absolute positioning only from acoustic range and bearing measurements. GNSS data were recorded for ground truth. Estimated trajectories with and without parameter optimization are shown. (c) Estimation of heading error parameter, θ_e, in the trajectory in (b). The inertial trajectory shows increasing heading drift towards the end in (b), which is correctly estimated.

Acoustic positioning devices have lower accuracy as compared to GNSS. The error characteristics of the system (after outlier rejection) are shown in Fig. 12.6. With this system, we did not find a significant change in accuracy of range or bearing measurements with increasing distance. Note, however, that the uncertainty in the calculated position will increase (in the tangential direction) with distance. This is because in the calculation of the position in Cartesian coordinates from a range and bearing measurement, a multiplication with the range is required. This error characterization is used to set the measurement uncertainty parameters in the corresponding EKF measurement model and is accounted for by the error metric used for minimization (Eq. (12.2)).

Trajectories estimated with the acoustic positioning system are shown in Fig. 12.7. GNSS was used only to provide the initial position. In the rest of the experiment, only acoustic range and bearing measurements were used for position updates. Parameter estimation was performed after every acoustic measurement, and the window size for estimation was set to 15 s (encompassing three last measurements). The estimated trajectory with parameter optimization has better accuracy. Note that in Fig. 12.7b, the heading drift increases significantly in the



(a) GNSS interval 4s (b) GNSS interval 8s (c) GNSS interval 12s

Figure 12.8: Evolution of error in position estimates for various GNSS position update intervals. Note that the error drops each time there is an external position update. Optimal parameters result in lower error in estimated position, with 95 % of the samples below the level indicated by the blue horizontal line.

inertial trajectory towards the end. This is correctly estimated by the optimization framework, as shown by the increasing error in the estimated heading error in Fig. 12.7c, resulting in better inertial estimation in between acoustic updates.

12.4.5 Evolution of Trajectory Error

Between external position updates, when performing inertial localization, the error in the estimated position increases. With optimal parameters estimates, this increase in error can be minimized. Fig. 12.8 shows the evolution of instantaneous error in the estimated position for different GNSS update intervals. These data correspond to the trajectory in Section 12.4.2 (Fig. 12.4), which had a high initial heading error. While GNSS was continuously available, we restricted access to it to once in 4, 8, or 12 s. We then recomputed the estimated trajectory with and without parameter estimation. The window size for parameter estimation was set to include the last three GNSS updates, i.e., 3× the update interval.

With online parameter estimation enabled, once the parameter values converge, the estimated trajectory shows significantly lower estimation error. Results in Fig. 12.8 show that for a specified level of acceptable error, optimal parameter values can tolerate longer intervals between absolute position updates. This is important in the context of acoustic positioning systems, which have long update intervals, especially when shared by several robots.

12.4.6 Computational Overhead

We used gradient descent to solve the problem in Eq. (12.5) and obtain optimal parameter estimates. The most expensive component of this operation is the evaluation of the error $E_{[t_1,t_2]}(\mathbf{P})$ defined in Eq. (12.4). It involves computation of the EKF process and measurement updates corresponding to the time window $[t_1, t_2]$. This computation is performed on the computation module (1 GHz processor and 512 MB of RAM). The C++ code is compiled with `-O2` optimization flag. For experiments of usual length of about 15-20 minutes, the RAM is sufficient to store the measurement and control history.

For a time window of 15 s, the evaluation of Eq. (12.4) takes about 10 ms. In practice, online estimation of the three variable parameters $[\theta_e, s_x, s_y]$ typically requires 30-50 evaluations of the error before convergence. We have used a 15 s time window with an acoustic update interval of 4 to 5 s. This amounts to about 300-500 ms of computation time. Since the computation is performed every 5 s, this delay is affordable.

12.5 Conclusion

We have presented an approach for estimating parameters for aquatic robots, using only GNSS or acoustic absolute positioning information. It can run both online during operation, or offline using recorded data. Our approach can be easily integrated with any kind of state estimation framework. A key feature of the approach is that it can correct the past trajectory online after a parameter optimization step. This is useful in experiments where improved estimates of past trajectory can enhance performance, such as adaptive sampling.

We evaluated our approach with real-world experiments using GNSS as well as acoustic range and bearing information. We also simulated longer measurement update intervals and performed parameter estimation on data from real-world robot operation. We showed that optimal parameter estimates result in better position estimation in between external position updates. We also showed that by estimating variable parameters online, we can tolerate longer update intervals for a given level of acceptable position error.

We performed estimation of fixed parameters offline for several robots. We found that the parameter values differ for different robots of the same type. This is due to differing levels of wear after use and different on-board peripherals. Our approach allows calibration of these fixed parameters automatically for several robots, saving repetitive effort.

In the context of aquatic robotics, communication and external position references are sparse resources. Our approach addresses this challenge by exploiting computation, which has become inexpensive in both price and power consumption in recent years.

We used the Kalman update error as the residual to be minimized for theoretical reasons. In future, we will evaluate other formulations of the minimization problem, in particular using Maximum Likelihood. Further, we have not considered the relative positions between the GNSS receiver, IMU, and acoustic receiver, assuming them to coincide with the center of mass. Given that the localization errors in aquatic robots are of the order of few meters, this assumption is warranted. However, accounting for these geometric relationships can potentially result in further improvement.

Multi-Robot System Part IV

13 Introduction

Multi-robot cooperation and information sharing can bring potential benefits, such as improved localization and faster task completion, despite the challenges posed by communication limitations. The subsequent chapters in this part introduce the algorithms we have developed for cooperation among underwater robots. The robots share information and obtain relative measurements using acoustic transceivers.

The increasing logistical complexity in performing outdoor experiments with multiple robots, coupled with damages and production challenges forced us to shift towards simulation. We used the experience we gathered during outdoor experiments in the previous part to recreate relevant aspects of real-world experiments in simulation. Further, we replaced the acoustic transceivers we developed and presented earlier in Chapter 10 with a commercially available product, which is used in Chapter 15. A significant drawback in doing so was that the commercial system obtained range measurements from round-trip travel time in a query-response cycle, unlike the one-way travel-time approach in the system we developed.

13.1 Outline

This part starts with **Chapter 14**, which introduces our simulation setup using the Webots simulation software [42]. Since we use the simulation to evaluate high-level coordination algorithms and not low-level control, we argue that we do not need an accurate simulation of the vehicle dynamics. Instead, we calibrate the inertial measurement errors and actuator errors in simulation so that the errors in inertial position estimates in simulation are similar to those shown by real robots. Further, we characterize errors in acoustic ranging and communication with outdoor experiments and reproduce those errors in simulation.

In **Chapter 15**, we present a method for cooperative localization based on covariance intersection with pairwise range-only relative measurements. Given that range measurements are only one-dimensional, covariance intersection is performed in a transformed space along the estimated relative position vector of the robots. We also present a heuristic for choosing a peer robot for a range measurement by maximizing mutual information. We evaluated our method for accuracy and estimation consistency, and show results from simulations as well as outdoor experiments. This chapter is based on the following publication:

1. A. Quraishi and A. Martinoli, "Distributed Cooperative Localization with Efficient Pairwise Range Measurements", in *International Symposium on Distributed Autonomous Robotic Systems, 2021*, Springer Proceedings in Advanced Robotics, 2021. To appear.

Subsequently, in **Chapter 16**, we extend our work in Chapter 11 on acoustic beacons and exploit the fact that they are mobile. We develop an approach for planning their trajectory in a way that they provide the best possible navigation support for a group of underwater vehicles. We use an information-theoretic approach to beacon path planning that minimizes the group's position uncertainty. We evaluate our approach with realistic simulations and present results. This chapter is based on the following publication:

2. A. Quraishi and A. Martinoli, "Coordinated Path Planning for Surface Acoustic Beacons for Supporting Underwater Localization", in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2021. To appear.

Finally, in **Chapter 17**, the closing chapter, we shift gears towards the multi-robot underwater sampling task. We start with a review of various approaches for the same. We then implement one such existing approach in simulation and study the effect of communication and localization errors on performance.

13.2 Related Work

This section reviews the existing work related to our contributions covered in this part of the manuscript. Note that in Chapter 17, we implemented an existing approach for multi-robot adaptive sampling. Several existing approaches for the same are compared within that chapter.

13.2.1 Underwater Robotics Simulation

Accurate simulation of aquatic environments is difficult, in large part because modeling physical underwater phenomena itself is challenging. Nonetheless, a number of robotics simulators targeted towards underwater environments exist, such as the ones presented in [83]–[86]. While they do usually simulate hydrodynamic forces, they often only have a very basic water-current simulation (such as with a constant velocity field and laminar flow). A separate set of tools [87], [88] aim to simulate underwater acoustic signal propagation. In [89], the authors simulate acoustics within an underwater robotics simulator. However, no calibration to reality or practical comparison with real-world behavior of acoustic systems is presented.

In addition to simulation features, the choice of a simulator software is also driven by the suitability for a specific application and ease of use. The simulated behavior of various peripherals (in particular, acoustic transceivers) may not be representative of the actual hardware employed by a certain user. As a result, we decided to use a robotics simulator with a basic simulation of hydrodynamic forces. To it, we separately plugged-in real-world error characteristics of acoustic transceivers and inertial measurements specific to our hardware.

13.2.2 Cooperative Localization and Peer-Selection

In Cooperative Localization (CL), a team of robots shares and fuses information and relative measurements to improve individual-robot localization accuracy. Such information sharing makes position estimates of all robots correlated [90]. Therefore, relative measurement updates in CL must be handled carefully to avoid inconsistent and overconfident position estimates.

The simplest approach for CL is to gather all robot observations and relative measurements and process them at a central location [91]. Roumeliotis et al. in [92] showed that a centralized Kalman filter for CL can be decomposed into smaller, communicating filters which are distributed among the robots. Later, Luft et al. in [93] limited communication exclusively between the pair of robots that obtained a relative measurement. This reduced the communication cost for N robots from $\mathcal{O}(N)$ to $\mathcal{O}(1)$. However, both approaches require inter-robot cross-correlation terms to be communicated along with robot position estimates.

Treating correlated information as independent, as in [94], can make new position estimates overconfident [95]. In the work of Bahr et al. [90], each robot maintains a set of several state estimates, and keeps track of their dependencies with other peers through careful book-keeping. A robot can use only those estimates of another peer that are not correlated (directly or via other peers) with its own estimate. However, the memory, computation, and communication requirements of this approach grow exponentially with the number of robots. In [96] and [97], the authors use an approach based on distributed Maximum Likelihood Estimation, where each robot optimizes its own state given the relative measurements. The problem of inter-robot correlations does not arise in this approach.

Another technique for addressing the problem of inter-robot correlation is Covariance Intersection (CI) [95] [98] [99] [100], which treats estimates as if they were maximally correlated. They do not require robots to communicate cross-correlation estimates, saving communication bandwidth. The price to pay is that they are pessimistic in that they overestimate uncertainty. This overestimation is addressed in a hybrid approach, called Split-CI [101], [102]. It splits the covariances into dependent and independent components. However, it requires communication of both, the independent and the dependent covariance matrices.

A related problem when performing pairwise relative measurements is that of optimal peer-selection, which is similar to optimal sensor or beacon selection. In [103], three best (fixed) ultrasonic ranging beacons are selected based on Geometric Dilution of Precision (GDOP) for indoor localization. An entropy-minimization-based sensor selection approach for fixed target tracking is presented in [104]. In [105], a selection criterion based on mutual information for general Bayesian filtering problems is presented.

In our work, we used covariance intersection keeping in mind the communication limitations with underwater robots. We then derive a ‘usefulness’ expression to choose a peer with which to perform a relative range measurement.

13.2.3 Path Planning for Mobile Beacons

Much of the existing work in underwater acoustic localization utilizes beacons deployed at a known location, usually on the surface [50], [51]. Guo et. al. in [106] perform localization considering that an anchored beacon can sway. However, beacon mobility is not actively controlled. In [55], the authors recognize that fixed beacons limit the operational area of the robots. This is because the ranging beacons have a limited range, and the accuracy of relative measurements (ranging or bearing) often decreases with increasing distance. Further, some relative position configurations (such as collinearity between beacons and an AUV) result in degraded localization accuracy.

In [107], AUVs equipped with a multi-receiver array measure range and bearing to a beacon. They use this to localize themselves and also maintain a fixed relative position with respect to the beacon. Then, the AUVs move in response to the motion of the beacon, effectively sidestepping the problem of limitation in operational area. However, it is the beacon that influences the AUV motion in a centralized fashion, not the other way around.

In contrast, we seek to develop a method for surface acoustic beacons to move automatically in response to the localization needs of AUVs, in a way that the drawbacks outlined earlier are avoided. In a similar vein, several approaches have previously been developed for localizing fixed sensor nodes at unknown locations in a Wireless Sensor Network (WSN). Erdemir et.al. [108] proposed a method where a mobile beacon traverses an optimal path that provides sufficient range measurements to the fixed nodes for localization by trilateration. Note that here the beacon is mobile, while the localization targets – the sensing nodes – are static at unknown positions. Other approaches to the same problem have been presented in [109]–[111]. The distinction in our robotics problem is that both the beacon and the localization targets – the AUVs – are mobile, and the positions of the AUVs are not completely unknown.

14 Multi-Robot Simulation

SIMULATIONS are essential for evaluating and testing newly developed algorithms under a variety of conditions. This is especially true for multi-robot systems, given the effort and complexity involved in performing outdoor multi-robot operations. We used Webots [42], a high-fidelity, physics-based robotics simulation software for simulating multi-robot experiments.

Note that Webots was not specifically designed for simulating underwater environments, and provides a rather basic simulation of robots in an aquatic medium (buoyancy, drag forces, motor thrust, etc.). It also does not simulate acoustic transceivers and biological environmental phenomena. However, note also that the dynamics of individual robots were calibrated and tested (along with other developed subsystems) in outdoor environments in the earlier chapters. Therefore, accurately simulating underwater robot dynamics is not our goal. We are primarily interested in simulating acoustic communication and range measurements, high-level control, and multi-robot coordination algorithms. To that end, Webots simulates communication and interaction between robots. To augment that, we used data from outdoor experiments in the previous part of this thesis to calibrate in simulation the inertial positioning errors and errors in acoustic ranging and communication. In this chapter, we describe our simulation setup that is used in the subsequent chapters.

14.1 Basic Elements

Webots provides elements for constructing solid objects in aquatic environments. The density of the elements and viscosity for a fluid medium can be specified. It internally simulates buoyancy, viscous drag forces under laminar conditions, and propeller thrust. The user-provided software can control actuators (thrusters in this case) and obtain measurements from the usual simulated sensors (IMU, GNSS receiver, etc.). For each simulated AUV, we paired the application layer code of the navigation/communication module with a new hardware abstraction layer for accessing the simulated sensors and actuators in Webots.

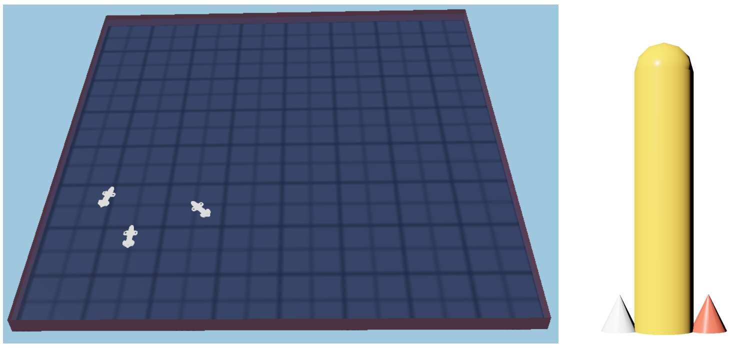


Figure 14.1: Left: Webots simulation environment with three AUVs. Right: The shape of the solid model for dynamics simulation in Webots.

14.1.1 Robot Model

An accurate model of the AUV would be difficult to replicate and calibrate in simulation However, given our objectives for using simulation, an accurate model is not necessary. We used a simplified AUV model with a cylindrical hull and protruding rear thrusters as shown in Fig. 14.1. We used the same external dimensions as the Vertex AUV. We only broadly simulate the external drag forces and differential propeller thrust, i.e, only linear dynamics. We do not simulate rotational dynamics and attitude control to avoid simulation complexity. In essence, the dynamics are only *roughly* calibrated.

We use the actual shape of the AUV for visual purposes, but the dynamics are computed as per the simplified geometry.

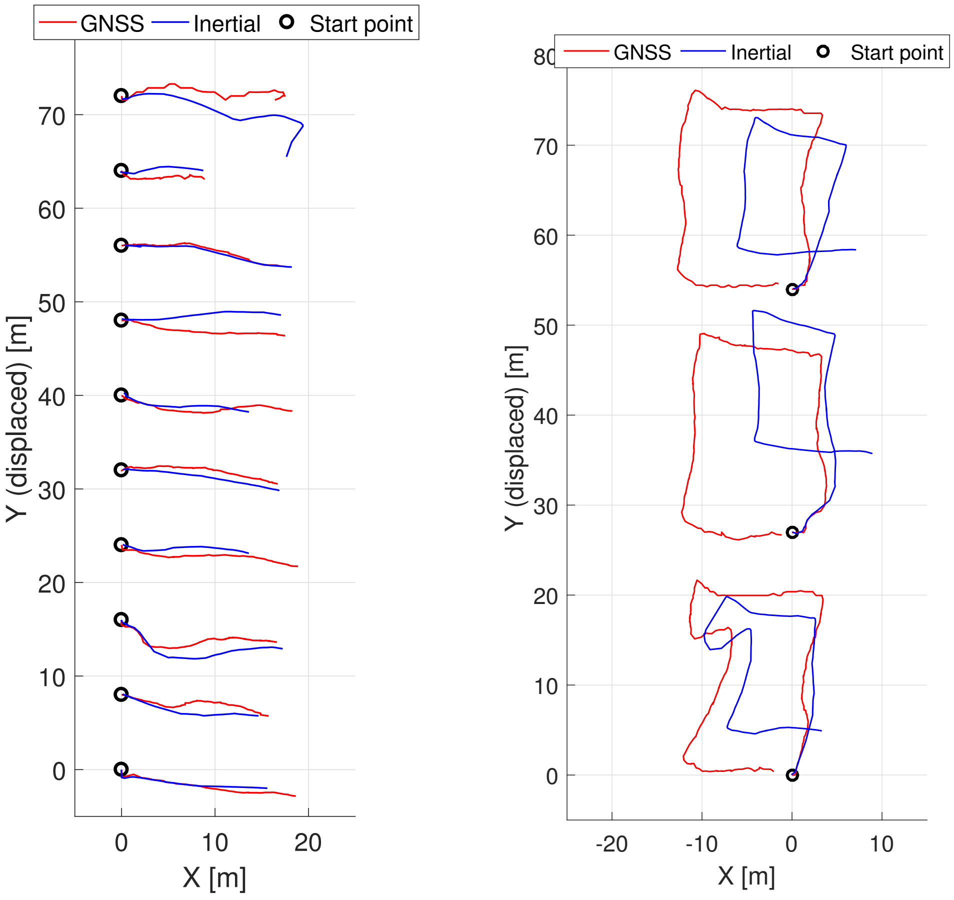
14.1.2 Sensing

Webots simulates the usual robotics sensors and allows specification of noise parameters. We used an IMU and a GNSS receiver. In Webots, the GNSS receiver provides 3D position. We use it to separately simulate a GNSS receiver providing 2D position and a depth sensor for the vertical dimension. The simulation of acoustic ranging is explained in a later section.

14.2 Calibration of Robot Odometry

In order to accurately reproduce the performance of acoustic localization methods, we require that the errors in the inertial position estimates in simulation be similar to those in reality. To achieve that, we run the real robots along simple reference trajectories several times and observe the error in the inertial estimates. We assume that the yaw angle is known (either using a magnetometer or the estimation method in Chapter 12). We then calibrate the simulation to obtain similar errors in the inertial estimates along the same trajectories.

We ran the robots along a straight line and a quadrilateral trajectory several times. The real robot trajectories are shown in Fig. 14.2. Note that local water currents sometimes displace



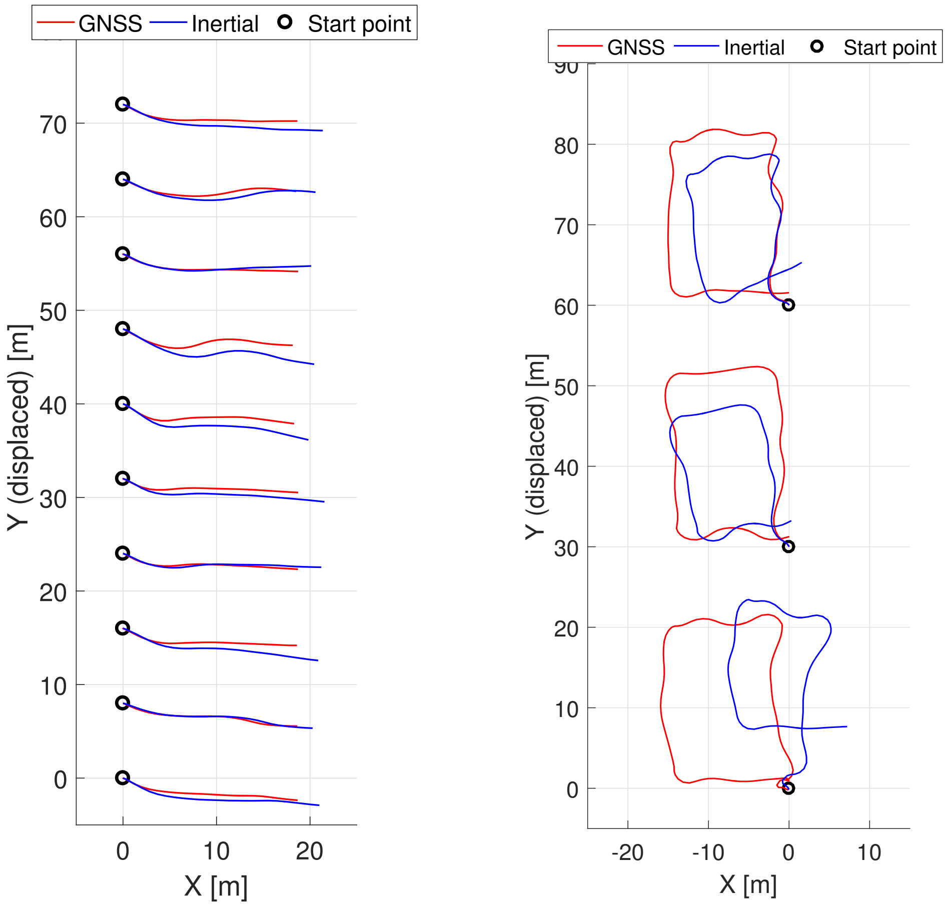
(a) Trajectory 1

(b) Trajectory 2

Figure 14.2: Two different real robot trajectories and corresponding inertial estimates. Note that the starting points of the trajectories are actually coincident, but they are displaced in the figure for clarity.

the robot from its planned path.

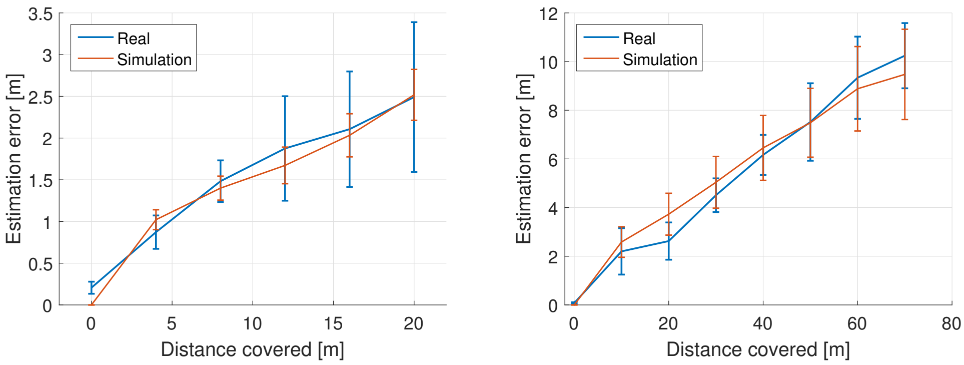
In order to have similar errors in the inertial trajectory in simulation, we tuned the simulated actuator noise and IMU measurement noise. Trajectories of the robot in simulation along with the corresponding inertial estimates are shown in Fig. 14.3. The errors in inertial estimates with real robots are compared with those in simulation in Fig. 14.4.



(a) Trajectory 1

(b) Trajectory 2

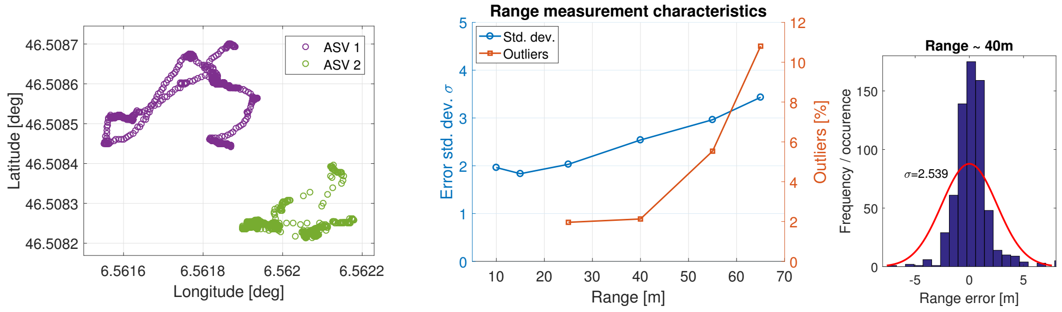
Figure 14.3: Inertial estimation in simulation with two different robot trajectories, after calibration.



(a) Trajectory 1

(b) Trajectory 2

Figure 14.4: Errors in inertial estimation (instantaneous) vs. distance covered by the robots, in simulation as well as with real robots. The errors are averaged over 10 runs in each instance, except for the real runs in Trajectory 2, where 5 runs are used.



(a) ASV locations

(b) Ranging error and outliers

(c) Range error dist.

Figure 14.5: (a) A part of the ASV locations at which range measurements were recorded. (b) Characteristics of error in range measurements, showing standard deviation in the error and occurrence of outliers for various values of range. Outlier occurrence is shown as a percentage of the total number of range measurements performed. (c) Distribution of error in range and bearing measurements at 40 m range, with a Gaussian fit, after outlier rejection.

14.3 Simulating Acoustic Ranging and Communication

We characterized the ranging and communication performance of the DiveNET Microlink acoustic transceivers in order to simulate them accurately. We measured the accuracy of range measurements and gathered statistics on the occurrence of outliers in a shallow water environment. We also gathered statistics on the occurrence of errors in communication. We then simulated acoustic transceivers with the same aggregate characteristics.

14.3.1 Range Measurements

We performed a series of measurements near the shore in Lake Geneva. We mounted the acoustic transceivers on two ASVs, and recorded range measurements as well as GNSS positions by placing both ASVs at a variety of locations. A subset of ASV locations are shown in Fig. 14.5a. Note that due to forces from wind and waves, it is not possible to hold the ASV static. We then measured the accuracy with respect to GNSS and frequency of occurrence of outliers. The GNSS itself has a positioning error of about 1 m. Note that these measurements were performed in a shallow region of the lake, where the depth of the water column ranges between 5-15 m. These conditions are not perfectly ideal for acoustic transceivers, and they are likely to perform better in deeper regions.

We found that with increasing distance, the error in range measurements as well as the occurrence of outliers increases significantly. The accuracy characteristics are shown in Fig. 14.5b. We also show the distribution of range measurement errors after outlier rejection and compare it with a Gaussian fit in 14.5c. We found that the time taken to obtain a range measurement, Δt_{range} is approximately

$$\Delta t_{\text{range}} = 1.25\ \text{s} + 2 \cdot t_{\text{prop}}, \tag{14.1}$$

where t_{prop} is the propagation delay, which depends on the distance between the two robots.

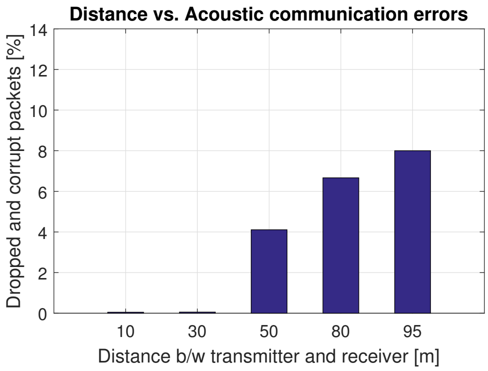


Figure 14.6: Occurrence of acoustic communication errors (sum of corrupt and lost data packets) at varying distances between the transmitter and receiver.

14.3.2 Communication

We used a similar setup as the previous section to characterize communication errors. We mounted the transceivers on ASVs and placed them at varying distances from each other, and transmitted a known sequence of data packets. We then observed the occurrence of corrupt and lost data packets. We found an increase in communication errors with increasing distance between the receiver and transmitter, as shown in Fig. 14.6. The approximate duration between transmitting a data packet and its reception by the other robot, Δt_{comm} approximately is

$$\Delta t_{\text{comm}} = 1.0\,\text{s} + t_{\text{prop}}. \tag{14.2}$$

The data throughput of the acoustic transceivers is 78 bits/s.

14.4 Conclusion

This chapter introduces the setup used for simulating the multi-robot experiments in the following chapters. Our goal is to evaluate new algorithms and methods in a variety of scenarios under conditions that are representative of real-world environments. Therefore, we use data from real-world operation of robots to calibrate some aspects of the simulation in Webots.

We reiterate that the robot dynamics are not calibrated in simulation since we do not need an accurate simulation of the dynamics. We have already developed robot control and localization for single robots in the previous part. Our goal here is to simulate the interaction between robots, cooperative localization, and coordination.

15 Cooperative Localization

Cooperative localization is a paradigm where a team of robots operating together share information and relative measurements to improve their individual localization accuracy. In the underwater domain, where external positioning references are limited, cooperative localization allows new, more accurate position information acquired by one robot to be propagated to other robots. We have previously demonstrated acoustic localization for underwater robots with static and moving surface beacons. However, by using CL and taking turns to resurface periodically for GNSS reception, underwater robots can function without relying on external beacons, as illustrated in Fig. 15.1.

This chapter introduces a method for cooperative localization based on covariance intersection with pair-wise, range-only relative measurements in underwater robots. Then, it addresses the problem of optimal peer selection, i.e., for a given robot, determining which other robot provides the most informative range measurement. We have developed our method keeping in mind the limitations of underwater acoustic ranging and communication. We evaluated our method for accuracy and consistency using simulations as well as outdoor experiments.

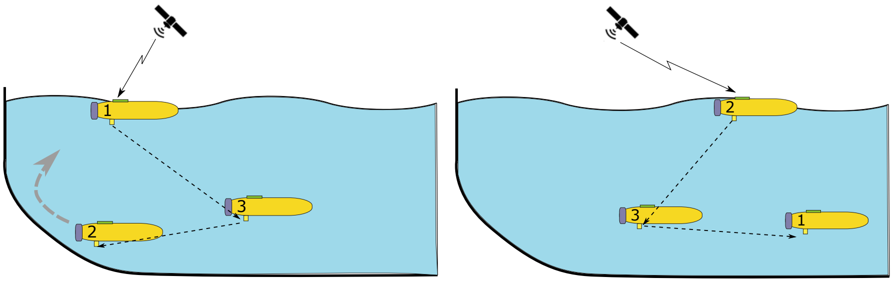


Figure 15.1: Underwater robots periodically resurface for GNSS position reception, and then use CL to share the improved position estimate with other robots.

15.1 Motivation and Background

Given the communication limitations and restricted access to the acoustic channel in underwater robotics, covariance intersection (CI) offers important advantages. It assumes maximal correlation between position estimates of robots in order to void overconfidence in fused estimates, side-stepping an important problem in CL as we saw before (see Section 13.2.2). Consequently, it reduces communication overhead by not requiring inter-robot correlations to be communicated. Then, a relative measurement update requires information only from the two robots involved in it. While this is inefficient in that the uncertainty in robot positions is *overestimated*, it allows for a completely distributed implementation of CL. Additionally, CI is provably consistent [99].

When a relative measurement comprises of a full relative pose along with the position estimate of another robot, it is straightforward to perform CI. However, range measurements are one-dimensional (while robot positions are two- or three-dimensional). They can have a direct influence on the robot position only along the relative position vector between the two robots. Therefore, we perform CI in a transformed space aligned with an estimate of this vector to update the robot position and the corresponding position covariance. It is important to note that range measurements can indirectly influence all state variables via the cross-correlations between them. Our method accounts for and preserves the cross-correlation between the state variables. While our method adds computational cost, the cost of internal computation is much lower than that of acoustic communication.

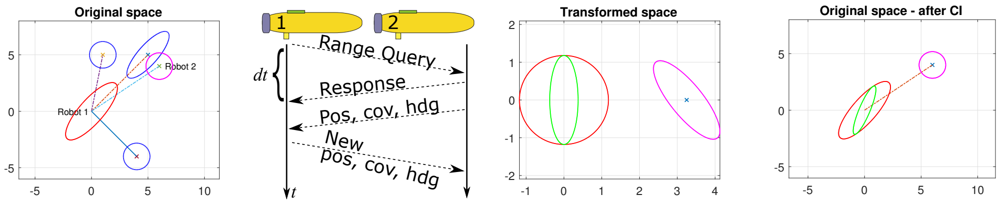
We also derive a peer-selection heuristic for choosing the most informative peer for performing a pairwise range measurement. In the trivial case when uncertainty in positions of other robots is not known, the best choice is a peer robot that is along (or closest to) the direction of highest uncertainty. However, the knowledge of the said uncertainty exists because robots broadcast their position estimates during a range measurement. We use the mutual information between the current position estimate of the robot and a potential range measurement to derive a mathematical expression for scoring the ‘usefulness’ of peer robots.

In summary, this work consists of two main contributions:

1. We derive a linear transform of the robot state in which to perform CI with range-only relative measurements while preserving the cross-covariances in the robot state, and
2. We derive a mathematical expression to rank peer robots in the order of the amount of information a future potential relative update would provide.

15.2 Covariance Intersection for Cooperative Localization

We consider a team of N underwater robots navigating in a three-dimensional space. Robots choose a peer with which to perform a range measurement and transmit a ranging query. On receiving a response, CI is performed using our *projected covariance intersection* method,



(a) Peer selection (b) Query (c) CI (d) New estimate

Figure 15.2: (a) Robot 1 chooses Robot 2 for a range measurement. (b) R2 responds to the range query of R1 and also sends its position estimate. The robots exchange their 3D position, 2D covariance (3 elements of the covariance matrix) and heading. (c) CI is performed in a transformed space. (d) Updated estimate of R1 (in green).

while preserving all the cross-correlation terms in the state. The sequence of steps is illustrated in Fig. 15.2, and explained in the rest of this section.

15.2.1 State Description

The states of the robots are assumed to be Gaussian random variables and are expressed using the mean-covariance parametrization, as introduced earlier in Chapter 8.1.

$$s_i^t \sim \mathcal{N}(\mathbf{X}_i^t, \Sigma_i^t), \tag{15.1}$$

where $i \in \{1, \ldots, N\}$. Our algorithm is agnostic to the formulation of the state variable, except that it requires the three-dimensional position represented in a fixed frame to be a part of the state.

$$\mathbf{X}_i^t = [x_i, y_i, z_i, \ldots] \tag{15.2}$$

15.2.2 Motion and Individual Measurements

Motion and private measurement updates are purely internal to a robot do not require any communication between them. They are processed individually by robots using any kind of sensor fusion framework such as a Kalman filter (see Chapter 8.1).

15.2.3 Range-Based Covariance Intersection

Communication between two robots is required only when they perform a range measurement. No other robots are required to be involved in the communication. However, we assume all communication is broadcast, so other robots can listen to this communication. With the view to minimize data exchange, only robot position, heading and covariance are transmitted. This information is then used to update the position estimate using covariance intersection.

The problem is that one-dimensional range measurement between two robots combined with 2D or 3D position estimate of one robot is not sufficient to compute the position of the other robot. It is then not possible to perform covariance intersection directly. In our method, the state of the robot is transformed into a space where the range measurement is associated with one and only one dimension of the transformed state. 1D Covariance intersection is performed in this space. A final updated state is obtained with an inverse transformation of the space. Note that given that the positions of the robots are estimated and not truly known, the direction of the range measurement is a heuristic.

Consider a robot, i with position and position covariance $\vec{p}_i, \Sigma_{pi}$. It chooses to query another robot j and obtains a range measurement r_{ij}, which has an error with standard deviation σ_r. Robot j also transmits its position estimate $\vec{p}_j, \Sigma_{pj}$. We define the range vector as

$$\vec{r}_{ij} = r_{ij} \frac{\vec{p}_j - \vec{p}_i}{\|\vec{p}_j - \vec{p}_i\|}, \tag{15.3}$$

Note that only the magnitude of $\vec{r}_{ij}$, i.e., r_{ij}, is measured, the actual vector is obtained from estimated quantities.

A transform $\mathscr{F}$ is applied to the state of robot i so that the dimension along $\vec{r}_{ij}$ is decoupled from the rest of the elements of the state.

$$q = \mathscr{F}(s_i - \mathbf{X}_i), \tag{15.4}$$

$$\Sigma'_q = \mathscr{F}\Sigma_i\mathscr{F}^{\mathrm{T}}, \tag{15.5}$$

where q is the transformed state vector. Such a transform can be obtained by setting

$$\mathscr{F} = VW^{-\frac{1}{2}}T^{\mathrm{T}}, \tag{15.6}$$

where T and W are obtained from the eigenvalue decomposition of the state covariance matrix Σ_i. V is obtained from the Gram-Schmidt orthogonalization [112] starting with the vector $\vec{r}_{ij}$. The key feature of this transform is that q is zero-mean and the covariance matrix is identity, i.e., all the elements are uncorrelated.

$$q \sim \mathcal{N}(0_{M\times 1}, \mathrm{I}_{M\times M}). \tag{15.7}$$

This method has similarly been used in [113] for truncating Gaussian PDFs given a hard constraint. The correlations between elements of $\mathbf{X}_i$ are not lost but encoded in T and W. The matrix V in the transform rotates the space in a way that the first dimension in the transformed space is along the range vector. CI is then easily performed in one dimension, and only applies to the first element of q, q_x.

$$q_x \sim \mathcal{N}(0,1). \tag{15.8}$$

Next, the transformation is applied to the other relevant quantities, namely $\vec{r}_{ij}$, σ_r, $\vec{p}_j$ and

Σ_{pj}. Note that $\mathscr{F}$ applies rotation as well as scaling.

$$\vec{r}' = \mathscr{F}\vec{r}_{ij}, \tag{15.9}$$

$$\sigma'_r = \|\mathscr{F}\hat{r}\|\sigma_r, \tag{15.10}$$

$$\vec{p}'_j = \mathscr{F}\vec{p}_j, \tag{15.11}$$

$$\Sigma'_{pj} = \mathscr{F}\Sigma_{pj}, \tag{15.12}$$

where $\hat{r}$ is the unit range vector. Note that the transformed range vector $\vec{r}'$ will be along the 'x' axis of the transformed space. To perform CI, we are interested in the conditional probability distribution of $\vec{p}'_j$ along the transformed range vector $\vec{r}'$ (i.e., the 'x' axis) as below. This is trivial to obtain for a Gaussian distribution.

$$P_j(p'_j | y' = 0, z' = 0) \sim \mathscr{N}(x'_j, \sigma'^2_j). \tag{15.13}$$

In the transformed space, using the range measurement and position of robot j, an estimate of the position of robot i is calculated as

$$\hat{x}'_i = x'_j - \vec{r}', \tag{15.14}$$

$$\hat{\sigma}'^2_i = \sigma'^2_j + \sigma'^2_r. \tag{15.15}$$

Thus, we have

$$\hat{q}_x \sim \mathscr{N}(\hat{x}'_i, \hat{\sigma}'^2_i). \tag{15.16}$$

CI is now performed between q_x and $\hat{q}_x$ (see Eqs. (15.8), (15.16)) to obtain an updated estimate of the position of robot i in the transformed space.

$$\left[\sigma'^2_{x'}\right]^{-1} = \omega\,[1]^{-1} + (1-\omega)\left[\hat{\sigma}'^2_i\right]^{-1}, \tag{15.17}$$

$$\mu_{x'} = \left[\sigma'^2_{x'}\right] \cdot \left[0 \cdot \omega\,[1]^{-1} + \hat{x}'_1 \cdot (1-\omega)\left[\hat{\sigma}'^2_i\right]^{-1}\right] \tag{15.18}$$

where ω is chosen to minimize $\sigma'^2_{x'}$. After this update, the resulting state and covariance of robot i in the transformed space are

$$q = \left[\mu_{x'}, 0, 0, \ldots\right], \tag{15.19}$$

$$\Sigma'_q = \operatorname{diag}\left(\sigma^2_{x'}, 1, 1, \ldots\right). \tag{15.20}$$

This is illustrated in Fig. 15.2c. Finally, the new estimate of the state of robot i in the original space is computed by performing an inverse transform.

$$\hat{\mathbf{X}}_{i|j} = TW^{\frac{1}{2}}V^{\mathrm{T}}q + \mathbf{X}_i, \tag{15.21}$$

$$\hat{\Sigma}_{i|j} = TW^{\frac{1}{2}}V^{\mathrm{T}}\Sigma'_q V W^{\frac{1}{2}}T^{\mathrm{T}}, \tag{15.22}$$

which follows from the inverse of the transform, $\mathscr{F}^{-1} = TW^{\frac{1}{2}}V^{\mathrm{T}}$, given that T and V are orthonormal matrices (see Fig. 15.2d).

15.2.4 Range Queries and Communication

The limitation of pairwise measurements is because of constraints of most underwater acoustic transceivers. To avoid interference, only one robot can transmit a signal at a time. However, all robots can listen to any transmitted signal, if they are within communication range. We use a Time Division Multiple Access (TDMA) scheme, where each robot is assigned predetermined time slots by rotation. Implementation of such a scheme has been discussed earlier in Chapter 10.2.2.

During its assigned slot, a robot initiates a range query to one of the other robots. The subsequent exchange is shown in Fig. 15.2b. Range is computed by measuring two-way-travel-time. Finally, the robot performs CI and broadcasts its updated position estimate. The duration of individual time slots depends on the time it takes to perform the exchange in Fig. 15.2b, which is a characteristic of the acoustic transceiver hardware used.

15.3 Peer Selection

Consider a robot i that needs to choose another robot $j \in [1, n] \setminus i$ and perform a pairwise range update (we performed experiments with $n = 3$ and $n = 4$). We seek to choose a robot j that results in the lowest posterior uncertainty, as shown in Fig. 15.2a. Let $\vec{p}_k$ and Σ_{pk} be the *current* position and position covariance of robot k. Let s_k be the random variable representing this position estimate. Then,

$$s_k \sim \mathcal{N}(\vec{p}_k, \Sigma_{pk}). \tag{15.23}$$

Let z_{ij} be the random variable describing the potential range measurement between i and j (including the position of a peer robot j). We would like to choose a j for which the mutual information $I(s_i; z_{ij})$ is maximized. Formally, we seek to solve the problem

$$j^* = \underset{j \in [1,n] \setminus i}{\operatorname{argmax}} \; I(s_i; z_{ij}). \tag{15.24}$$

We know that this mutual information between two random variables can be written in terms of their differential entropy as

$$I(s_i; z_{ij}) = h(s_i) - h(s_i | z_{ij}), \tag{15.25}$$

where $h(s)$ is the differential entropy of random variable s [114]. When s has a Gaussian distribution, $h(s)$ can be evaluated as

$$h(s) = \frac{1}{2} \log\left((2\pi e)^n |\Sigma_s|\right) \text{ bits}, \tag{15.26}$$

where $|\Sigma_s|$ is the determinant of the covariance matrix Σ_s. In Eq. 15.25, The probability density $P(s_i)$ is fixed, and only the posterior density $P(s_i|z_j)$ depends on the choice of j. Therefore, we can combine Eqs. (15.25), (15.26) to reformulate the problem in Eq. 15.24 as

$$j^* = \underset{j\in[1,n]\setminus i}{\arg\min} \; |\hat{\Sigma}_{i|j}|, \tag{15.27}$$

where $\hat{\Sigma}_{i|j}$ is the covariance of the posterior probability density. This was previously computed in Eq. 15.22. On the right side of the equation, note that T and V are orthonormal matrices, hence their determinant is 1. The posterior covariance in the transformed space Σ'_q is shown in Eq. 15.20 to be a diagonal matrix such that its determinant will evaluate to $\sigma^2_{x'}$. Therefore, we have

$$|\hat{\Sigma}_{i|j}| = |W||\Sigma'_q| = |W|\sigma^2_{x'}. \tag{15.28}$$

Considering that that W is independent of j, and following from Eqs. (15.15) and (15.17), the problem can be further simplified to

$$j^* = \underset{j\in[1,n]\setminus i}{\arg\min} \; \sigma'^2_j + \sigma'^2_{r,j}. \tag{15.29}$$

We recall here that the first term is the conditional distribution of the position of robot j, and the second term is the variance of the range measurement, both in the transformed space. Note that it can be easily deduced that if the first term is ignored, the minimum is obtained for robot j which is along the direction of the highest uncertainty of robot i.

15.3.1 Predicting Peer Positions

Solving the optimization problem in Eq. 15.29 would require the knowledge of current position estimates of all robots, which is not feasible in view of the communication constraints. However, since all communication is broadcast, all robots receive position estimates and heading of other robots during their recent range measurement (see Fig. 15.2b and Section 15.2.4). This information is used to compute the predicted position (and covariance) of other robots, which serves as a heuristic for solving the optimization problem.

For this prediction, the robots employ the process model presented earlier in Chapter 8.1.5. To recap, we have

$$\hat{\mathbf{X}}^{t+dt} = f\left(\mathbf{X}^t, dt\right), \tag{15.30}$$

$$\hat{\Sigma}^{t+dt} = F\Sigma^t F^{\mathrm{T}} + \mathscr{R}. \tag{15.31}$$

The robots assume a constant speed of 1 m/s of their peers (which is the programmed speed of the robots) and use the heading angle to compute the velocity vector. Further, they assume that the control inputs are such that the actuation cancels out the drag forces. In other words, a zero acceleration is assumed. Note also that $\mathscr{R}$ is a constant matrix. This is sufficient to calculate the predicted position $\vec{p}_k$ and uncertainty in position Σ_{pk} for each peer robot k.

15.4 Simulation and Field Experiments

The proposed method was implemented and tested in a group of three to four robots. We evaluated our algorithms in terms of localization accuracy and estimation consistency, both in simulation as well as with outdoor experiments. In both, simulation and real experiments, the robots were programmed to follow a pre-planned trajectory. To emulate periodic surfacing events in a team of underwater robots, two of the three or four robots in the group were allowed periodic access to GNSS positioning information, which would be passed on to other robots through the cooperative localization framework. Otherwise, GNSS positions were used only for following the trajectory and as ground truth.

Experiments are performed with two strategies of choosing a peer for range measurement. In the first one, called *cyclic*, each robot queries other robots for a range measurement turn-by-turn, in a cycle. The second one, called best-selection or *bsel*, uses the proposed peer-selection approach.

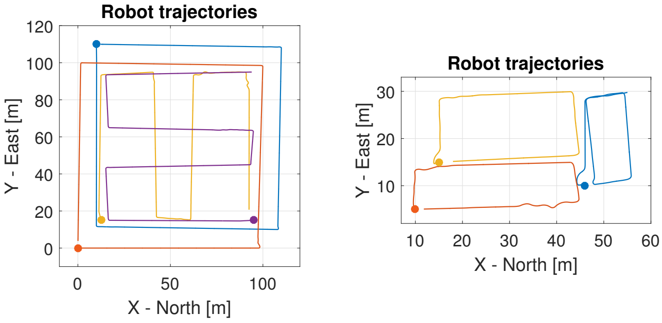
Range measurement updates are performed using the proposed CI approach, as well as the EKF approach. The EKF approach uses the standard update equations, ignoring the correlation between inter-robot position estimates. The robots use proprioceptive sensors and the motion model for inertial navigation, combined with range measurement updates either using EKF or CI. We do not implement any centralized EKF for joint estimation of positions of all robots. Each robot runs an independent state estimator.

15.4.1 Setup

Outdoor experiments were performed using three ASVs equipped with DiveNET Microlink acoustic transceivers. A range measurement, including related data exchange (Fig. 15.2b), takes about 3.5 s. Experiments were performed in Lake Geneva. The robot trajectories are occasionally disturbed by strong waves. Robot 1 has no access to GNSS for position estimation. Robots 2 and 3 are allowed GNSS updates once in 20 s (0.05 Hz). Acoustic range is recorded every 1.5 s between each of the three pairs of robots. For the purpose of this experiment, shorter acoustic time-slots were used without any guard times for avoiding echoes and interference, and only range measurements were recorded. Position information was recorded via radio.

The recorded data are re-processed offline with different range update methods (CI, EKF) and peer-selection strategies (cyclic, bsel). Depending on the peer-selection strategy, and subject to a more realistic time-slot length of 5 s, only a subset of the recorded range measurements are used. This gave us accuracy and consistency metrics for both range measurement update methods as well as peer-selection strategies.

In simulation, we restrict access to GNSS to two robots at 0.1 Hz or 0.05 Hz (once in 10 s or 20 s). The time-slot length for acoustic transmission is set to 5 s, which means there is one range measurement in the whole system every 5 s. We perform five experimental runs for each



(a) T1, 4 robots

(b) T2, 3 robots

Figure 15.3: Two sets of trajectories, T1 and T2, in simulation. The starting point of the trajectory for each robot is shown with a dot.

combination of trajectory, GNSS access, peer-selection method (cyclic or bsel), and range update method (EKF or CI).

15.4.2 Evaluation Metrics

We use the RMSE as a measure of accuracy. We compute it as the Euclidean distance between the true and estimated positions averaged over all time steps since the beginning of the experiment.

$$\mathrm{RMSE(T)} = \frac{1}{T}\sum_{t=1}^{T} \|\vec{p}^{\,t} - \hat{p}^{t}\|, \tag{15.32}$$

where T is the total number of evaluated time steps. RMSE is higher when the estimated trajectory is farther off from the true trajectory.

For measuring consistency, we use the Normalized Estimation Error Squared (NEES), averaged over all time steps since the beginning of the experiment.

$$\mathrm{NEES(T)} = \frac{1}{T}\sum_{t=1}^{T} (\vec{p}^{\,t} - \hat{p}^{t})^{\mathrm{T}} (\Sigma_p^t)^{-1} (\vec{p}^{\,t} - \hat{p}^{t}). \tag{15.33}$$

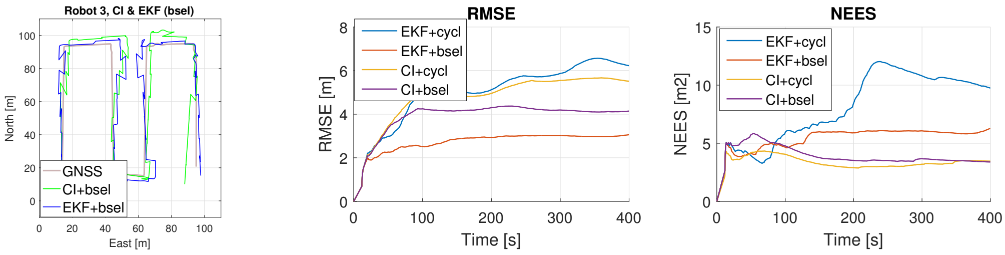
Higher values of NEES indicate a higher inconsistency between the estimated position covariance and the actual error in estimated position. This metric was introduced by Shalom et. al. in [115].

15.4.3 Results

In simulation, experiments are performed with two sets of trajectories, with two and three robots. The trajectories are shown in Fig. 15.3. In both simulation and real experiments, two of the robots are allowed GNSS access at varying rates. The results of various experimental scenarios, averaged over all runs and across all robots are tabulated in Table 15.1.

	CI		EKF	
Scenarios	RMSE-Cyclic NEES-Cyclic	RMSE-bsel NEES-bsel	RMSE-Cyclic NEES-Cyclic	RMSE-bsel NEES-bsel
1. T1,GNSS 0.10Hz,2 Rob/4	5.51 3.45	4.14 3.38	6.22 9.76	3.06 6.27
2. T1,GNSS 0.05Hz,2 Rob/4	6.19 5.43	4.57 4.02	6.77 11.89	3.85 6.10
3. T2,GNSS 0.10Hz,2 Rob/3	2.23 2.71	2.14 3.69	2.34 4.58	2.05 4.31
4. T2,GNSS 0.05Hz,2 Rob/3	2.60 2.74	2.33 3.41	2.88 5.93	1.94 4.17
5. Real,GNSS 0.05Hz,2 Rob/3	3.28 5.03	2.17 4.56	3.23 6.33	2.44 7.22

Table 15.1: RMSE and NEES for various scenarios and estimation methods.



(a) EKF vs CI

(b) Evolution of RMSE

(c) Evolution of NEES

Figure 15.4: For scenario 1: (a) Example estimated trajectory of one of the robots using CI and EKF. (b) Comparison of RMSE over time for the two peer-selection strategies. Our peer-selection method results in higher accuracy. (c) Comparison of NEES for CI and EKF. EKF has higher estimation error owing to ignoring inter-robot correlations.

Simulations

An example of a trajectory of one of the robots estimated with CI and EKF is shown in Fig. 15.4a. A comparison of evolution of RMSE in scenario 1 is shown in Fig. 15.4b. Our peer-selection strategy, bsel, results in improved accuracy of estimated position compared to the cyclic strategy. This is because robots are able to predict which peer can provide the most useful information based on the heuristic in Eq. 15.29.

Evolution of NEES for scenario 1 is compared in Fig. 15.4c. EKF initially performs similar to CI. However, as uncertainties increase over time, they must be accounted for. Therefore, after a few range updates, NEES for EKF increases in comparison with that of CI.

Outdoor Experiments

For outdoor experiments, the estimated trajectory for one of the three robots along with ground-truth trajectories of the other two peers is shown in Fig. 15.5. The purely inertial estimate (without any GNSS or acoustic updates) is also shown for comparison. The evolution of RMSE and NEES is shown in Fig. 15.6a, 15.6b. The accuracy metrics are shown in the last row of Table 15.1.

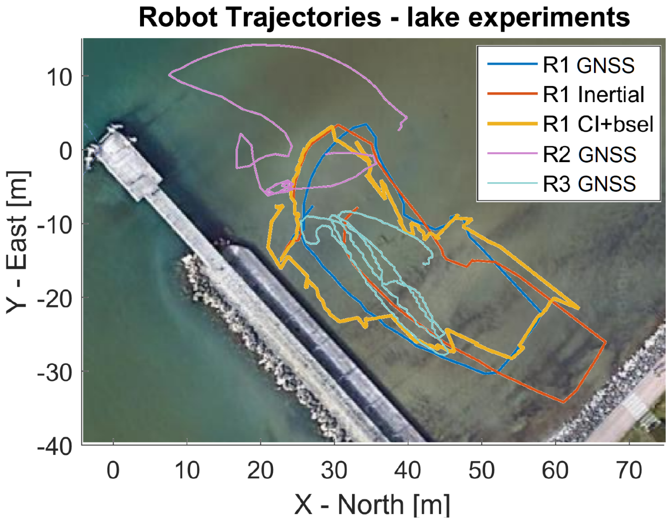
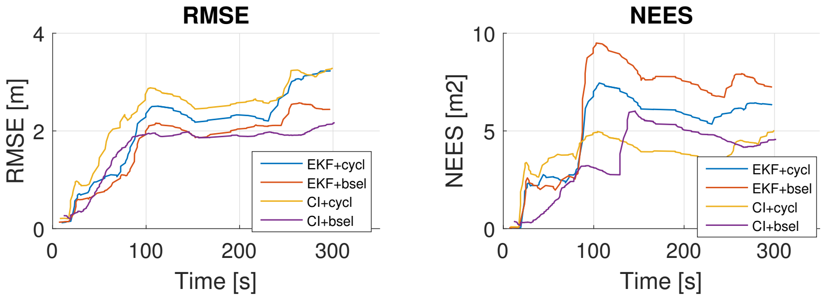


Figure 15.5: Robot trajectories from the real experiment. For Robot 1, trajectory estimated with with CI and the proposed peer-selection strategy is shown. The true (GNSS) trajectory and the inertial estimate (without range measurements) are also shown for comparison. For Robot 2 and Robot 3, only the true (GNSS) trajectory is shown.



(a) RMSE

(b) NEES

Figure 15.6: RMSE and NEES for the real experiment. The plots show that CI results in lower estimation error (NEES), which demonstrates more accurate estimation of uncertainty. They also show that using the proposed peer-selection strategy results in more accurate trajectory estimation.

The results demonstrate that the trajectory estimated with CI is more consistent compared to EKF. This is shown by a lower value of NEES, implying that the estimated uncertainty is in better agreement with the actual estimation error. Correct estimation of uncertainty is important because it is used to adaptively weight the influence of incoming measurements (e.g, via the Kalman gain in EKF). Therefore, erroneous uncertainty estimates are likely to cause higher trajectory RMSE eventually. The results also show that regardless of the range update method (CI or EKF), the proposed peer-selection strategy resulted in a lower RMSE. This shows the improvement in accuracy brought by an educated choice of a peer for a relative measurement. The experimental results with real robots are in agreement with those obtained from simulations.

15.4.4 Computation and Data Overhead

The proposed CI approach is more expensive in computation compared to EKF. A range update with EKF would require a 2 × 2 matrix inversion (assuming position estimation in 2D, the vertical dimension is provided by a depth sensor), regardless of the size of the state variable. The proposed transformation requires eigenvalue decomposition, inversion of a diagonal matrix, and several matrix multiplications, the sizes of which depend on the size of the state. This adds an overhead in computation time (EKF: 0.1 ms, CI: 1.5 ms, approx.). These computations are performed on the computation module (1 Ghz processor, see Section 5.3.1).

For the peer-selection heuristic, the only additional information added to the communication is the robot heading. All the other information is also needed for range updates. Computing this metric also requires many of the operations used for CI-based range update (eigenvalue decomposition, etc.). The computation time was found to be about 1.0 ms per peer on our setup. Since there is one range update in approximately 5 s, this combined computation delay is affordable.

15.5 Conclusion

This chapter presented a range-based covariance intersection method for cooperative localization. CI provides a number of advantages for cooperative localization with underwater robots, which have severe communication constraints. CI does not require communication of inter-robot correlation terms and uses information only from the pair of robots involved in a relative measurement. It can thus be run in a fully distributed fashion. We showed that CI-based cooperative localization results in a better estimation of uncertainty.

We also derived a peer-selection heuristic for performing range measurements based on an information-theoretic approach. We showed that our peer-selection strategy improves localization accuracy in comparison to sequentially querying peers for a range measurement.

During real-world operation, range queries or responses made by a robot may get lost or corrupted. At the moment, we do not try to resend a query or query another peer, resulting in no updates being performed during some time slots. A recovery strategy may reduce this ‘dead time’ with no updates.

The peer-selection heuristic is based on estimating the position of peers using a constant velocity model. When a peer changes direction, this estimate becomes invalid. A better estimate can be made if some information about future planned trajectories of peer robots is communicated in advance.

16 Coordinated Mobile Beacons

Acoustic ranging beacons mounted on ASVs form a flexible system for aiding underwater localization. We showed earlier in Chapter 11 that they are easy to deploy, and could transmit their own positions using acoustic communication. Thus, they provided an acoustic range measurement from a known position without having to be stationary. However, uncoordinated motion of the robots and beacons poses a number of drawbacks. First, with increasing distance, the accuracy of range measurements decreases and the occurrence of outliers increases. Second, certain relative position configurations (such as collinearity between two beacons and a robot) provide range measurements resulting in position updates with high error. This is illustrated in Fig. 16.1.

In this chapter, we present an actively controlled beacon system for providing localization support. Each beacon is mounted on an ASV, whose trajectory is controlled in a way that the aforementioned problems are avoided. Further, they are commanded to move to a position that provides the underwater robots with the highest possible gain in position information.

We tested our approach with realistically simulated experiments. We show that our approach for path planning of mobile beacons improves the localization accuracy in comparison to static beacons as well as mobile beacons moving in a fixed formation.

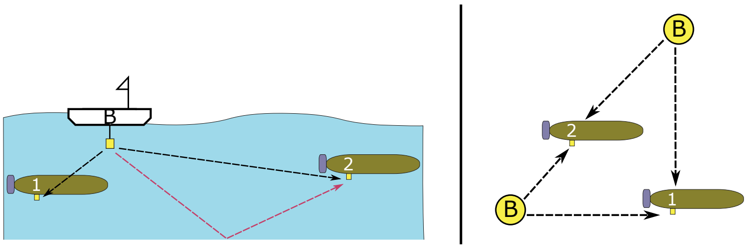


Figure 16.1: Schematic showing effect of beacon placement. Left (vertical plane): At longer distances, reflected signals and multi-path receptions are more difficult to distinguish from true signal receptions, resulting in erroneous range measurements. Right (horizontal plane): When two beacons and a robot are collinear, two range measurements provide information only along one direction, resulting in inaccurate position updates.

16.1 Problem Definition

We consider the deployment of a team of AUVs for any cooperative activity, for example, environmental sampling. The AUVs plan their own path, possibly adapting it during operation. The only assumption we make about AUV trajectories is that they are always within a distance of $d_{\max}$ from each other such that $d_{\max} < r_{\max}$, where $r_{\max}$ is the maximum range of the acoustic transceivers. The beacons (i.e., ASVs) have no prior knowledge of the motion of the AUVs. Our goal is to estimate optimal positions for the beacons in real-time.

To summarize, given a set of beacons and a set of AUVs, as the AUVs travel while carrying out their activity, we require that

1. The beacons move so that they are within a specified distance of $r_{\max}$ from all AUVs,
2. No AUV is collinear with the beacons, and
3. The beacon placement minimizes the posterior uncertainty in AUVs' positions after future potential range measurements.

16.2 Methodology

As before, a TDMA-based schedule is used to assign fixed time-slots of 4 s duration to the AUVs, during which they are allowed to transmit range queries. The AUVs first initiate a range query to one of the acoustic beacons. In the ensuing exchange, they each transmit their position and heading information. This exchange is the same as the one used earlier in cooperative localization (see Fig. 15.2b in Chapter 15). The mobile beacons predict the future position and position covariance of the AUVs using recent broadcasts received from the AUVs. This is then used to compute an optimal placement for the beacons in the next few seconds.

16.2.1 Position Prediction

The beacons use the position estimates and heading broadcast by the AUVs, coupled with a constant velocity assumption, to predict the future position $\hat{x}_i$ and uncertainty in position $\hat{\Sigma}_{x_i}$ for each AUV i. They use the same approach as the one used for peer selection earlier (see Section 15.3.1). Note that since the AUVs can change their direction, this prediction only serves as a heuristic.

16.2.2 Optimal Beacon Placement

We aim to find the beacon location that would provide the best reduction in uncertainty in AUVs' positions after the range update. We do so by minimizing the collective posterior (differential) entropy. Let $\vec{x}_\text{b}$ be the future position of the beacon, and let s_i be the random variable representing the posterior position estimate of AUV i after the future range update.

We seek to solve the following minimization problem:

$$\vec{x}_{\mathrm{b}}^{*}=\underset{\vec{x}_{\mathrm{b}}}{\arg\min}\ h\left(s_1,s_2,\ldots\right), \tag{16.1}$$

where $h(.)$ is the differential entropy of a continuous distribution. Here, we minimize the entropy of the joint distribution of posterior AUV positions.

It is reasonable to assume that the AUV positions are independent since they do not use each others' position estimates. Therefore, the joint entropy reduces to a sum of individual entropies.

$$h\left(s_1,s_2,\ldots\right)=h\left(s_1\right)+h\left(s_2\right)+\ldots. \tag{16.2}$$

Further, given the Gaussian assumption, we have

$$s_i\sim\mathcal{N}\left(\vec{x}_i,\Sigma_{x_i}\right), \tag{16.3}$$

where $\vec{x}_i$ and Σ_i are posterior position and covariance of AUV i. Therefore, for s_i, the differential entropy $h(s_i)$ can be evaluated as [114]

$$h(s_i)=\frac{1}{2}\log\left((2\pi e)^n|\Sigma_{x_i}|\right)\ \text{bits}. \tag{16.4}$$

The posterior covariance Σ_{x_i} can be obtained by predicting the prior position and covariance at a future point in time, $\hat{x}_i,\hat{\Sigma}_{x_i}$ (see Section 16.2.1), and then applying the Kalman update equation in Eq. (8.25) using the future potential range measurement. Following from the range measurement model (Eqs. (8.14), (8.15)) and Eq. (8.23), we obtain

$$\begin{aligned}\Sigma_{x_i} &= \left(\mathbf{I}-KG\right)\hat{\Sigma}_{x_i}, &(16.5)\\ &= \left(\mathbf{I}-\frac{\hat{\Sigma}_{x_i}\left(\hat{x}_i-\vec{x}_{\mathrm{b}}\right)\left(\hat{x}_i-\vec{x}_{\mathrm{b}}\right)^{\mathrm{T}}/g(\hat{x}_i)^2}{\left(\left(\hat{x}_i-\vec{x}_{\mathrm{b}}\right)^{\mathrm{T}}\hat{\Sigma}_{x_i}\left(\hat{x}_i-\vec{x}_{\mathrm{b}}\right)/g(\hat{x}_i)^2\right)+\sigma_r}\right)\hat{\Sigma}_{x_i}, &(16.6)\end{aligned}$$

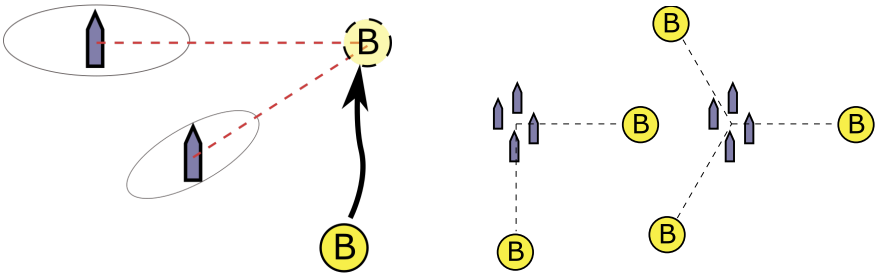
where $g(.)$ is the measurement model for range measurements, G the corresponding Jacobian, K is the Kalman gain, σ_r is the variance of the error in range measurements and $\mathbf{I}$ is an identity matrix of appropriate dimension. The expression in the denominator evaluates to a scalar.

Following from Eqs. (16.2), (16.4), the joint entropy of AUV positions is calculated as

$$\begin{aligned}h\left(s_1,s_2,\ldots\right) &= \sum_i\frac{1}{2}\log\left((2\pi e)^n|\Sigma_{x_i}|\right) &(16.7)\\ &= \frac{1}{2}\log\prod_i\left((2\pi e)^n|\Sigma_{x_i}|\right). &(16.8)\end{aligned}$$

Using this, we can reformulate the optimization problem in Eq. (16.1) as

$$\vec{x}_{\mathrm{b}}^{*}=\underset{\vec{x}_{\mathrm{b}}}{\arg\min}\ \prod_i|\Sigma_{x_i}|. \tag{16.9}$$



(a) Optimal placement (b) Desired beacon distribution

Figure 16.2: (a) Illustration of the optimal placement of the beacon given two AUVs and their position uncertainties. (b) Desired placement of several beacons around a group of AUVs. The ideal placement is a uniform distribution of beacons around the centroid of the group.

In Eq. (16.6), note the dependence of Σ_{x_i} on position of the beacon, $\vec{x}_\mathrm{b}$. This minimization is performed by doing a search on a uniform grid of granularity 1 m around the centroid of the predicted AUV positions.

For a single AUV, $\vec{x}_\mathrm{b}^*$ corresponds to any position along the direction of maximum prior uncertainty in the AUV's position estimate. With a Gaussian distribution, this is the line along the first eigenvector of the prior covariance matrix $\hat{\Sigma}_x$. For two AUVs, $\vec{x}_\mathrm{b}^*$ lies on the intersection of two such lines, one for each AUV. This is illustrated in Fig. 16.2.

16.2.3 Placement of Subsequent Beacons

After range updates to all AUVs from an optimally placed beacon (see Fig. 16.2a), the uncertainties in their positions change, resulting in a new optimal beacon location. Theoretically, one beacon can provide sufficient localization support by repeatedly moving to new optimal locations. However, in practice, each cycle of range updates may result in large changes in the optimal beacon location, and a single beacon may not be able to carry out the amount of motion required. Further, additional beacons will improve localization by immediately providing range information along a complementary direction.

Given that range updates provide position information along only one direction, we ideally require additional range measurements from a diversity of directions. Therefore, the desired placement of subsequent beacons is such that they have a uniform circular distribution around the AUVs, as shown in Fig. 16.2b. In this work, we considered the case of two beacons. We place the second beacon such that the two beacons are in perpendicular directions with respect to the centroid of the group of AUVs.

16.2.4 Master and Slave Beacons

We designate one of the beacons as the *master*, which computes and assigns future placement to itself and the *slave* beacon. Note that decentralized approaches are also possible, but we

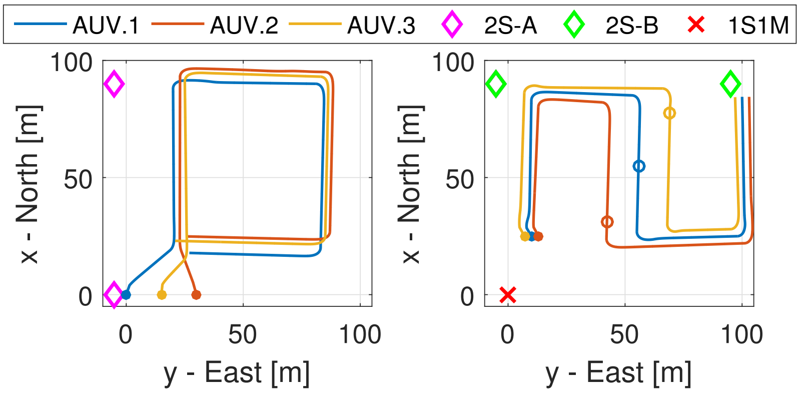


Figure 16.3: Preplanned sets of trajectories T1 (left) and T2 (right). The starting point is shown with a dot. The circles in T2 indicate the positions of the three AUVs at a particular instant of time. The position of static beacons in various configurations is also shown. Experiments with one AUV use only AUV.1 trajectory. Note that we also perform experiments with the combination '2S-A' with T2 and vice-versa.

exploited the fact that the beacons, being on the surface, have access to radio communication.

The master beacon obtains the first desired position using Eq. (16.9) and computes the second such that

- it is in a perpendicular direction with respect to the centroid of the AUV group (see Fig. 16.2b),
- it does not cause collinearity with the other beacon and any of the AUVs to the extent possible, and
- it is within a distance of $r_{\max}$ from each AUV.

Then, the two positions are assigned to the closest ASVs in a way that the total travel distance is minimized.

16.3 Experiments in Simulation

The proposed method was implemented and tested with one and three AUVs and one and two beacons in simulation. We evaluated the algorithm in terms of localization accuracy. We compared the proposed optimal placement approach with a variety of static and moving beacon configurations.

16.3.1 Simulation Scenarios

We simulated various beacon configurations with different AUV operation scenarios. We used a combination of static and mobile beacons. Experiments were performed with a single as well as a group of three AUVs, with two different sets of trajectories. The AUV trajectories were pre-planned (but the beacons had no prior knowledge of them). The trajectories and static

Config.	Description
2SB-A	Two static beacons at position set A
2SB-B	Two static beacons at position set B
1M.O	One mobile beacon with optimal position-based motion
1S1M.O	One static beacon, one mobile beacon with optimal position-based motion
2M.F	Two mobile beacons with formation based motion
2M.O	Two mobile beacons with optimal position-based motion

Table 16.1: Description of various beacon configurations. Positions of static beacons are shown in Fig. 16.3

	RMSE [m]					
Scenarios	2S-A	2S-B	1M.O	1S1M.O	2M.F	2M.O
T1, 1AUV	3.45	3.49	2.18	2.50	2.04	**1.87**
T2, 1AUV	3.81	4.34	2.32	2.65	2.04	**1.94**
T1, 3AUV	4.54	4.78	3.56	3.61	3.10	**2.98**
T2, 3AUV	4.57	4.64	3.09	3.50	2.98	**2.77**

Table 16.2: RMSE of the estimated trajectory in various scenarios. S=static beacon, M=mobile beacon, F=motion in fixed formation, O=motion towards optimal beacon position.

beacon locations are shown in Fig. 16.3. A summary of the various beacon configurations is provided in Table 16.1. In optimal beacon positioning configurations ('xx.O'), the optimal positions were calculated using the strategy described in Section 16.2.4. In the formation configuration ('2M.F'), the two beacons maintained a fixed relative position with respect to the centroid of the AUV group.

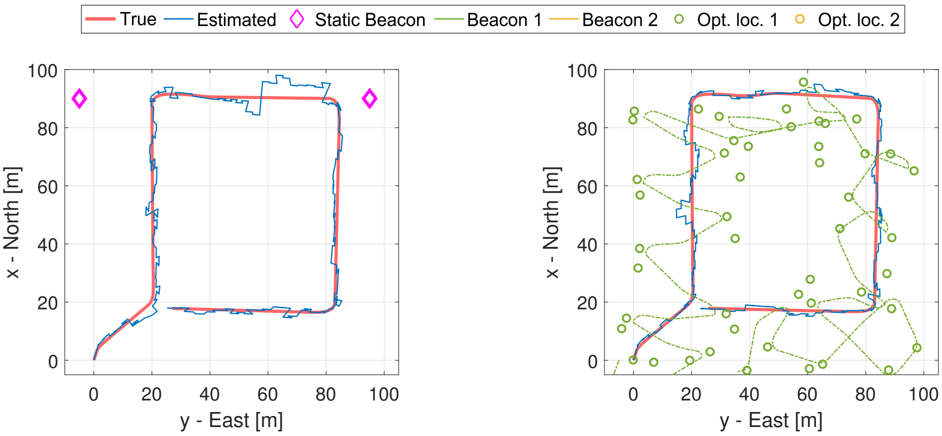
We set the maximum speed of the AUVs to 1 m/s and that of the mobile beacons to 2.5 m/s. The maximum separation between the AUVs, $d_{\max}$ in these trajectories was 25 m. By exploiting motion, we aim to place the beacons such that the maximum range measurement, $r_{\max} < 50$ for low noise (although the rated maximum range of the acoustic transceivers is about 1 km). We compared the accuracy of the estimated trajectory for each combination of AUV operation scenario and beacon configuration.

16.3.2 Evaluation Metric

As before, we use the RMSE as a measure of accuracy. We compute it as the Euclidean distance between the true and estimated positions averaged over all time steps since the beginning of the experiment.

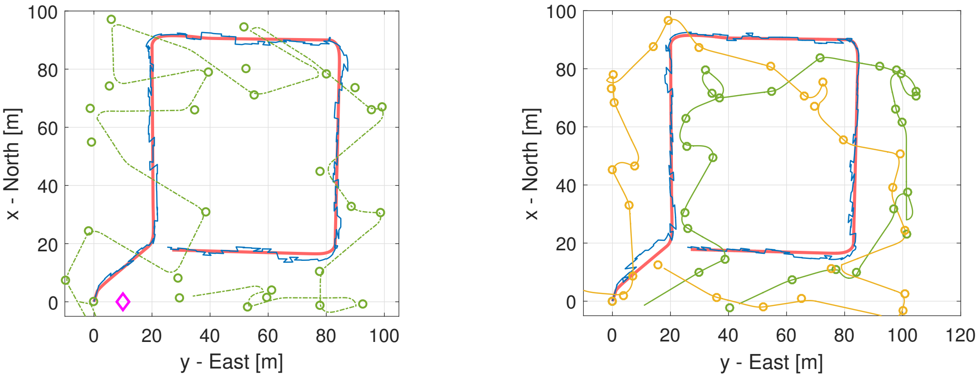
$$\text{RMSE(T)} = \frac{1}{T}\sum_{t=1}^{T} \|\vec{x}^t - \hat{x}^t\|. \tag{16.10}$$

The RMSE of the estimated trajectories in various scenarios averaged over 5 runs are shown in Table 16.2.



(a) Two static beacons (T1, 1AUV, 2S-B)

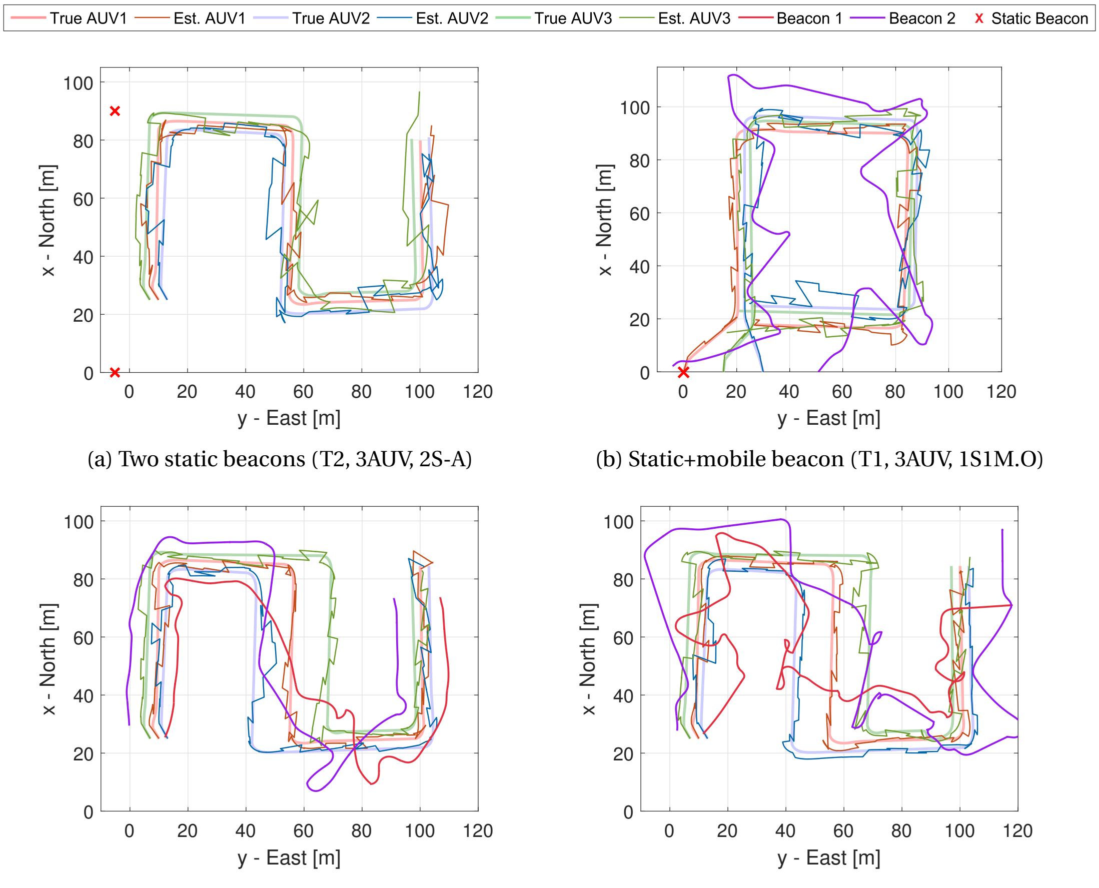
(b) One mobile beacon (T1, 1AUV, 1M.O)



(c) Static+mobile beacon (T1, 1AUV, 1S1M.O)

(d) Two mobile beacons (T1, 1AUV, 2M.O)

Figure 16.4: Trajectories estimated in experiments with one AUV, with various beacon configurations. Locations of the fixed beacons, trajectories of mobile beacons, and their optimal locations are shown. (a) With two static beacons, the collinear section of the trajectory has higher error. (b) A single moving beacon is often unable to reach successive optimal locations, even with a higher speed setting. (c) With one static and one mobile beacon, the distance between successive optimal locations is more reasonable, yet not all locations are reached. (d) With two mobile beacons using the proposed approach, beacons are able to reach the calculated optimal positions. This configuration provides the best localization accuracy.



(a) Two static beacons (T2, 3AUV, 2S-A)

(b) Static+mobile beacon (T1, 3AUV, 1S1M.O)

(c) Beacons in formation (T2, 3AUV, 2M.F)

(d) Two mobile beacons (T2, 3AUV, 2M.O)

Figure 16.5: Trajectories estimated in experiments with three AUVs, with various beacon configurations. Location of fixed beacon and trajectories of mobile beacons are shown. (a) Two fixed beacons. (b) With one static and one mobile beacon, the mobile beacon still often misses the calculated optimal locations. (c) Two mobile beacons moving in formation with respect to the AUV group provide an improvement in the estimated trajectory in comparison with other beacon configurations except for 2M.O. (d) Two mobile beacons using the proposed approach also provide the best results with a group of three AUVs.

16.3.3 Two Static Beacons

An example of a trajectory estimated with two static beacons is shown in Fig. 16.4a (one AUV) and Fig. 16.5a (three AUVs). Static beacons result in decreasing measurement accuracy as the AUVs move further from the beacons. Further, a bad combination of relative positions between the beacons and the AUVs (such as collinearity) results in inaccurate position estimates. This can be seen in the upper segment of the trajectory in Fig. 16.4a. As a result, the two static beacons result in the highest estimated error in most experimental scenarios (see 2S-A and 2S-B in Table 16.2).

16.3.4 One Mobile Beacon

We performed experiments with a single mobile beacon using the proposed approach (1M.O in Table 16.2). To show that successive optimal beacon locations can be diverse and to the stress importance of having multiple beacons, we set the beacon velocity to a hypothetically large value of 4 m/s. We found that the beacon was not able to reach the optimal locations in time, owing to large distances between successive optimal locations, despite the higher velocity. The estimated trajectory of the AUV and that of the mobile beacon in a one AUV experiment is shown in Fig. 16.4b. A subset of the optimal locations obtained by solving Eq. (16.9) are also indicated, showing large variation in successive locations. This generally provides an improvement in accuracy as compared to two static beacons, but a lower accuracy compared to that obtained with two mobile beacons.

We also used one mobile beacon in combination with one static beacon (1S1M.O), with the maximum velocity of the beacon set to the default value of 2.5 m/s. While an additional static beacon reduces the amount of movement required by the mobile beacon to some extent, the mobile beacon still misses some of the calculated optimal locations. Estimated AUV trajectories with this beacon configuration with one AUV are shown in Fig. 16.4c, while those with three AUVs are shown in Fig. 16.5b.

16.3.5 Two Mobile Beacons

We found that two mobile beacons generally outperformed other beacon configurations. The proposed optimal approach (2M.O) resulted in the lowest RMSE in all the scenarios. However, fixed formation motion (2M.F) had only a slightly higher aggregate error in many scenarios, and a lower error in some individual runs of the simulation. This is because the constant velocity assumption yields a poor prediction of future AUV positions (and uncertainties) in case an AUV changes direction shortly after the prediction is made. This results in a sub-optimal beacon placement. Regardless of the relative positions between the beacons and AUVs, closer distance provides more accurate range measurements, due to the characteristics of the error in range measurements. Fig. 16.4d shows a plot of the trajectory estimated with range measurements from two mobile beacons using the proposed approach (2M.O) in a one

AUV experiment. Estimated trajectories in a three AUV experiment with two mobile beacons in formation (2M.F) are shown in Fig. 16.5c, while those with the two beacons using the proposed method (2M.O) are shown in Fig. 16.5d.

16.4 Conclusion

This chapter presented an approach for repeated, optimal placement of mobile acoustic beacons providing range measurements to a group of AUVs. The resulting mobile beacon trajectory follows the motion of the AUVs. It attempts to provide range measurements from a direction that is close to the direction of maximum uncertainty in the position of the AUVs. Our approach addresses the problem of limitation in operational area imposed by fixed, static beacons. It also avoids the problem of collinearity. Our approach is especially useful in shallow water and in coastal areas, where the error in range measurements increases significantly with distance due to reflections and occlusions caused by nearby obstacles.

Ranging systems based on two-way transmissions require dedicated time-slots for each AUV. This limits the scalability of the system because the time required to serve all the AUVs increases linearly with the number of AUVs. In order to achieve the best results with a large number of AUVs, a tradeoff needs to be made between how often to reevaluate optimal positions and which AUVs to serve first. Ranging systems based on one-way-travel-time would be more efficient since the number of acoustic transmissions is reduced. Further, for localization, the AUVs would be passive receivers (although they would still need to broadcast their position estimates). These systems can potentially result in better accuracy, especially when a larger group of AUVs needs to be served by the beacons.

A number of improvements to the system are possible. The beacons rely on position estimates reported by the AUVs to track them. It is theoretically possible for the beacons to obtain range measurements to the AUVs and track them using trilateration. Further, the beacons assume a constant velocity (and heading) to predict the future position and uncertainty of the AUVs. A better prediction can be made if the AUVs share additional information, such as a future waypoint.

17 Impact on Multi-Robot Adaptive Sampling

ENVIRONMENTAL sampling with a team of robots was the primary application that shaped much of the work done in this thesis. Robots need to communicate in order to cooperate with each other. They also need an accurate estimate of their position for accurately geo-tagging environmental measurements. So far, we have implemented techniques for improving localization and developed miniature acoustic modems for communication. In this chapter, we review some methods for cooperative, multi-robot adaptive sampling. We then study the effect of localization accuracy and communication errors on the performance of such a method. Experiments are performed in simulation.

17.1 Related Work

While the use of robots for environmental sampling provides several benefits, we have seen earlier in Chapter 9 that not all sensing locations may be interesting. By processing already recorded measurements and using a model of the environment, robots can adapt their trajectory to target the most informative sampling locations and regions of interest. We earlier used knowledge about the vertical distribution of the region of interest (i.e., a stratified layer rich in bacteria) to adapt the robot trajectory.

Several existing works utilize already sampled data to compute future trajectory or sampling points and use various strategies for multi-robot coordination. In [116], the authors use a combination of static sensing nodes and a robot. They reconstruct the temperature field using measurements from the static nodes and generate an optimal robot path based on it for further sampling. Soares et al. in [117] proposed an algorithm for formation-based plume tracing using two ASVs and an AUV, which they evaluated in simulation. The ASVs attempt to steer the group based on chemical measurements combined from all the three robots. Luo et al. in [118] use a mixture of locally learned Gaussian Processes (GPs), one per robot, for collaborative sampling. An information-theoretic criterion is used for guiding the robots. A number of approaches similarly using Gaussian Process regression have been presented, such as those in [81], [119], [120]. In [81], [120], the authors further use Voronoi tessellation to assign sampling regions to robots. In [121], the authors additionally take into account the disturbances caused by ocean currents for planning robot motion. Various path planning approaches for adaptive sampling are compared in [122].

When using a GP (or a log-GP for log-normal distributions) to model the field using already sampled data, an entropy criterion to evaluate yet unobserved locations is derived in [123]. The authors in [81] then use this criterion to choose the neighboring location with the highest entropy as the future sampling point. For coordination, the robots dynamically recalculate Voronoi partitions repeatedly and then sample within their own partitions. In [124], the the performance of the adaptive sampling approach is compared with continuous acoustic communication and radio communication on the surface.

We adopt an approach similar to the one presented in [81], and evaluate the impact of errors in both, localization and communication, on performance on the sampling task.

17.2 Methodology

Robots continuously gather environmental measurements using the on-board sensors, and occasionally broadcast their measurements. Note that we consider robots that are equipped with sensing probes (like the Vertex AUV) for in-situ measurements of a scalar field. Each robot uses its own measurement data and data communicated by others as inputs for Gaussian Process regression. It then uses the posterior entropy of the inferred field to decide on future sampling points. For coordination, the robots periodically calculate and broadcast their Voronoi cell centers, and then sample within their own Voronoi partitions. They also notify other robots when a specific area has been sufficiently sampled.

17.2.1 Path Planning for Adaptive Sampling

Since environmental data often have log-normal distributions, it is common practice to directly use log-Gaussian Process (lGP) for regression, instead of using a GP with the logarithm of the raw measurement. Suppose y_x is a measurement recorded at location $\vec{x}$, and Y_x denotes a lGP. Let $z_x = \log_e y_x$. Then, z_x is normally distributed. Let Z_x denote a GP for the log-measurement z_x.

The following three relations are derived in [123]. The posterior mean and variance (for a query location $\vec{x}$) for the lGP are given by

$$\mu_{Y_x|d_i} = \exp\left(\mu_{Z_x|d_i} + \sigma^2_{Z_x|d_i}/2\right), \tag{17.1}$$

$$\sigma^2_{Y_x|d_i} = \mu^2_{Y_x|d_i}\left[\exp\left(\sigma^2_{Z_x|d_i}\right) - 1\right], \tag{17.2}$$

where $\mu_{Z_x|d_i}$ and $\sigma^2_{Z_x}$ are the posterior mean and variance of the GP, and d_i is the set of already sampled data at time step i. Further, the posterior map entropy for the lGP at an unvisited query location x_{i+1} is given by

$$\mathbb{H}\left[Y_{x_{i+1}}|d_i\right] = \log\sqrt{2\pi e\sigma^2_{Z_{x_{i+1}}|d_i}} + \mu_{Z_{x_{i+1}}|d_i}. \tag{17.3}$$

Unvisited locations in the neighborhood with the highest posterior map entropy according to Eq. (17.3) are chosen as the next target sampling point. This directs the robots towards either sparsely sampled areas or to regions with high expected measurement values. Note that the robots continue to collect measurements along the way as they travel towards their respective computed target points.

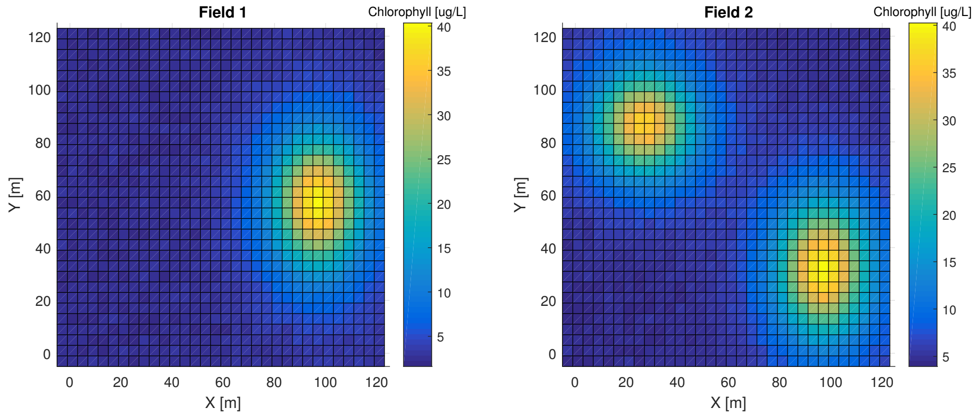
17.2.2 Coordination

For coordination between the robots, we use Voronoi partitioning, with robots sampling within their own partitions. The Voronoi centers are periodically recalculated as follows. Since robots occasionally share their measurements, they have a recent estimate of each others' position. Each robot calculates the weighted mean of unvisited locations closest to itself. The locations are weighted by posterior entropy calculated using Eq. (17.3). The result is used as the new Voronoi centroid of the robot and is broadcast to others. This is the same as the approach used in [81]. We recalculate the Voronoi centroids after fixed intervals of 300 s.

Additionally, we divide the area of operation into a coarse grid of cell-size 14 m ×14 m. This coarse grid cell is further divided into fine cells of size 1.4 m ×1.4 m. A coarse grid is marked as sampled if 20 % of the fine cells within it correspond to already measured locations. Note that the measurement locations themselves are not precisely known and are estimated by the state estimator of the robot. The size of the fine grid roughly corresponds to the spatial sampling resolution (with a maximum speed of 0.7 m/s and a sampling period of 2 s, the robots sample roughly every 1.4 m). The 20 % figure corresponds to the robot traversing the span of the coarse grid cell twice.

17.2.3 Sharing Data

Each robot broadcasts three different types of messages. Firstly, they broadcast their Vononoi centroids at pre-determined intervals, as explained in the previous section. Secondly, they broadcast the coarse grid cells marked as having been already sampled. Finally, if there is no new Voronoi centroid or recently sampled grid cell, the robots broadcast their recent measurement with the highest magnitude. To do so, they store recent measurements in a cache, which is cleared after a broadcast. Empirically, each robot ends up broadcasting approximately 25 % of its measurements.



(a) Field 1

(b) Field 2

Figure 17.1: Two different chlorophyll fields, with one or two hotspots.

17.3 Experimental Setup

We consider a simulation environment with 3 AUVs. The AUVs gather measurements of chlorophyll concentration once every 2 s as they travel. This sampling frequency is in line with the sensing module on the Vertex AUV. We set the maximum cruise speed of the AUVs to 0.7 m/s. Two static beacons in the arena provide range measurements to aid localization. Starting from their initial position, the robots record measurements and use them to plan their future path.

We first run lawnmower surveys with one robot (in both fields) and record the amount of time required to complete the survey. Then, we run adaptive multi-robot sampling missions for the same duration of time for each scenario.

We evaluate the performance of the adaptive sampling method as follows. As the robots gather measurements, they periodically perform regression. We use the RMSE between the estimated scalar field (posterior from the regression) and the actual field to quantify the estimation error. Then, as new measurements are included in the regression, the estimation would improve and the RMSE would decrease. We expect the adaptive sampling approach to direct the robot towards more informative regions, resulting in quicker decrease in RMSE. We compare the performance with different levels of localization and communication errors.

17.3.1 Environment

We simulate 2D spatial scalar fields of chlorophyll within a 140 m ×140 m region. High chlorophyll concentrations are found in nature in algal blooms, with photosynthetic bacteria such as in Lake Cadagno and Lake Zurich (as described in Chapter 9), and in various other biological phenomena in lakes. We set the maximum value of chlorophyll concentration to 40 μg/L, which is within the range of values measured during our sampling missions (see

Data field	Description	Size (bits)
Id	Id of the sender robot	4
Type	Type of message (Voronoi centroid / sampled coarse grid cell / chlorophyll measurement)	2
X	X coordinate (meters, 0.25 m resolution)	12
Y	Y Coordinate (meters, 0.25 m resolution)	12
Arg	Value of the data associated with the position	16

Table 17.1: List of fields in a data packet along with their sizes. The total size of a data packet is 46 bits.

Fig. 9.5). To simulate measurement noise, we add Gaussian noise with a standard deviation $\sigma = 1.0$. We simulate two chlorophyll fields, having one or two separate chlorophyll hotspots, as shown in Fig. 17.1. The spatial variation of chlorophyll follows a log-normal distribution.

The choice of a log-normal distribution to simulate the spatial field is justified. Environmental data and biological phenomena are often found to fit a log-normal distribution [125]. They also model the behavior of many water-quality variables well [126]. In particular, spatial variation of chlorophyll has been found to follow log-normal distribution in [127], [128].

17.3.2 Communication

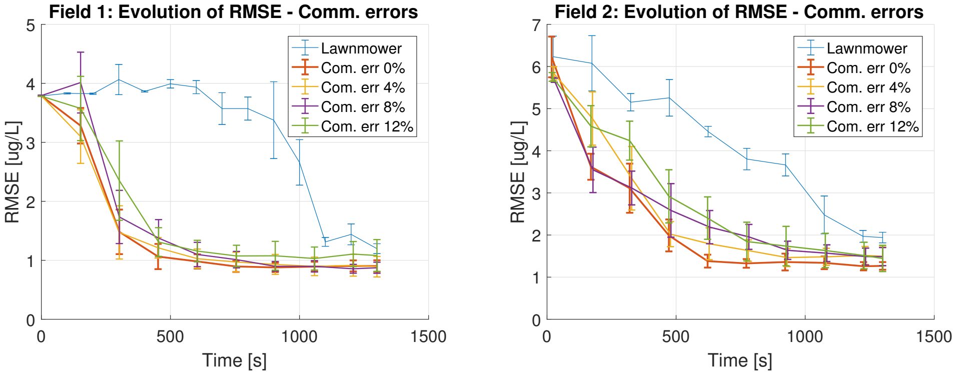
We use the simulated acoustic transceiver described in Section 14.3 for communication. Each robot is assigned a time-slot of 2 s in a round-robin fashion, similar to our other outdoor and simulated experiments. Data packets are structured as shown in Table 17.1. The data packet has a total size of 46 bits (without error correction), which is within the throughput of our acoustic hardware.

17.3.3 Acoustic Ranging

Robots perform inertial localization, combined with range measurements from two external, fixed acoustic beacons. The error characteristics are as described in Section 14.3. The beacons are each assigned 4 s time-slots during which they provide range measurements to the three robots. For the purpose of this simulation, we assume no interference between acoustic communication and range measurements. Further, the beacons are simulated as being perfectly fixed, with their locations precisely known.

17.4 Impact of Communication and Localization Errors

We simulate different degrees of loss of messages. In reality, messages with errors are discarded, and have the same effect as lost messages (errors can be detected with a high degree of accuracy with a simple checksum). For localization, we compare three different scenarios: error-free



(a) Field 1 (b) Field 2

Figure 17.2: Evolution of RMSE for different levels of communication loss in two different fields. (a) In Field 1 with one hotspot, while there is a trend towards degrading performance with higher communication error, the difference in performance is not significant. (b) In Field 2, the reduction in RMSE is slower when communication loss is higher.

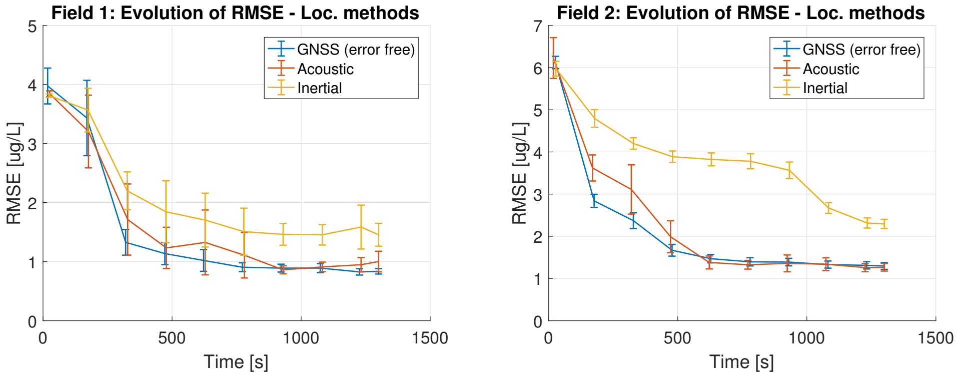
GNSS, positioning aided by acoustic range measurements, and inertial positioning (dead reckoning). Note that when using dead reckoning, we provide GNSS position updates once every 200 s to prevent the estimated position from diverging too far and being outside the operational arena. Results are averaged over 15 datasets.

17.4.1 Communication Errors

We performed adaptive sampling missions with communication loss at a rate of up to 12 %. The robots used acoustic range measurements for localization. A plot of the evolution of RMSE for different degrees of message loss over time is shown in Fig. 17.2. A plot for the lawnmower mission is also shown for reference.

We found that generally, with higher communication loss, it takes longer to achieve a smaller RMSE of the estimated field. However, the difference in performance in Field 1 with one hotspot is insignificant. For Field 2, the difference is more pronounced.

Communication errors affect performance in two ways. Firstly, robots do not receive some of the measurements shared by other robots. Secondly, they may not receive the message containing Voronoi centroids or those marking grid-cells as sampled. The robots then end up sampling within Voronoi partitions of other robots at the same time, or in grid cells already sampled earlier by other robots. Since Field 1 has only one contiguous region of interest, the difference in performance in it is not significant. In Field 2, with lost messages, the robots may end up sampling near the same hotspot, ignoring the other one for a while.



(a) Field 1

(b) Field 2

Figure 17.3: Evolution of RMSE for different localization methods. In both fields, the reduction in RMSE is faster with better localization. With inertial localization, the RMSE starts to plateau at a higher level.

17.4.2 Localization Quality

We performed adaptive sampling missions with no communication loss, and with different localization methods. A plot of the evolution of RMSE of the estimated field over time is shown in Fig. 17.3. In either field, better localization resulted in better estimation of the field, as expected. With inaccurate localization, the measured data are associated with inaccurate geo-references, resulting in an erroneous estimation of the field. The effect is especially pronounced with inertial localization, where the position error grows without bound (although it is somewhat bounded in our simulation due to the rare position updates). This highlights the importance of improved localization for such sampling tasks.

17.5 Conclusion

The previous chapters were largely dedicated to improving underwater robot localization in various scenarios, with some chapters implementing communication. This closing chapter aimed at studying the effect of quality of localization and communication on multi-robot environmental sampling, the key application driving the work in this thesis. Simulation results showed that improved communication or localization resulted in improved performance on the sampling application. While the results were qualitatively expected, they quantify the impact of both localization and communication and highlight their importance for multi-robot applications.

Communication loss results in erroneous or ineffective coordination behavior. For example, in the presented application, if a new Voronoi centroid broadcast by one robot is not received by another, the second robot will use an incorrect Voronoi partitioning, possibly sampling in the same region as the first robot. Between two robots, this can be trivially solved with an acknowledgement. With a large number of robots, simple individual acknowledgement protocols can become unscalable, especially considering the use of long time-slot-based acoustic channel access. In such cases, borrowing ideas from a number of algorithms in the domain of distributed computing can help keep overhead to a minimum. Algorithms such as the two- and three-phase commit [129] are essentially systems of acknowledgement or confirmation between multiple entities. Some of those concepts can be borrowed to develop lightweight, low-overhead discrete consensus algorithms for guaranteeing agreement between multiple robots. While any such method will add overhead in terms of time and communication bandwidth, that has to be weighed against the cost of erroneous coordination behavior – such as two robots spending time sampling in the same region.

Conclusion and Outlook Part V

18 Conclusion and Outlook

Over the course of this thesis, our primary focus has been facilitating environmental sampling in underwater environments using multi-robot teams. The primary challenges to that end were that of localization and communication – fundamental requirements for a multi-robot system. Accordingly, the core of our work dealt with these two requirements. While our methods were tested in aquatic environments, some of them may also be applied to other scenarios where GNSS positioning is unavailable and communication is expensive.

This project started with a miniature AUV prototype equipped with environmental sensing probes. We began this thesis with the development of subsystems for individual robots for control, localization, and communication. Due to the lack of an underwater positioning system, the sampling missions at the time were performed with only inertial positioning. In the early phases of the project, there were few commercially available peripherals for communication or localization compatible with miniature underwater robots. Therefore, we developed acoustic transceivers for data transmission and range measurements and tested them in outdoor environments. We also used acoustic modems that became commercially available later in the project. We subsequently moved on to multi-robot systems, developing methods for coordination and cooperative localization.

Many of the experiments in this work were performed in real-world environments. As a result, a significant portion of the work involved engineering effort to prepare the equipment for outdoor deployments. For the simulation experiments, we reproduced realistic conditions to the extent possible. Further, we developed all our methods with the constraint that they should be feasible to implement on existing hardware. This includes taking into account communication and computation limitations on real robots.

This chapter provides a summary of our work and outlines possible future research directions.

18.1 Summary

The work done in this dissertation can be broadly divided into three components. The first component comprises the engineering work done for setting up tools and experimental facilities. The next two components describe our research contributions respectively towards individual underwater robot subsystems and underwater multi-robot systems. Each component is explained in detail in a dedicated part of the manuscript.

In **Part II: Platforms and Experimental Setup**, we introduced the robot platforms and tools developed for facilitating systematic and safe outdoor experiments. We also described our experimental setup in detail. This engineering effort spanned the length of the thesis, running in parallel with the research in the two subsequent parts. Our work is summarized below.

- The development of Vertex AUV prototype was already in progress at the start of this thesis. As a part of the thesis, contributions were made to the development of software components (for navigation, localization, and control) as well as hardware modifications and enhancements (such as the addition of new IMU and GNSS receivers, modification to acoustic transceiver hardware).
- We developed a new ASV platform from the ground up, using hardware components and architecture similar to the Vertex AUV. The ASVs served as a tool for testing new hardware and algorithms before they were deployed on the AUV. They also served as proxies for the AUVs for outdoor experiments, leveraging GNSS availability for providing ground truth.
- We developed software tools for monitoring and sending commands to the robots during outdoor operations. The tools were developed using open-source libraries. However, existing software products were not adequate for multiple aquatic robots.
- We evaluated different experimental arenas – swimming pools, lake shore, and a boat – for performing experiments. We also set up a multi-tiered simulation framework, combining different levels of simulated and realistic conditions, for evaluating the developed methods.

In **Part III: Single-Robot Subsystems**, we described the development of control, localization and communication methods for individual robots. Following is a summary of our contributions.

- For individual robots, we developed and implemented methods for state estimation, control and navigation. We employed an EKF for state estimation, which fused inertial and other measurements with a model of the dynamics of the robot. We further implemented waypoint-based navigation, which allowed us to specify the trajectory of the robot with a collection of waypoints to visit.
- We implemented and demonstrated an approach for feature tracing and adaptive sampling in real-world environments. We showed that the AUV was able to locate, trace and

capture measurements within a thin stratified layer rich in photosynthetic bacteria in Lake Cadagno in southern Switzerland. Since the acoustic positioning system was not developed yet, the AUV performed inertial localization during this experiment.

- We developed acoustic transceivers for underwater range measurements and communication. We performed range measurements based on one-way-travel-time, making our system scalable. We also embedded data in acoustic signals for communication. In addition to using traditional error detection/correction methods, we leveraged kinematic constraints of robots to detect errors in navigation data transmitted by other robots. We used ASVs as mobile beacons providing range measurements to AUVs.
- Since the time between acoustic ranging updates can be several seconds, we noted that accurate inertial localization during this time is crucial. Hence, we developed a framework for estimating the kinematic and dynamic model parameters used for inertial localization. Our method was also able to correct trajectory estimation errors in the recent past using newly estimated parameters.

Finally, in **Part IV: Multi-Robot System**, we focused on how a team of robots could benefit from cooperation and information sharing, primarily with regard to localization. Following is a summary of our contributions.

- We began by introducing our high-fidelity, physics-based simulation setup. We used the Webots simulation software. We used data from outdoor experiments to calibrate inertial positioning errors in individual robots. We further calibrated errors in acoustic ranging and communication to reproduce the performance of real hardware in outdoor environments.
- We developed an approach for cooperative localization using pairwise range-only relative measurements between robots. We also derived an information-theoretic criterion for choosing a peer robot for a pairwise range measurement. We showed that our methods improved localization accuracy and provided consistent position estimates.
- We observed that the accuracy of range measurements tends to decrease with distance. We also noted that range measurements are the most useful when they come from a direction coinciding with the direction of maximum uncertainty in a robot's position estimate. Therefore, we developed an approach for calculating the most optimal position of two or more mobile acoustic beacons providing navigation support to a group of AUVs. In simulation, we demonstrated that our approach leads to an improvement in localization accuracy of the AUVs.
- Having developed methods for underwater localization and communication, we assessed the impact of quality of both on environmental sampling using a team of robots. We implemented an existing approach for multi-robot adaptive sampling, and compared the sampling performance with varying degrees of localization and communication errors.

18.2 Discussion and Future Work

The field of underwater robotics has seen continued advancement over the years. Development of new hardware as well as newly developed algorithms have improved the capabilities of underwater robots. We believe that, in the near future, underwater multi-robot systems will become ubiquitous for environmental sampling applications and beyond. This thesis attempted to tackle the fundamental problems of localization and communication when using miniature underwater robots. We also identified several facets of the field that have the potential for improvement and innovation. Following is a discussion of the impact of our work and potential future research directions.

Logistics and Hardware

In advancing towards our goal of using robots to perform measurements in underwater environments, a significant challenge we faced was that of availability of suitable, ready-made hardware. The recent years have seen increasing availability of miniature underwater robot platforms as well as various peripherals for localization and communication. However, the available platforms are still few in number and variety (in comparison to ground and aerial robots), and each platform is highly customized to their specific, intended application. The lack of generalization slows down research progress since it is difficult to reuse existing methods that may have been developed on a different platform. This problem has been tackled to some extent in the field of aerial and ground robotics.

Localization

An important approach to acoustic localization lacking in our work is the use of Ultra Short Base Line (USBL) multi-receiver array. When installed on the AUV, it can measure the direction of arrival (i.e., bearing) of an acoustic signal in addition to range/time-of-flight. This immediately provides a 2D or 3D relative position with respect to a beacon, an improvement over a 1D range measurement. While this has been implemented in numerous works (for instance, [52]), we were constrained due to several factors. A smaller baseline implies a smaller difference in time of arrival. Resolving that requires higher sampling frequency, and hence, higher memory and processing capability beyond what our hardware could provide. We also did not find equivalent commercial products. A redesigned acoustic transceiver built using a more powerful processor would be able to perform direction of arrival measurement.

We used ranging-only measurements to develop the cooperative localization algorithm in Chapter 15, as well as for coordinated movement of mobile acoustic beacons in Chapter 16. With acoustic range and bearing measurements combined with suitable adjustments to both methods, localization accuracy can be further improved.

Communication

We introduced our implementation of acoustic communication in Chapter 11. We also introduced our approach to error detection using repetition as well as using kinematic constraints. We noted in Section 11.2.5 that many of the existing error detection approaches (such as Hamming [80]) assume symmetric binary errors (i.e., bit 1 and bit 0 are equally likely to be corrupted), which is not the case for the strategy implemented by us. There is room for the development of new error detection and correction algorithms that take into account the characteristics of the acoustic channel, as well as the specific implementation of communication. Further, with improved computation capability, there is also potential for implementation of existing, more complex modulation schemes on miniature hardware.

Optical modems can provide high bandwidth communication but are usually limited in range. Small-sized, lightweight optical modems are starting to become available, and can be used for short-range communication. This has many potential benefits. An example is fast data exchange with a nearby surface vessel without completely resurfacing and waiting to establish a radio link. Resurfacing events, a common approach with underwater robots to broadcast data, disrupt the ongoing activity by requiring robots to wait on the surface. However, optical modems are highly directional and require alignment for communication, which is challenging to achieve. An approach to hybrid acoustic-optical communication with underwater robots is presented in [130].

It must be noted that radio communication on the surface also poses unintuitive challenges. When a radio antenna is close to (but still above) the water surface, water acts as a lossy ground and absorbs a significant amount of the energy of the radio waves. Attaching a long vertical fin or a tower housing a radio antenna on a miniature AUV may not be feasible, further exacerbating the problem. Approaches such as adding a ground plane of a suitable shape and material to address the problem need to be studied and experimented with.

Multi-Robot Coordination

While advancements in underwater acoustic communication will likely bring about an improvement in data throughput, communication errors and losses are unavoidable. A number of approaches have been developed for multi-robot coordination for various purposes such as mapping and task-allocation. However, many of them do not address communication failures and are not tolerant to them [131]–[133].

We attempted to develop a generic approach to coordination by bringing robots to an agreement. For robustness to communication failures, we borrowed ideas from the domain of distributed computing, in particular from the Raft [134] and Paxos [135] consensus algorithms. The most trivial approach to address potential communication losses between two agents is to use acknowledgements. At a fundamental level, these algorithms are systematic protocols of acknowledgements and confirmations in a multi-agent group. We attempted to tailor these ideas to robotic systems using low-bandwidth, TDMA-based acoustic communication.

There is scope for analyzing the potential improvements in multi-robot applications brought about by robust coordination approaches.

18.3 Lessons Learnt

Research in underwater multi-robot systems is an ambitious endeavor. It is both, a research and technical challenge as well as a logistical one. The challenges posed by the environment make it significantly more difficult than research in ground or aerial robotic systems. In this section, we share some key insights and lessons we learnt during this project.

We started this thesis with real-world experiments using an AUV prototype and a few ASVs. The development of the AUV hardware and software continued in parallel. We later used the ASVs as surrogates for the AUV for quick experiments, and eventually shifted towards simulation. This approach was not planned, but was forced due to the difficulty in performing outdoor experiments and eventual issues in production of the AUV hardware.

Our approach required significant engineering effort in preparing the AUV and related equipment for outdoor experiments, and ensuring safety from catastrophic failure. In addition, we needed to set up a workflow of thorough testing before outdoor deployment. This effort often diverted focus away from our core research goals. In hindsight, we believe a better approach would be to (i) disassociate prototype development from research activity and (ii) use a combination of partial emulation and simulated experiments in the initial stages of the development of new algorithms and methods.

We, however, stress the importance of real-world experiments and avoiding overreliance on simulation. Care must be taken to not make assumptions that may not hold true in reality. Working with the AUV initially provided us with hands-on experience of real-world conditions and constraints, which we accounted for in simulation. In the later chapters (Chapter 15 in particular), it helped us transition easily from simulation back to reality.

We have attempted to use several different experimental arenas – water tanks and swimming pools. While they offered a structured and controlled environment, we found that they were not realistic in several aspects. We eventually performed all our experiments in lakes.

Developing and fabricating underwater hardware is difficult. The harshness of the environment requires that the hardware prototypes be of a very high quality. Therefore, whenever possible, development of new hardware from the ground up must be avoided, and existing hardware devices and peripherals must be acquired or re-used. This is perhaps true for robotics in general, but especially so for underwater robotics. We initially developed acoustic transceivers in-house. However, owing to production difficulties, we acquired and integrated the newly available DiveNET acoustic transceivers.

Finally, an important consideration when planning outdoor, underwater multi-robot experiments is the logistical complexity. They require transporting material between the lab and the place of experiment. Errors detected in the field often cannot be rectified immediately, and require a trip to the lab. Since the robots are out of sight when submerged, the quality or success of an experiment can often be deduced only after studying the experimental data.

Bibliography

[1] *World Oceans Day: Visualizing our impact on our ocean economy - World Economic Forum*, https://www.weforum.org/agenda/2020/06/human-impact-ocean-economy, Accessed: 2021-04-29.

[2] J. Yuh, G. Marani, and D. R. Blidberg, "Applications of marine robotic vehicles", *Intelligent service Robotics*, vol. 4, no. 4, pp. 221–231, 2011.

[3] Whitcomb, Louis and Yoerger, Dana R. and Singh, Hanumant and Howland, Jonathan, "Advances in Underwater Robot Vehicles for Deep Ocean Exploration: Navigation, Control, and Survey Operations", in *Robotics Research*, J. M. Hollerbach and D. E. Koditschek, Eds., Springer London, 2000, pp. 439–448.

[4] *Bathymetric Surveys – United States Geological Survey*, http://www.usgs.gov/centers/oki-water/science/bathymetric-surveys, Accessed: 2021-07-27.

[5] J. G. Bellingham, M Deffenbaugh, J. Leonard, and H Schmidt, "Arctic under-ice survey operations", in *International Symposium on Unmanned Untethered Submersible Technology*, 1993, pp. 1–9.

[6] A. Q. Li, I. Rekleitis, S. Manjanna, N. Kakodkar, J. Hansen, G. Dudek, L. Bobadilla, J. Anderson, and R. N. Smith, "Data correlation and comparison from multiple sensors over a coral reef with a team of heterogeneous aquatic robots", in *International Symposium on Experimental Robotics*, Springer, 2016, pp. 717–728.

[7] B. Bingham, B. Foley, H. Singh, R. Camilli, K. Delaporta, R. Eustice, A. Mallios, D. Mindell, C. Roman, and D. Sakellariou, "Robotic tools for deep water archaeology: Surveying an ancient shipwreck with an autonomous underwater vehicle", *Journal of Field Robotics*, vol. 27, no. 6, pp. 702–717, 2010.

[8] H. Stuart, S. Wang, O. Khatib, and M. R. Cutkosky, "The Ocean One hands: An adaptive design for robust marine manipulation", *The International Journal of Robotics Research*, vol. 36, no. 2, pp. 150–166, 2017.

[9] *MH370 – Ocean Infinity*, http://oceaninfinity.com/projects/mh370/, Accessed: 2021-07-15.

[10] C. Mai, S. Pedersen, L. Hansen, K. L. Jepsen, and Z. Yang, "Subsea infrastructure inspection: A review study", in *2016 IEEE International Conference on Underwater System Technology: Theory and Applications (USYS)*, 2016, pp. 71–76.

[11] I. Schjølberg, T. B. Gjersvik, A. A. Transeth, and I. B. Utne, "Next generation subsea inspection, maintenance and repair operations", *IFAC-PapersOnLine*, vol. 49, no. 23, pp. 434–439, 2016.

[12] J. Vaganay, M. Elkins, D. Esposito, W. O'Halloran, F. Hover, and M. Kokko, "Ship Hull Inspection with the HAUV: US Navy and NATO Demonstrations Results", in *OCEANS 2006*, 2006, pp. 1–6.

[13] O. Sepúlveda Steiner, D. Bouffard, and A. Wüest, "Persistence of bioconvection-induced mixed layers in a stratified lake", *Limnology and Oceanography*, vol. 66, no. 4, pp. 1531–1547, 2021.

[14] *Bluefin-21 Unmanned Underwater Vehicle (UUV)*, gdmissionsystems.com/products/underwater-vehicles/bluefin-21-autonomous-underwater-vehicle, Accessed: 2020-12-30.

[15] *REMUS 100 AUV*, www.hydroid.com/products, Accessed: 2020-12-30.

[16] *Bluefin-9 Unmanned Underwater Vehicle (UUV)*, gdmissionsystems.com/products/underwater-vehicles/bluefin-9-autonomous-underwater-vehicle, Accessed: 2020-12-30.

[17] *YSI EcoMapper AUV*, www.ysi.com/ecomapper, Accessed: 2021-07-30.

[18] *Riptide (TM) family of autonomous unmanned undersea vehicles*, www.baesystems.com/en-us/product/riptide-family-of-autonomous-undersea-vehicles, Accessed: 2020-12-30.

[19] J. E. Manley and J. Smith, "Rapid development and evolution of a micro-UUV", in *OCEANS 2017-Anchorage*, IEEE, 2017, pp. 1–4.

[20] C. Osterloh, T. Pionteck, and E. Maehle, "MONSUN II: A small and inexpensive AUV for underwater swarms", in *ROBOTIK 2012; 7th German Conference on Robotics*, 2012, pp. 1–6.

[21] A. Amory and E. Maehle, "SEMBIO - a small energy-efficient swarm AUV", in *OCEANS 2016 MTS/IEEE Monterey*, 2016, pp. 1–7.

[22] A. Hackbarth, E. Kreuzer, and E. Solowjow, "HippoCampus: A micro underwater vehicle for swarm applications", in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2015, pp. 2258–2263.

[23] P. C. Abreu, J. Botelho, P. Góis, A. Pascoal, J. Ribeiro, M. Ribeiro, M. Rufino, L. Sebastião, and H. Silva, "The MEDUSA class of autonomous marine vehicles and their role in EU projects", in *OCEANS 2016 - Shanghai*, 2016, pp. 1–10.

[24] F. S. Schill, A. Bahr, and A. Martinoli, "Vertex: A New Distributed Underwater Robotic Platform for Environmental Monitoring", in *International Symposium on Distributed Autonomous Robotic Systems, 2016*, vol. 6, Springer Proceedings in Advanced Robotics, 2018, pp. 679–693.

[25] *Heron Unmanned Surface Vessel*, clearpathrobotics.com/heron-unmanned-surface-vessel, Accessed: 2020-12-30.

[26] *C-Worker 4 Compact Work Class ASV*, www.asvglobal.com/product/c-worker-4, Accessed: 2020-12-30.

[27] *DiveNET Microlink Acoustic Modem*, https://www.divenetgps.com/sealink.

[28] E. Gallimore, J. Partan, I. Vaughn, S. Singh, J. Shusta, and L. Freitag, "The WHOI micromodem-2: A scalable system for acoustic communications and networking", in *OCEANS 2010 MTS/IEEE SEATTLE*, 2010, pp. 1–7.

[29] M. Chitre, I. Topor, and T. Koay, "The UNET-2 modem — An extensible tool for underwater networking research", in *2012 Oceans - Yeosu*, 2012, pp. 1–7.

[30] B. Benson, Y. Li, B. Faunce, K. Domond, D. Kimball, C. Schurgers, and R. Kastner, "Design of a Low-Cost Underwater Acoustic Modem", *IEEE Embedded Systems Letters*, vol. 2, no. 3, pp. 58–61, 2010.

[31] G. Cario, A. Casavola, M. Lupia, and C. Rosace, "SeaModem: A low-cost underwater acoustic modem for shallow water communication", in *OCEANS 2015 - Genova*, 2015, pp. 1–6.

[32] A. Quraishi, A. Bahr, F. Schill, and A. Martinoli, "Autonomous Feature Tracing and Adaptive Sampling in Real-World Underwater Environments", in *IEEE International Conference on Robotics and Automation*, 2018, pp. 5699–5704.

[33] A. Quraishi, A. Bahr, F. Schill, and A. Martinoli, "Easily Deployable Underwater Acoustic Navigation System for Multi-Vehicle Environmental Sampling Applications", in *IEEE International Conference on Robotics and Automation*, 2019, pp. 3464–3470.

[34] A. Quraishi, A. Bahr, F. Schill, and A. Martinoli, "A Flexible Navigation Support System for a Team of Underwater Robots", in *IEEE International Symposium on Multi-Robot and Multi-Agent Systems (MRS)*, 2019, pp. 70–75.

[35] A. Quraishi and A. Martinoli, "Online Kinematic and Dynamic Parameter Estimation for Autonomous Surface and Underwater Vehicles", in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2021. To appear.

[36] A. Quraishi and A. Martinoli, "Distributed Cooperative Localization with Efficient Pairwise Range Measurements", in *International Symposium on Distributed Autonomous Robotic Systems, 2021*, Springer Proceedings in Advanced Robotics, 2021. To appear.

[37] A. Quraishi and A. Martinoli, "Coordinated Path Planning for Surface Acoustic Beacons for Supporting Underwater Localization", in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2021. To appear.

[38] *YSI EXO2 Multiparameter Sonde*, http://www.ysi.com/EXO2, Accessed: 2021-04-06.

[39] *MAV'RIC autopilot library*, http://lis-epfl.github.io/MAVRIC_Library/, Accessed: 2021-04-06.

[40] *Raspberry Pi Zero W*, https://www.raspberrypi.org/products/raspberry-pi-zero-w/, Accessed: 2021-04-06.

[41] *MAVLink messaging protocol*, https://mavlink.io/en/, Accessed: 2021-04-06.

[42] O. Michel, “Cyberbotics Ltd. Webots TM: Professional Mobile Robot Simulation”, *International Journal of Advanced Robotic Systems*, vol. 1, no. 1, pp. 39–42, 2004.

[43] Y. Zhang and M. A. Godin and J. G. Bellingham and J. P. Ryan, “Using an Autonomous Underwater Vehicle to Track a Coastal Upwelling Front”, *IEEE Journal of Oceanic Engineering*, vol. 37, no. 3, pp. 338–347, 2012.

[44] S. Petillo, H. Schmidt, P. Lermusiaux, D. Yoerger, and A. Balasuriya, “Autonomous & adaptive oceanographic front tracking on board autonomous underwater vehicles”, in *IEEE OCEANS*, DOI: 10.1109/OCEANS-Genova.2015.7271616 (10 pages), 2015.

[45] J. Das, F. Py, T. Maughan, T. O’Reilly, M. Messie, J. Ryan, K. Rajan, and G. S. Sukhatme, “Simultaneous tracking and sampling of dynamic oceanographic features with autonomous underwater vehicles and lagrangian drifters”, in *International Symposium on Experimental Robotics*, Springer Tracts in Advanced Robotics, vol. 79, 2014, pp. 541–555.

[46] Cazenave, François and Zhang, Yanwu and McPhee-Shaw, Erika and Bellingham, James G. and Stanton, Timothy P., “High-resolution surveys of internal tidal waves in Monterey Bay, California, using an autonomous underwater vehicle”, *Limnology and Oceanography: Methods*, vol. 9, no. 12, pp. 571–581, 2011.

[47] Y. Zhang and B. Kieft and M. J. Stanway and R. S. McEwen and B. W. Hobson and J. G. Bellingham and J. P. Ryan and T. C. O’Reilly and B. Y. Raanan and M. Messié and J. M. Smith and F. P. Chavez, “Isotherm Tracking by an Autonomous Underwater Vehicle in Drift Mode”, *IEEE Journal of Oceanic Engineering*, vol. 42, no. 4, pp. 808–817, 2017.

[48] K.-C. Ma, Z. Ma, L. Liu, and G. S. Sukhatme, “Multi-robot informative and adaptive planning for persistent environmental monitoring”, in *International Symposium on Distributed Autonomous Robotic Systems, 2016*, vol. 6, Springer Proceedings in Advanced Robotics, 2018, pp. 285–298.

[49] S. Manjanna, A. Q. Li, R. N. Smith, I. Rekleitis, and G. Dudek, “Heterogeneous Multi-Robot System for Exploration and Strategic Water Sampling”, in *IEEE International Conference on Robotics and Automation*, 2018, pp. 4873–4880.

[50] J. J. Leonard and A. Bahr, “Autonomous Underwater Vehicle Navigation”, in *Springer Handbook of Ocean Engineering*, M. R. Dhanak and N. I. Xiros, Eds. Cham: Springer International Publishing, 2016, pp. 341–358.

[51] P. A. Miller, J. A. Farrell, Y. Zhao, and V. Djapic, “Autonomous Underwater Vehicle Navigation”, *IEEE Journal of Oceanic Engineering*, vol. 35, no. 3, pp. 663–678, 2010.

[52] N. R. Rypkema, E. M. Fischell, and H. Schmidt, “One-way travel-time inverted ultra-short baseline localization for low-cost autonomous underwater vehicles”, in *IEEE International Conference on Robotics and Automation*, 2017, pp. 4920–4926.

[53] S. E. Webster, R. M. Eustice, H. Singh, and L. L. Whitcomb, “Advances in single-beacon one-way-travel-time acoustic navigation for underwater vehicles”, *The International Journal of Robotics Research*, vol. 31, no. 8, 935–950, 2012.

[54] C. Becker, D. Ribas, and P. Ridao, “Simultaneous sonar beacon localization & AUV navigation”, *IFAC Proceedings Volumes*, vol. 45, no. 27, pp. 200–205, 2012.

[55] A. Munafò, T. Furfaro, G. Ferri, and J. Alves, “Supporting AUV localisation through next generation underwater acoustic networks: Results from the field”, in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2016, pp. 1328–1333.

[56] L. Stutters, H. Liu, C. Tiltman, and D. J. Brown, “Navigation technologies for autonomous underwater vehicles”, *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 38, no. 4, pp. 581–589, 2008.

[57] S. Tuohy, N. Patrikalakis, J. Leonard, J. Bellingham, C Chryssostomidis, *et al.*, “Map based navigation for autonomous underwater vehicles”, *International Journal of Offshore and Polar Engineering*, vol. 6, no. 01, 1996.

[58] J. L. Leonard, R. N. Carpenter, and H. J. S. Feder, “Stochastic mapping using forward look sonar”, *Robotica*, vol. 19, no. 5, pp. 467–480, 2001.

[59] R. M. Eustice, O. Pizarro, and H. Singh, “Visually Augmented Navigation for Autonomous Underwater Vehicles”, *IEEE Journal of Oceanic Engineering*, vol. 33, no. 2, pp. 103–122, 2008.

[60] O. Hegrenaes and E. Berglund, “Doppler water-track aided inertial navigation for autonomous underwater vehicle”, in *OCEANS 2009-EUROPE*, 2009, pp. 1–10.

[61] Z. Song and K. Mohseni, “Long-Term Inertial Navigation Aided by Dynamics of Flow Field Features”, *IEEE Journal of Oceanic Engineering*, vol. 43, no. 4, pp. 940–954, 2018.

[62] M. Khaghani and J. Skaloud, “Autonomous Vehicle Dynamic Model-Based Navigation for Small UAVs”, *J. Inst. Navigation*, vol. 63, no. 3, pp. 345–358, 2016.

[63] J. Kelly and G. S. Sukhatme, “Visual-Inertial Sensor Fusion: Localization, Mapping and Sensor-to-Sensor Self-calibration”, *The International Journal of Robotics Research*, vol. 30, no. 1, pp. 56–79, 2011.

[64] M. Li, H. Yu, X. Zheng, and A. I. Mourikis, “High-fidelity sensor modeling and self-calibration in vision-aided inertial navigation”, in *IEEE International Conference on Robotics and Automation*, 2014, pp. 409–416.

[65] Z. Taylor and J. Nieto, “Motion-based calibration of multimodal sensor arrays”, in *IEEE International Conference on Robotics and Automation*, 2015, pp. 4843–4850.

[66] T. Schneider, M. Li, C. Cadena, J. Nieto, and R. Siegwart, “Observability-Aware Self-Calibration of Visual and Inertial Sensors for Ego-Motion Estimation”, *IEEE Sensors Journal*, vol. 19, no. 10, pp. 3846–3860, 2019.

[67] V. Wüest, V. Kumar, and G. Loianno, “Online Estimation of Geometric and Inertia Parameters for Multirotor Aerial Vehicles”, in *2019 International Conference on Robotics and Automation (ICRA)*, 2019, pp. 1884–1890.

[68] J. Svacha, J. Paulos, G. Loianno, and V. Kumar, “IMU-Based Inertia Estimation for a Quadrotor Using Newton-Euler Dynamics”, *IEEE Robotics and Automation Letters*, vol. 5, no. 3, pp. 3861–3867, 2020.

[69] M. Dhaybi and N. Daher, “Accurate Real-time Estimation of the Inertia Tensor of Package Delivery Quadrotors*”, in *2020 American Control Conference (ACC)*, 2020, pp. 1520–1525.

[70] M. Burri, M. Bloesch, Z. Taylor, R. Siegwart, and J. Nieto, “A framework for maximum likelihood parameter identification applied on MAVs”, *Journal of Field Robotics*, vol. 35, no. 1, pp. 5–22, 2018.

[71] M. Tonolla, S. Peduzzi, A. Demarta, R. Peduzzi, and D. Hahn, “Phototropic sulfur and sulfate-reducing bacteria in the chemocline of meromictic Lake Cadagno, Switzerland”, *Journal of Limnology*, vol. 63, no. 2, pp. 161–170, 2004.

[72] T Sommer, F Danza, J Berg, A Sengupta, G Constantinescu, T Tokyay, H Bürgmann, Y Dressler, O. Sepúlveda Steiner, C. Schubert, *et al.*, “Bacteria-induced mixing in natural waters”, *Geophysical Research Letters*, vol. 44, no. 18, pp. 9424–9432, 2017.

[73] T. Posch, O. Köster, M. M. Salcher, and J. Pernthaler, “Harmful filamentous cyanobacteria favoured by reduced water turnover with lake warming”, *Nature Climate Change*, vol. 2, no. 11, pp. 809–813, 2012.

[74] B. F. Castro, O. S. Steiner, D. Knapp, T. Posch, D. Bouffard, and A. Wüest, “Inhibited vertical mixing and seasonal persistence of a thin cyanobacterial layer in a stratified lake”, *Aquatic Sciences*, vol. 83, no. 2, pp. 1–22, 2021.

[75] *MicroRider – Self-Contained Turbulence Instrument Package – Rockland Scientific*, rocklandscientific.com/products/modular-systems/microrider/, Accessed: 2021-06-15.

[76] F. Dellaert, “Factor graphs and GTSAM: A hands-on introduction”, Georgia Institute of Technology, Tech. Rep., 2012.

[77] J. A. Catipovic, “Performance limitations in underwater acoustic telemetry”, *IEEE Journal of Oceanic Engineering*, vol. 15, no. 3, pp. 205–216, 1990.

[78] M. Stojanovic, J. A. Catipovic, and J. G. Proakis, “Phase-coherent digital communications for underwater acoustic channels”, *IEEE Journal of Oceanic Engineering*, vol. 19, no. 1, pp. 100–111, 1994.

[79] A. Tadayon and M. Stojanovic, “Low-Complexity Superresolution Frequency Offset Estimation for High Data Rate Acoustic OFDM Systems”, *IEEE Journal of Oceanic Engineering*, pp. 1–11, 2018.

[80] M. Tomlinson, C. J. Tjhai, M. A. Ambroze, M. Ahmed, and M. Jibril, *Error-Correction Coding and Decoding*. Springer, 2017.

[81] S. Kemna, J. G. Rogers, C. Nieto-Granda, S. Young, and G. S. Sukhatme, “Multi-robot coordination through dynamic Voronoi partitioning for informative adaptive sampling in communication-constrained environments”, in *IEEE International Conference on Robotics and Automation*, 2017, pp. 2124–2130.

[82] *DiveNET Commander Ultra-Short Base Line (USBL) Positioning and Navigation*, https://www.divenetgps.com/commander.

[83] M. M. M. Manhães, S. A. Scherer, M. Voss, L. R. Douat, and T. Rauschenbach, "UUV Simulator: A Gazebo-based package for underwater intervention and multi-robot simulation", in *OCEANS 2016 MTS/IEEE Monterey*, 2016, pp. 1–8.

[84] E. H. Henriksen, I. Schjølberg, and T. B. Gjersvik, "UW MORSE: The underwater Modular Open Robot Simulation Engine", in *2016 IEEE/OES Autonomous Underwater Vehicles (AUV)*, 2016, pp. 261–267.

[85] H. M. Schmidt-Didlaukies, A. J. Sørensen, and K. Y. Pettersen, "Modeling of Articulated Underwater Robots for Simulation and Control", in *2018 IEEE/OES Autonomous Underwater Vehicle Workshop (AUV)*, 2018, pp. 1–7.

[86] S. Choi, S. Menor, and J. Yuh, "Distributed virtual environment collaborative simulator for underwater robots", in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 2, 2000, pp. 861–866.

[87] G. Xie, J. Gibson, and L. Diaz-Gonzalez, "Incorporating Realistic Acoustic Propagation Models in Simulation of Underwater Acoustic Networks: A Statistical Approach", in *OCEANS 2006*, 2006, pp. 1–9.

[88] A. Essebbar and V. Vercelloni, "Simulation of communication system for underwater acoustics", in *OCEANS '95 MTS/IEEE*, vol. 2, 1995, pp. 1204–1207.

[89] A. Sehgal, D. Cernea, and A. Birk, "Simulating Underwater Acoustic Communications in a High Fidelity Robotics Simulator", *7th IFAC Symposium on Intelligent Autonomous Vehicles*, vol. 43, no. 16, pp. 587–592, 2010.

[90] A. Bahr, J. J. Leonard, and M. F. Fallon, "Cooperative Localization for Autonomous Underwater Vehicles", *The International Journal of Robotics Research*, vol. 28, no. 6, pp. 714–728, 2009.

[91] J. Borenstein, "Experimental results from internal odometry error correction with the OmniMate mobile robot", *IEEE Transactions on Robotics and Automation*, vol. 14, no. 6, pp. 963–969, 1998.

[92] S. I. Roumeliotis and G. A. Bekey, "Distributed multirobot localization", *IEEE Transactions on Robotics and Automation*, vol. 18, no. 5, pp. 781–795, 2002.

[93] L. Luft, T. Schubert, S. I. Roumeliotis, and W. Burgard, "Recursive decentralized localization for multi-robot systems with asynchronous pairwise communication", *The International Journal of Robotics Research*, vol. 37, no. 10, pp. 1152–1167, 2018.

[94] S. Panzieri, F. Pascucci, and R. Setola, "Multirobot Localisation Using Interlaced Extended Kalman Filter", in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2006, pp. 2816–2821.

[95] S. J. Julier and J. K. Uhlmann, "A non-divergent estimation algorithm in the presence of unknown correlations", in *Proceedings of the 1997 American Control Conference (Cat. No.97CH36041)*, vol. 4, 1997, 2369–2373 vol.4.

[96] A. Howard, M. J. Matarić, and G. S. Sukhatme, “Localization for mobile robot teams using maximum likelihood estimation”, in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 1, 2002, pp. 434–439.

[97] A. Howard, M. J. Matarić, and G. S. Sukhatme, “Localization for Mobile Robot Teams: A Distributed MLE Approach”, in *Experimental Robotics VIII*, B. Siciliano and P. Dario, Eds., Springer Berlin Heidelberg, 2003, pp. 146–155.

[98] J. Klingner, N. Ahmed, and N. Correll, “Fault-tolerant covariance intersection for localizing robot swarms”, *Robotics and Autonomous Systems*, vol. 122, pp. 103–306, 2019.

[99] L. C. Carrillo-Arce, E. D. Nerurkar, J. L. Gordillo, and S. I. Roumeliotis, “Decentralized multi-robot cooperative localization using covariance intersection”, in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2013, pp. 1412–1417.

[100] M. Vasic, D. Mansolino, and A. Martinoli, “A system implementation and evaluation of a cooperative fusion and tracking algorithm based on a Gaussian Mixture PHD filter”, in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2016, pp. 4172–4179.

[101] H. Li, F. Nashashibi, and M. Yang, “Split Covariance Intersection Filter: Theory and Its Application to Vehicle Localization”, *IEEE Transactions on Intelligent Transportation Systems*, vol. 14, no. 4, pp. 1860–1871, 2013.

[102] H. Li and F. Nashashibi, “Cooperative Multi-Vehicle Localization Using Split Covariance Intersection Filter”, *IEEE Intelligent Transportation Systems Magazine*, vol. 5, no. 2, pp. 33–44, 2013.

[103] J. Park and J. Lee, “Beacon selection and calibration for the efficient localization of a mobile robot”, 1, vol. 32, Cambridge University Press, 2014, pp. 115–131.

[104] H. Wang, K. Yao, G. Pottie, and D. Estrin, “Entropy-Based Sensor Selection Heuristic for Target Localization”, in *Proceedings of the 3rd International Symposium on Information Processing in Sensor Networks*, Berkeley, California, USA, 2004, pp. 36–45.

[105] E. Ertin, J. W. Fisher, and L. C. Potter, “Maximum Mutual Information Principle for Dynamic Sensor Query Problems”, in *Information Processing in Sensor Networks*, Springer Berlin Heidelberg, 2003, pp. 405–416.

[106] Y. Guo, Q. Han, and X. Kang, “Underwater sensor networks localization based on mobility-constrained beacon”, *Wireless Networks*, vol. 26, no. 4, pp. 2585–2594, 2020.

[107] E. M. Fischell, N. R. Rypkema, and H. Schmidt, “Relative Autonomy and Navigation for Command and Control of Low-Cost Autonomous Underwater Vehicles”, *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 1800–1806, 2019.

[108] E. Erdemir and T. E. Tuncer, “Path planning for mobile-anchor based wireless sensor network localization: Static and dynamic schemes”, *Ad Hoc Networks*, vol. 77, pp. 1 –10, 2018.

[109] E. Erdemir and T. E. Tuncer, “Path planning and localization for mobile anchor based wireless sensor networks”, in *25th European Signal Processing Conference (EUSIPCO)*, 2017, pp. 131–135.

[110] J. Rezazadeh, M. Moradi, A. S. Ismail, and E. Dutkiewicz, “Superior Path Planning Mechanism for Mobile Beacon-Assisted Localization in Wireless Sensor Networks”, *IEEE Sensors Journal*, vol. 14, no. 9, pp. 3052–3064, 2014.

[111] G. Han, H. Xu, J. Jiang, L. Shu, T. Hara, and S. Nishio, “Path planning using a mobile anchor node based on trilateration in wireless sensor networks”, *Wireless Communications and Mobile Computing*, vol. 13, no. 14, pp. 1324–1336, 2013.

[112] L. Pursell and S. Y. Trimble, “Gram-Schmidt Orthogonalization by Gauss Elimination”, *The American Mathematical Monthly*, vol. 98, no. 6, pp. 544–549, 1991.

[113] D. Simon and D. L. Simon, “Constrained Kalman filtering via density function truncation for turbofan engine health estimation”, *International Journal of Systems Science*, vol. 41, no. 2, pp. 159–171, 2010.

[114] T. M. Cover and J. A. Thomas, *Elements of information theory*. John Wiley & Sons, Inc, 1991.

[115] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan, *Estimation with applications to tracking and navigation: theory algorithms and software*. John Wiley & Sons, 2004.

[116] B. Zhang and G. S. Sukhatme, “Adaptive Sampling for Estimating a Scalar Field using a Robotic Boat and a Sensor Network”, in *IEEE International Conference on Robotics and Automation (ICRA)*, 2007, pp. 3673–3680.

[117] J. M. Soares, A. P. Aguiar, A. M. Pascoal, and A. Martinoli, “An algorithm for formation-based chemical plume tracing using robotic marine vehicles”, in *OCEANS 2016 MTS/IEEE Monterey*, 2016, pp. 1–8.

[118] W. Luo and K. Sycara, “Adaptive Sampling and Online Learning in Multi-Robot Sensor Coverage with Mixture of Gaussian Processes”, in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 2018, pp. 6359–6364.

[119] A. Krause, A. Singh, and C. Guestrin, “Near-Optimal Sensor Placements in Gaussian Processes: Theory, Efficient Algorithms and Empirical Studies”, *Journal of Machine Learning Research*, vol. 9, no. 8, pp. 235–284, 2008.

[120] A. Marino, G. Antonelli, A. P. Aguiar, A. Pascoal, and S. Chiaverini, “A Decentralized Strategy for Multirobot Sampling/Patrolling: Theory and Experiments”, *IEEE Transactions on Control Systems Technology*, vol. 23, no. 1, pp. 313–322, 2015.

[121] K.-C. Ma, L. Liu, and G. S. Sukhatme, “An information-driven and disturbance-aware planning method for long-term ocean monitoring”, in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2016, pp. 2102–2108.

[122] C. Denniston, A. Kumaraguru, and G. S. Sukhatme, “Comparison of Path Planning Approaches for Harmful Algal Bloom Monitoring”, in *MTS/IEEE OCEANS*, 2019.

[123] K. H. Low, J. Dolan, J Schneider, and A Elfes, “Multi-robot adaptive exploration and mapping for environmental sensing applications”, *PhD thesis, CMU*, 2009.

[124] S. Kemna, D. A. Caron, and G. S. Sukhatme, “Adaptive informative sampling with autonomous underwater vehicles: Acoustic versus surface communications”, in *MTS/IEEE OCEANS 2016 Monterey*, 2016.

[125] B. Dennis and G. Patil, “Applications in ecology”, in *Lognormal distributions: theory and applications*, 2018, pp. 303–330.

[126] J. C. Loftis, R. C. Ward, and G. M. Smillie, “Statistical Models for Water Quality Regulation”, *Journal (Water Pollution Control Federation)*, vol. 55, no. 8, pp. 1098–1104, 1983.

[127] J. W. Campbell, “The lognormal distribution as a model for bio-optical variability in the sea”, *Journal of Geophysical Research: Oceans*, vol. 100, no. C7, pp. 13 237–13 254, 1995.

[128] D. Siegel, M. Behrenfeld, S. Maritorena, C. McClain, D. Antoine, S. Bailey, P. Bontempi, E. Boss, H. Dierssen, S. Doney, R. Eplee, R. Evans, G. Feldman, E. Fields, B. Franz, N. Kuring, C. Mengelt, N. Nelson, F. Patt, W. Robinson, J. Sarmiento, C. Swan, P. Werdell, T. Westberry, J. Wilding, and J. Yoder, “Regional to global assessments of phytoplankton dynamics from the SeaWiFS mission”, *Remote Sensing of Environment*, vol. 135, pp. 77–91, 2013.

[129] D. Skeen, “A quorum-based commit protocol”, in *Berkeley Workshop on Distributed Data Management and Computer Networks*, 1982, 69–80.

[130] J. Quintas, R. Petrocciay, A. Pascoal, J. Cruz, P. Gois, L. Morlandoy, and M. Stipano, “Hybrid acoustic-optical underwater communication networks for next-generation cooperative systems: the EUMR experience”, in *MTS/IEEE OCEANS*, 2021, To appear.

[131] H. Choi, L. Brunet, and J. P. How, “Consensus-Based Decentralized Auctions for Robust Task Allocation”, *IEEE Transactions on Robotics*, vol. 25, no. 4, pp. 912–926, 2009.

[132] M. Alighanbari and J. P. How, “Decentralized Task Assignment for Unmanned Aerial Vehicles”, in *Proceedings of the 44th IEEE Conference on Decision and Control*, 2005, pp. 5668–5673.

[133] T. Cieslewski, S. Lynen, M. Dymczyk, S. Magnenat, and R. Siegwart, “Map API - Scalable Decentralized Map Building for Robots”, in *IEEE International Conference on Robotics and Automation*, 2015, pp. 6241–6247.

[134] D. Ongaro and J. Ousterhout, “In Search of an Understandable Consensus Algorithm”, in *USENIX ATC*, 2014.

[135] Lamport, Leslie and others, “Paxos made simple”, *ACM Sigact News*, vol. 32, no. 4, pp. 18–25, 2001.

Curriculum Vitae

Anwar Ahmad Quraishi

Education

2016–2021	Ph.D. in Robotics, Control, and Intelligent Systems *Ecole Polytechnique Fédérale de Lausanne (EPFL), Switzerland*
2013–2015	M.Sc. in Microengineering, specialization in Robotics *Ecole Polytechnique Fédérale de Lausanne (EPFL), Switzerland*
2008–2012	Bachelor of Technology in Mechanical Engineering *Indian Institute of Technology Hyderabad, India*

Work Experience

2014–2015	Software Engineering Intern *Cyberbotics Ltd., Lausanne, Switzerland*
2012–2013	Senior Software Engineer *Polycom India, Hyderabad, India*

Publications

1. **A. Quraishi**, A. Martinoli, “Online Kinematic and Dynamic Parameter Estimation for Autonomous Surface and Underwater Vehicles,” in IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Prague, Czech Republic (Online), 2021. To appear.

2. **A. Quraishi**, A. Martinoli, “Coordinated Path Planning for Surface Acoustic Beacons for Supporting Underwater Localization,” in IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Prague, Czech Republic (Online), 2021. To appear.

3. **A. Quraishi**, A. Martinoli, “Distributed Cooperative Localization with Efficient Pairwise Range Measurements,” in International Symposium on Distributed Autonomous Robotic Systems (DARS), Kyoto, Japan (Online), 2021. To appear.

4. **A. Quraishi**, A. Bahr, F. Schill, A. Martinoli, “A Flexible Navigation Support System for a Team of Underwater Robots,” in IEEE International Symposium on Multi-Robot and Multi-Agent Systems (MRS), New Brunswick, NJ, USA, 2019, pp. 70–75.

5. **A. Quraishi**, A. Bahr, F. Schill, A. Martinoli, “Easily Deployable Underwater Acoustic Navigation System for Multi-Vehicle Environmental Sampling Applications,” in IEEE International Conference on Robotics and Automation (ICRA), Montreal, Canada, 2019, pp. 3464–3470.

6. **A. Quraishi**, A. Bahr, F. Schill, A. Martinoli, “Autonomous Feature Tracing and Adaptive Sampling in Real-World Underwater Environments,” in IEEE International Conference on Robotics and Automation (ICRA), Brisbane, Australia, 2018, pp. 5699–5704.
7. **A. Quraishi**, T. Cieslewski, S. Lynen, R. Siegwart, “Robustness to Connectivity Loss for Collaborative Mapping,” in IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Daejeon, Korea, 2016, pp. 4580–4585.

Project Supervision

1. Kamyar Taher, Semester Project (Spring 2020)
   AUV Localization as an Optimization Problem
2. Hannes Kaspar Rovina, External Master Project at University of Pennsylvania (Spring 2019), Co-supervised with Prof. Ani Hsieh
   Control and Coordination of Heterogeneous Mobile Robot Teams Operating in Dynamic and Complex Environments
3. Darko Lukic, Internship (Summer, 2019)
   Underwater Acoustic Ranging and Communication
4. Hannes Kaspar Rovina (Spring 2019)
   Cooperative Localization for a Swarm of AUVs
5. Paul Callens, Semester Project (Fall 2018)
   Underwater Localization with Acoustic Ranging
6. Darko Lukic, Semester Project (Fall 2018)
   Underwater Acoustic Communication
7. Fabian Schultz, Semester Project (Spring 2018)
   Cooperative Localization for a Swarm of AUVs
8. Mehdi Nejjar, Semester Project (Fall 2017)
   Framework for Underwater Positioning with Acoustic Beacons
9. Ahmed Saadalah, Internship (Summer, 2017)
   Framework for Localization of AUVs based on Range Measurements
10. Jean-François Burnier, Semester Project (Spring 2017)
    Sigma-Point Kalman Filtering for State Estimation
11. Ahmed Saadallah, Semester Project (Fall 2016)
    Graphical Interface for Simultaneous Interaction with Multiple Robots
12. Alberto Arrighi, Semester Project (Spring 2016), Co-supervised with Dr. Felix Schill
    Formation Control of Autonomous Boats

Languages

Hindi/Urdu	native
English	fluent
French	elementary

Personal Details

Date of birth:	18 May 1990
Place of birth:	Hyderabad, India
Citizenship:	India
Email:	anwarahmadq@gmail.com