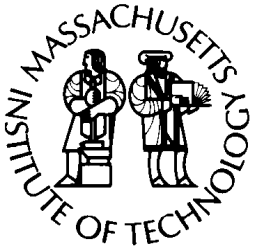
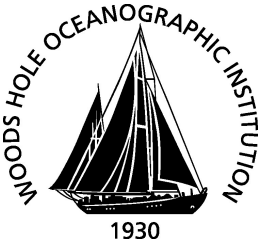
MIT/WHOI 2009-05

Massachusetts Institute of Technology
Woods Hole Oceanographic Institution



Joint Program in Oceanography/ Applied Ocean Science and Engineering



DOCTORAL DISSERTATION

Cooperative Localization for Autonomous Underwater Vehicles

by

Alexander Bahr

February 2009

MIT/WHOI

2009-05

Cooperative Localization for Autonomous Underwater Vehicles

by

Alexander Bahr

Massachusetts Institute of Technology
Cambridge, Massachusetts 02139

and

Woods Hole Oceanographic Institution
Woods Hole, Massachusetts 02543

February 2009

DOCTORAL DISSERTATION

Funding was provided by the Office of Naval Research, ASAP MURI program and the Woods Hole Oceanographic Institution Academic Programs Office.

Approved for publication; distribution unlimited.

Approved for Distribution:

James F. Lynch, Chair
Department of Applied Ocean Physics and Engineering

Paola Malanotte-Rizzoli
MIT Director of Joint Program

James A. Yoder
WHOI Dean of Graduate Studies

Cooperative Localization for Autonomous Underwater Vehicles

by

Alexander Bahr

Diploma, Electrical Engineering, Technical University Aachen, Germany, 2002

Submitted to the Joint Program in Applied Ocean Science & Engineering
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

and the

WOODS HOLE OCEANOGRAPHIC INSTITUTION

February 2009

Author .....
Joint Program in Applied Ocean Science & Engineering
Massachusetts Institute of Technology
and Woods Hole Oceanographic Institution
22 December 2008

Certified by .....
John J. Leonard
Professor of Mechanical and Ocean Engineering
Massachusetts Institute of Technology
Thesis Supervisor

Accepted by .....
James C. Preisig
Chairman, Joint Committee for Applied Ocean Science & Engineering
Massachusetts Institute of Technology
Woods Hole Oceanographic Institution

Accepted by .....
David E. Hardt
Professor of Mechanical Engineering
Chairman, Committee for Graduate Students
Massachusetts Institute of Technology

Cooperative Localization for Autonomous Underwater Vehicles

by

Alexander Bahr

Submitted to the Joint Program in Applied Ocean Science & Engineering
on 22 December 2008, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy

Abstract

Self-localization of an underwater vehicle is particularly challenging due to the absence of Global Positioning System (GPS) reception or features at known positions that could otherwise have been used for position computation. Thus Autonomous Underwater Vehicle (AUV) applications typically require the pre-deployment of a set of beacons.

This thesis examines the scenario in which the members of a group of AUVs exchange navigation information with one another so as to improve their individual position estimates.

We describe how the underwater environment poses unique challenges to vehicle navigation not encountered in other environments in which robots operate and how cooperation can improve the performance of self-localization. As intra-vehicle communication is crucial to cooperation, we also address the constraints of the communication channel and the effect that these constraints have on the design of cooperation strategies.

The classical approaches to underwater self-localization of a single vehicle, as well as more recently developed techniques are presented. We then examine how methods used for cooperating land-vehicles can be transferred to the underwater domain. An algorithm for distributed self-localization, which is designed to take the specific characteristics of the environment into account, is proposed.

We also address how correlated position estimates of cooperating vehicles can lead to overconfidence in individual position estimates.

Finally, key to any successful cooperative navigation strategy is the incorporation of the relative positioning between vehicles. The performance of localization algorithms with different geometries is analyzed and a distributed algorithm for the dynamic positioning of vehicles, which serve as dedicated navigation beacons for a fleet of AUVs, is proposed.

Thesis Supervisor: John J. Leonard
Title: Professor of Mechanical and Ocean Engineering
Massachusetts Institute of Technology

Acknowledgments

This thesis would not have been possible without the help and support of many friends and colleagues who made the last five years at MIT an exceptionally fulfilling experience.

I would like to thank my advisor John Leonard who strongly supported me ever since our first email exchange in 2003. He guided me all the way through the application process, research, thesis and finding a post doc position. His broad range of research interests enabled me to find a thesis subject that exactly matched my interest. And I very much appreciated him allowing me to take a significant amount of time to pursue other projects as well as travel. I would also like to thank my other committee members: Henrik Schmidt for his continued support and great company during several research cruises, and Hanu Singh and Arjuna Balasuriya for their helpful suggestions during my thesis writing.

Many thanks to David Battle who helped me through my first steps with Autonomous Underwater Vehicles and was a great mate to have around. Many of the experiments presented in this thesis would not have been possible without the support of Andrew Patrikalakis. The results owe a lot to his countless hours of coding assistance and his efforts to ensure that the kayaks would be ready when needed. They were also made possible by Joe Curcio, the builder of the kayaks, and the support of Jacques Leedekerken and Kevin Cockrell.

The last years would not have been the same without the many great people I met at MIT. Most important was Matt Walter who convinced me in August 2003 that MIT is not only an interesting place, but that it can be very friendly as well. Throughout the years we shared many great personal and academic experiences. I wish him all the best, wherever life may take him. Patrycja unfortunately left our lab, but made up for it by taking me on a very memorable trip across the country. I wish Alec, Emma, Albert, Tom, Olivier and Aisha all the best for their remaining time and life beyond. Iuliu was always a welcome distraction in the United States and abroad and a great help with all hardware questions. I was glad to join Carrick, Marty and David on a number of exciting research and conference trips, as well as their advisor Daniela Rus. In the last months I was also very fortunate to meet a new group of people. First, I am very thankful to Maurice for carrying on what I started - I cannot imagine somebody better suited for it - and also Been, Georgios, Hordur and Rob. I would also like to thank the many people of the SEA 2007 cruise, particularly the B watch and Julian, Jamie, Heather, Chris, Toby and Jane.

One of the most exciting things during my time at MIT was that I was not only able to pursue my thesis topic but also two other projects. First, the flood warning project introduced me to Elizabeth Basha. We shared many joyful moments as well as blood, sweat and tears in the Central American wilderness. I hope that the end of my PhD only marks the beginning of that partnership. Second, the harbor porpoise tag project led by Stacy DeRuiter was a great design challenge. It also provided an opportunity to reach into other areas of ocean sciences by contributing to marine biology research. Her dedication along with the support from Mark Johnson, Peter Tyack and Tom Hurst ensured the project's success. The exciting results and the

process leading up to them rewarded me with a better experience at MIT than I could have ever hoped for. I would like to thank John Leonard for letting me take this scenic route.

The path that led me to MIT would not have been possible without the support from people in the early stages of my engineering career who I would like to thank here: Raimund Eich for patiently answering my first electrical engineering questions; my best friends Jan Horn, Daniel Steffensky, Alexander Zimmer and Ulf Radenz for helping me through my university time in Germany; and John Peatman, Ludger Becks and Uwe Zimmer for their academic guidance.

Finally, I would like to thank my parents for their continued support and especially Melissa Pitotti for her encouragement not only to start the work at this institution but also to finish it when the time had come.

This work was funded by Office of Naval Research grants N00014-97-1-0202, N00014-05-1-0255, N00014-02-C-0210, N00014-07-1-1102 and the ASAP MURI program led by Naomi Leonard of Princeton University.

"One degree is not a large distance. On a compass it is scarcely the thickness of a fingernail. But in certain conditions, one degree can be a very large distance. Enough to unmake a man."

The Mysterious Geographic Explorations of Jasper Morello, © 3D Films, Australia 2005

Meinen Eltern & meinem Bruder

Contents

List of Figures

List of Tables

List of Algorithms

List of Acronyms

AHRS	Attitude Heading Rate Sensor
ASC	Autonomous Surface Craft
ASW	Anti-Submarine Warfare
AUV	Autonomous Underwater Vehicle
CEP	Circular Error Probable
CN	Cooperative Navigation
CNA	Communication and Navigation Aid
DIDSON	Dual Frequency Identification Sonar
DR	Dead-Reckoning
DSP	Digital Signal Processor
DVL	Doppler Velocity Log
EKF	Extended Kalman Filter
FOG	Fiber-Optic Gyroscope
GDOP	geometric dilution of precision
GMM	Gaussian Mixture Model
GPS	Global Positioning System
HAUV	Hovering Autonomous Underwater Vehicle
INS	Inertial Navigation System
IVP	Interval Programming
IU	Interleaved Update
KF	Kalman Filter

LBL	Long Baseline
LNG	Liquefied Natural Gas
LF	Low Frequency
LSQ	Least Squares
MER	Mars Exploration Rover
MIT	Massachusetts Institute of Technology
MLBL	Moving Long Baseline
MMSE	Minimum Mean Squared Error
MOOS	Mission Oriented Operating Suite
MVC	Main Vehicle Computer
NEES	Normalized Estimation Error Squared
OWTT	One-Way Travel Time
PF	Particle Filter
PPS	Pulse Per Second
ROV	Remotely Operated Vehicle
RF	Radio-Frequency
SBL	Short Baseline
SCM	Search, Classify and Map
SLAM	Simultaneous Localization and Mapping
SLF	Super Low Frequency
SNR	Signal-to-Noise ratio
TAT	Turn-Around Time
TOF	Time-of-Flight
UAV	Unmanned Aerial Vehicle
UUV	Unmanned Underwater Vehicles
UHF	Ultra-High Frequency
USBL	Ultra-Short Baseline

VAN	Visually Augmented Navigation
VLF	Very Low Frequency
WHOI	Woods Hole Oceanographic Institution

Chapter 1

Introduction

A new wave of robots has led to exciting scientific findings over the past decade. Long after *static* manufacturing robots matured to become useful tools and common sights on most factory work floors, highly *mobile* exploration robots reach places which were previously hard or impossible to access by humans. They provide a platform for their onboard sensors which collect data in an environment which has never been previously visited.

Perhaps the best known examples of exploration robots are the two *Mars Exploration Rover (MER)* vehicles that landed on Mars in 2003 and the earlier *Pathfinder* mission vehicle *Sojourner*. Before *Sojourner* landed on Mars in 1997, a large number of static space probes had been sent to several planets in the solar system since the seventies, but *Pathfinder*'s mobility greatly increased the number of possible applications. From *Sojourner* to *MER* the mobility of the planetary exploration rovers was increased by three orders of magnitude from tens of meters to tens of kilometers within six years. *MER* also demonstrated the improvements made in reliability with the robots continuing to provide scientific data since 2003, after more than five years of operation.

Another environment in which autonomous exploration robots have operated very successfully that has gained much less attention is in the bodies of water covering 70 % of Earth's surface. The dive of the *Trieste*, a manned submersible, reaching the *Challenger Depth*[1] in 1960, demonstrated the potential to create vehicles capable of reaching every spot in the ocean. The amount of ocean floor explored by this vehicle however was only a few square meters and no vehicle, manned or unmanned, has returned to these depths since. Even today's much more sophisticated craft are not able to explore farther than a few kilometers beyond the vehicle's position due to the strong attenuation of electromagnetic and acoustic waves in the water body which limits the footprint of the vehicles' sensors. For example a single space probe, *Mars Express*, was able, in just a few years, to obtain a much more detailed image of the Martian surface, than a century of ocean exploration was able to create for the Earth's seafloor. While sensors for the ocean environment are continuously improving and satellites can measure surface features such as temperature and wave height, it

[1] the deepest point in the ocean at 10916 m [55]

Figure 1-1: *MER* and *Seabed*, two exploration robots in a hostile environment. Left: Artist impression of *MER* on Mars. Courtesy NASA/JPL-Caltech. Right: *Seabed* deployed during the Gakkel Ridge Expedition in July 2008. Photo courtesy of Hanu Singh.

remains difficult for the foreseeable future to see through the water column.

The Earth's oceans affect all of Earth's inhabitants as they strongly influence the climate and provide much needed resources (e.g. seafood as well as oil and minerals). The influence on the climate and the effect of the ocean's biomass are not well understood, mostly due to the lack of available data. Classical ocean exploration mostly relies on immobile buoys and manned surface and underwater vehicles. As a result of the high cost and the inherent danger to people's lives, the number of people and vessels deployed for ocean research world-wide is relatively small. The small footprint of ocean sensors, combined with the low number of vehicles carrying them, limits the speed at which the water body can be explored.

With the footprint of each vehicle limited to a small area around its position, the volume of ocean surveyed is proportional to the number of vehicles deployed. The only option to strongly increase the spatial and temporal sample density is to deploy a much larger number of vehicles. The high cost of manned vehicles prohibits a large increase in their use, but autonomous underwater and surface robots provide an attractive alternative. They only require a small amount of personnel during deployment and recovery and their fuel and maintenance costs are marginal when compared to those of conventional research vessels.

Having a good estimate of one's location is critical for manned and unmanned vehicles, but it is particularly important for autonomous vehicles as unsupervised decisions are made based on the location estimate. Furthermore the quality of the data collected is directly dependent on how well measurements can be referenced to

a geographic location. The underwater environment makes determining a vehicle's position particularly difficult, as we will show in this thesis, requiring new strategies for navigation. One such strategy, which we propose, is the concept of Cooperative Navigation (CN) in which a group of autonomous vehicles exchanges navigation information in order to improve the group's overall position estimate. This thesis investigates the challenges involved in implementing this concept and proposes solutions.

1.1 Autonomous Marine Vehicles

In the last 20 years, research in autonomous marine platforms has led to a large and ever growing number of different submarine vehicles. Autonomous Surface Crafts (ASCs) have only recently received more attention and underwater platforms continue to dominate the research. This thesis will show examples of how a joint deployment of ASCs and Autonomous Underwater Vehicles (AUVs) can be beneficial, with a focus on the underwater domain. The following sections will give an overview of the various classes of underwater vehicles and their specific characteristics. We will also illustrate some of the many applications for which AUVs are used today and how these applications shaped their characteristics. The type of AUV determines the accuracy of its on-board navigation sensors, while the application dictates the required localization accuracy.

1.1.1 Platforms

Propelled Vehicles

The earliest and still most common type of AUV consist of a torpedo shaped body with a single thruster. The diameter of larger models (*Bluefin21*) is often 533 mm (21") [17], the same as that of a heavy-weight torpedo while the smaller ones (*Remus 600*, *Bluefin9*) have the same diameter as a light-weight torpedo - 324 mm (12.75"). Their lengths vary between 1 m and 7 m and several models consist of segments which can be added or removed to adapt the payload section to the mission's specific needs. Note the transitions between the modules on the *Bluefin21* and *Gavia* in figure 1-2. The weight ranges between 20 kg for the very small and portable *IVER* (top left in figure 1-2) and 900 kg for a large *Remus 6000*, but all vehicles are usually positively buoyant. This is a security feature which ensures that a vehicle will return to the surface if all power is cut. As a result this type of AUV needs to move forward to stay submerged. The maximum achievable depth is 6000 m for the large vehicles and around 100 m for the smaller ones. Cruising speeds range between 1-3 m/s. The excellent hydrodynamic properties of the long slender body only require a relatively low-power propulsion for a given cruising speed and payload. As a result, variations of the torpedo-shaped AUV, such as the ones in figure 1-2 are the vehicle of choice for many applications. Early AUVs were usually large, but with recent advances in

Figure 1-2: Various torpedo-shaped AUVs. Top left: the low-cost, "human-portable" *IVER* from Ocean Server. Top right: two *Bluefin21* prior to being loaded onto the Leonardo during the GOATS04 experiment. Bottom row: the *Gavia* AUV made by Hafmynd.

miniaturization of the key components (battery, propulsion, navigation sensors) several smaller, "human-portable", models appeared recently. Other, more specialized vehicles exist such as WHOI's *Seabed* [79] shown in figure 1-4. This vehicle is specifically designed for underwater imaging. To accomplish this, it needs to be able to precisely control its altitude and to change its depth without forward motion. The shape of this vehicle, with a large separation between the center of gravity and the center of buoyancy, provides a very stable camera platform. An even more maneuverable AUV is the *HAUV* [91] shown in figure 1-5. Its eight thrusters allow it to rotate around and move along every axis in 3D-space. Its main application is ship hull inspection using a forward looking imaging sonar (figure 1-5). Another specialized design is Woods Hole Oceanographic Institution (WHOI)'s hybrid ROV/AUV *Nereus* which will be able to reach full ocean depth [18].

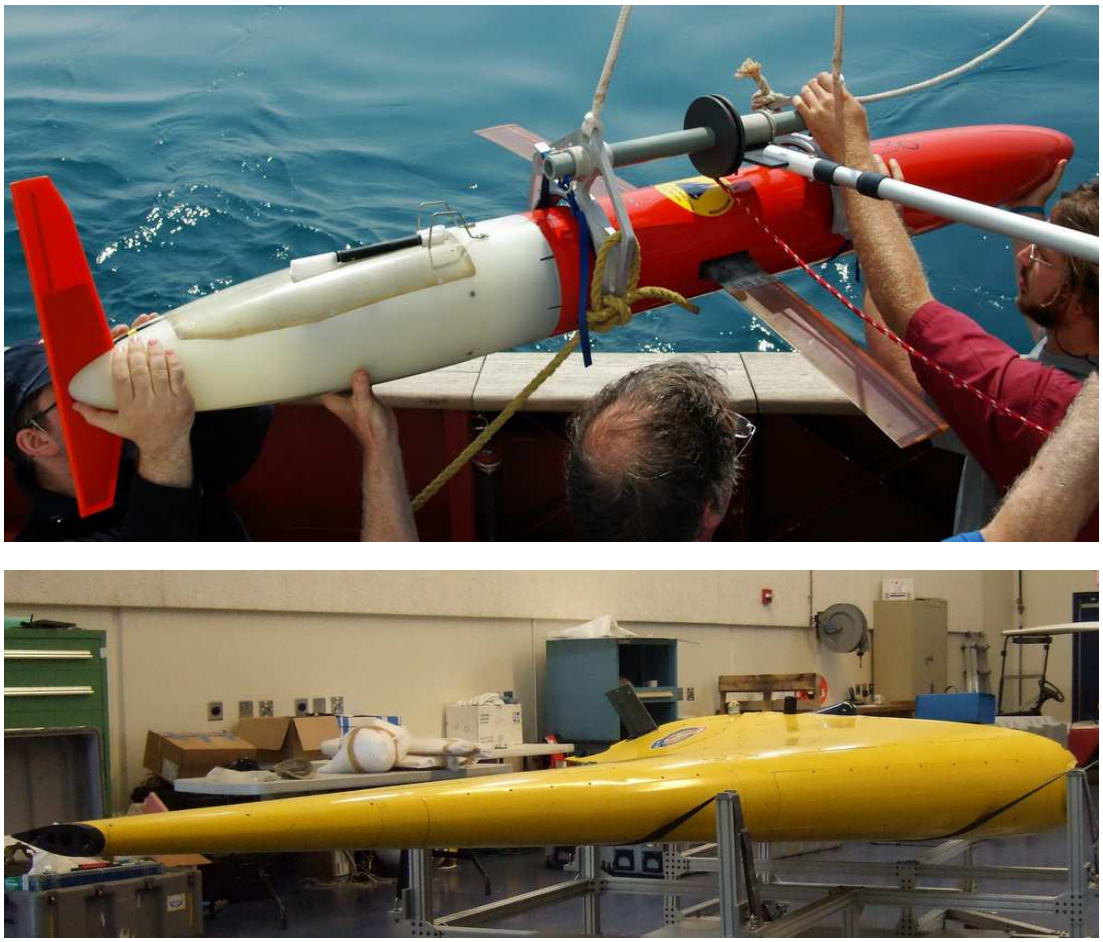


Figure 1-3: The *Spray* (top) and the *X-Ray* glider (bottom).

Buoyancy-Driven Vehicles

As the high power consumption of the propulsion system limits the range of propelled AUVs to a few 100 km [40] the need for vehicles capable of crossing an ocean led to the development of buoyancy driven gliders. All gliders such as the *Spray* or the *X-Ray*, both shown in figure 1-3, are capable of changing their displaced volume to become positively or negatively buoyant by pumping oil from an internal reservoir to an outside bladder. As a result of the buoyancy change, the glider ascends or descends within the water column. A set of "wings" then adds a forward component to the otherwise purely vertical motion. The glider performs a sawtooth pattern (figure 5-5b) which can take it to depths of more than 2000 m. The internal battery pack can be shifted along the longitudinal axis to provide pitch control as well as rolled around the longitudinal axis to provide yaw control in conjunction with a set of vertical fins. A detailed description of *Seaglider* can be found in [30]. The propulsion system only consumes power during the activation of the pump on the sea surface or near the bottom. This type of propulsion is very efficient, requiring only 0.5 W [30] averaged over the entire period of operation, and enables transects of

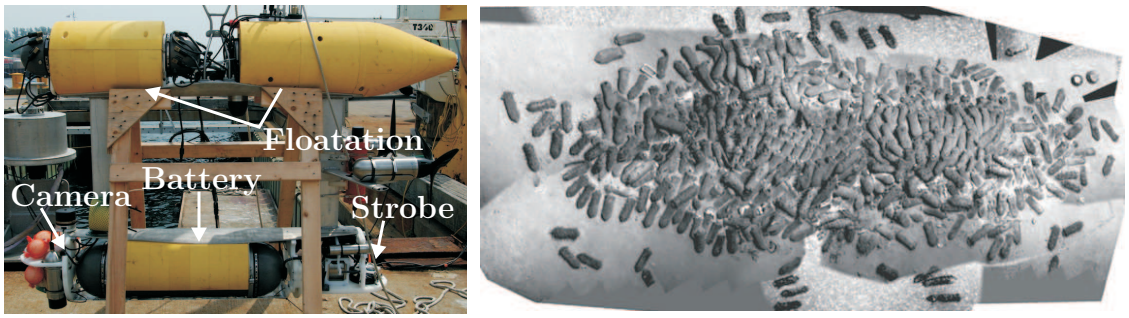


Figure 1-4: *Jaguar Seabed's* sister ship. An AUV specifically developed for underwater imaging. Left: *Seabed* without fairings showing the large floatation (yellow) on the top segment and the heavy battery compartment in the bottom segment. The flash is mounted at the stern in a glass bowl and the camera in the bow. This ensures maximum separation between light source and camera axis to avoid backscatter. Right: photo mosaic created from ≈100 pictures taken by an ROV of an iron age shipwreck off Ashkelon, Israel. Photo courtesy of Hanu Singh.

several 1000 km [63]. However it limits the glider to speeds of 0.2-0.5 m/s, which can become a problem in the presence of strong currents. The need to keep the total power consumption around 1 W does not allow for sophisticated navigation equipment as outlined in section 2.1.3. A recently developed glider, the *X-Ray*, shown in figure 1-3 and described in [1], mitigates some of the disadvantages of gliders. Being significantly larger than other gliders it provides enough battery capacity for more sophisticated payloads and navigation sensors. The *X-Ray's* steeper ascent and descent angles cause a forward speed of 0.75 to 1.5 m/s which is close to the cruising speed of propelled AUVs.

1.1.2 Applications

Due to the increasing availability and reliability of AUVs they are now used for a widespread range of applications. The following sections will illustrate a few examples from the large and growing number of tasks performed by underwater vehicles today. These applications increase in complexity which requires an increasing level in vehicle autonomy. The examples also show how the different designs emerged out of task-specific needs. The certainty of the vehicle's position estimate determines the quality of the results (e.g. maps) for all applications. Hence, these examples illustrate how the utility of autonomous marine vehicles is directly tied to their ability to self-localize and therefore use all available information to minimize the position uncertainty.

Mapping

The earliest and still most widespread application for AUVs is mapping. Early AUV payloads would only contain sensors for physical and biological water properties such

as temperature, salinity, turbidity and fluorescence, but the increased payload capability of today's vehicles allows for sonars which provide micro-bathymetry (pencil or multi-beam) [71], video-like acoustic images (DIDSON in figure 1-5) of underwater structures (side scan figure 1-6) and even information about buried objects, such as mines, from low-frequency sub-bottom profiling sonars [29].

More recently, cameras have been attached to AUVs. The absence of natural light, except in very shallow waters, requires artificial illumination. Also, the strong attenuation of light underwater limits the distance between camera and the photographed object, thus a single picture can only cover a few m^2. To cover larger areas an AUV typically takes several thousand pictures during a mission. Sophisticated mosaicking techniques are then used to combine the individual frames to a complete picture of the seafloor [80]. This technique has been used to map ancient shipwrecks [9] as well as coral reefs [4].

Most mapping applications require the vehicle to run in a pre-programmed "lawnmower" pattern to ensure that the sensors cover a predefined area. The data collected by the sensors is stored on-board. After the mission, sensor data is combined with the vehicle's navigation data in order to create a map. Consequently, mapping applications require the vehicle to have a very precise estimate of its position throughout the entire mission as the navigation data is later used to globally reference the sensor readings. This is particularly relevant for underwater imaging applications.

Inspection

A special case of a mapping mission is the inspection scenario. Here the AUV is required to map one or several features such as oil rigs, harbor structures or ship hulls. These features are mostly man-made and have a very complex shape. The inspection requires the vehicle to be very close to the feature so that it can be mapped in detail with underwater imaging or a high resolution sonar. Thus the vehicle needs to adapt its trajectory to the feature's shape to provide full sensor coverage. Figure 1-5 shows MIT/Bluefin's Hovering Autonomous Underwater Vehicle (HAUV) mapping the ship hull of a decommissioned cruiser. Future applications include the inspection of Liquefied Natural Gas (LNG) tankers for explosives attached to their hulls prior to entering port areas. An explosion of a ship with such cargo could have a catastrophic effect. Today these inspections are carried out manually, requiring a large number of divers. The HAUV would approach the ship on a preprogrammed track and would then move in a complex lawnmower pattern along the hull. Using the forward looking Doppler Velocity Log (DVL), it obtains information about its distance to the hull as well as relative speed. By extracting features from the Dual Frequency Identification Sonar (DIDSON) the HAUV ensures that it obtains a consistent map covering the entire hull [90].

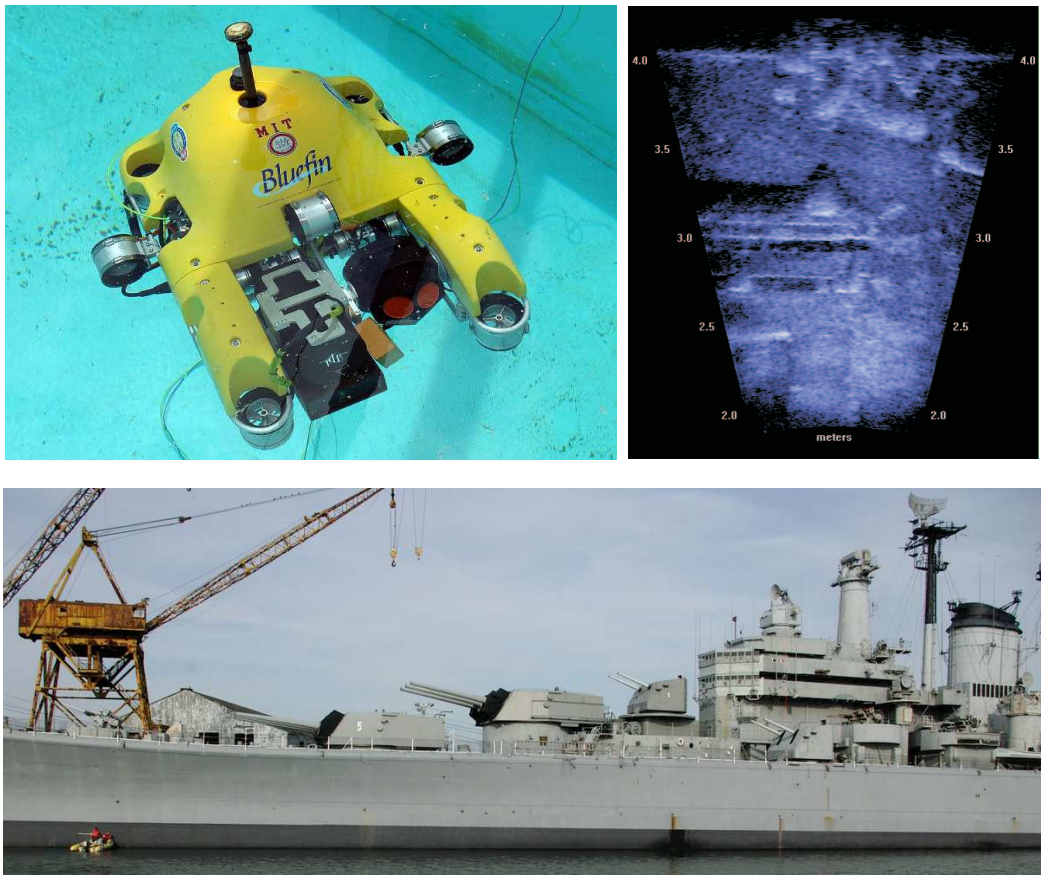


Figure 1-5: Ship hull inspection using the *HAUV* to inspect the *Salem* in Quincy, MA, USA. Top left: the *HAUV* in the pool at MIT. Note the eight thruster for increased maneuverability, the forward looking DVL (black cone with red circles) which provides distance and speed of the *HAUV* relative to the ship hull and the forward looking DIDSON to the left of the DVL. Top right: output of the forward looking DIDSON showing cooling systems and bio fouling on the hull of the *Salem*. Bottom row: the decommissioned heavy cruiser *USS Salem (CA-139)* in Quincy, MA, USA while inspected by the *HAUV*. Top pictures courtesy of Jerome Vaganay from Bluefin Robotics.

Tracking

Unlike mapping applications in which the area of interest is usually static and the vehicle trajectory can often be entirely preprogrammed, tracking applications require a higher level of vehicle autonomy. In a typical tracking application the vehicle, after being released, enters a search or loiter pattern until its sensors pick up signatures of the feature which is to be tracked. It then breaks from its initial search/loiter behavior and adapts its trajectory to maximize sensor effectiveness and information gain about the feature.

In [28] Eickstedt et al. simulate an Anti-Submarine Warfare (ASW) scenario in which an AUV with a towed array starts in a loiter pattern and listens for acous-

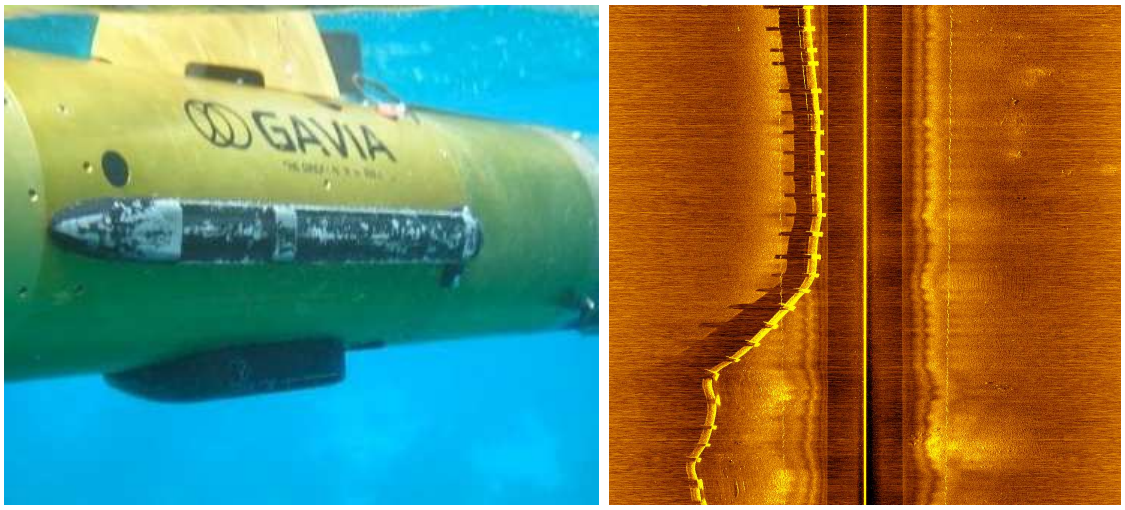


Figure 1-6: Autonomous pipeline inspection. Left: sidescan transducers on the *Gavia* AUV. Photo courtesy of the NOAA OE Bonaire 2008 Expedition. Right: sidescan image of an undersea-pipeline taken by a *Gavia* AUV. The image shows echoes from the raised pipeline as well as the acoustic shadow it casts on the sea floor. A vehicle course that is not parallel to the pipeline creates a distorted image. Image courtesy of Hamynd.

tic signatures of a potential target. After picking up a signature and obtaining an estimate of the target's course and speed it attempts to close the distance between itself and the target and to align its array for optimal tracking. Later experiments verified the feasibility of this approach. In [41] German et al. use the AUV *ABE* to find hydrothermal vents, an often very small feature (few m^2) on the sea floor. *ABE* initially performs a "lawn-mower" search over a predefined area. After completing its initial search it autonomously revisits all locations where chemical sensors suggested the presence of a vent for a finer grid search. If required, this process of "revisit and refine" is repeated until the vents are sufficiently well localized to warrant a small-scale photo mosaicing to visually confirm the hydrothermal vent's presence.

Another tracking application is shown in figure 1-6. The *Gavia* AUV is released close to an underwater pipeline. While underway it obtains readings from its side scan sonar which it synthesizes to the picture shown on the right side in 1-6. An online feature detector tries to extract the pipeline from the sonar image and sets a course which keeps the AUV parallel to the pipeline at a fixed distance. This is particularly difficult when the pipeline is partially buried.

1.2 Contributions of this Thesis

1.2.1 Problem Statement

The previous section showed the two reasons why almost all applications for underwater vehicles rely on the vehicle having a very accurate estimate of its position. First, the vehicle continuously makes unsupervised decisions based on its position while the mission is carried out. Second, the utility of the collected data is directly related to the precision with which the samples can be localized in a global frame.

For almost all robots operating outdoors - on land and in the air - the localization problem has been resolved with the advent of GPS. A very affordable receiver is able to provide an absolute position at a high rate leading to estimates which are accurate to a few meters. Using additional infrastructure such as differential GPS the accuracy can be increased to a few centimeters. Robots operating indoors usually do not have access to the GPS signal but modern sensors, such as laser scanners and high resolution cameras can extract a rich set of features which can greatly alleviate the localization problem.

This leaves underwater locations as the single largest domain excluded from today's most prevalent localization techniques. While underwater sensors continue to improve, the strong absorption of almost the entire Radio-Frequency (RF) spectrum in salt water will impose physical limitations on radio-based localization methods for the foreseeable future, just as generally feature-poor marine environments will limit the usability of natural feature based navigation methods. In the absence of these classic options new strategies such as cooperation for navigation will play an important role in ensuring that the navigation accuracy for underwater vehicles will be similar to that which has become standard for outdoor robots.

The particular strength of cooperative navigation is the fact that it does not require any additional infrastructure or even instrumentation of the vehicle. The sensor and communication package which is standard on today's underwater vehicles is sufficient and adding cooperative navigation requires merely a change in the vehicle's navigation and control software. The main requirement for this approach - deploying more then a single vehicle - will be satisfied for most of the upcoming deployments due through the increased reliability and availability of suitable platforms.

This thesis investigates the main shortcomings of conventional (non-cooperating) vehicle navigation. It identifies and proposes solutions for the three key problems which need to be resolved to successfully implement cooperative navigation.

1.2.2 Cooperative Localization Algorithm

The methods previously proposed for cooperative localization of land and air vehicles make assumptions, particularly on the reliability and bandwidth of the communications channel which do not hold underwater. We propose an algorithm which is specifically adapted to the underwater communication and messaging infrastructure as well as to the vehicle's sensor suites to provide a robust estimate of the actual position.

1.2.3 Maintaining Consistency

If a number of vehicles exchange navigation information, the position estimates of the participating vehicles will become correlated and will, over time, suffer from overconfidence. This is independent of the cooperative localization algorithm used and must be mitigated as overconfidence in the vehicle's position estimate can lead to catastrophic results. One option is the use of dedicated navigation beacons as this method ensures that the flow of navigation information is uni-directional. For the more general case in which each vehicle participates in cooperative navigation actively, by broadcasting, or passively, by receiving, we propose an algorithm which selectively incorporates information while keeping the positions of all vehicles decorrelated. This selective update algorithm works with any underlying navigation algorithm.

1.2.4 Motion Planning for Cooperating AUVs

The amount by which we can decrease the uncertainty in our position estimate through cooperation depends not only on the certainty in the position estimate of our cooperation partner, but also on the geometry between the vehicles. A special case of cooperative navigation uses vehicles which are dedicated navigation beacons. As the only purpose of these vehicles is to collectively minimize the position uncertainty of the receiving vehicles they need to constantly adapt their position. We propose a distributed algorithm which dynamically positions each of these designated beacon vehicles to maximize the effect of their position broadcasts.

1.2.5 Experiments

We first tested the performance of the communication infrastructure used in order to assess how well it is suited for cooperative navigation. We then carried out a series of experiments to verify the performance and robustness of our cooperative navigation algorithm. By first relying on surface vehicles only, which had access to GPS, but would only communicate through the acoustic channel available to submerged AUVs, we obtained ground-truth which the results of our localization algorithm could be compared against. We then substituted one of the surface craft with an AUV featuring a very sophisticated navigation suite and, in another experiment, with a glider possessing very minimalist sensors. Both experiments showed that the cooperation would lead to a noticeable reduction in the position uncertainty.

1.3 Thesis Outline

The remainder of the thesis is organized as follows:

Chapter 2: AUVs: Communication and Navigation Capabilities
We start by giving an overview of the state of the art in the two domains which are most crucial to cooperative navigation: underwater communication and vehicle navigation sensors. The most commonly used navigation sensors,

along with their relevant performance parameters, are introduced as well as the options for underwater communication.

Chapter 3: Cooperative Localization
This chapter establishes the probabilistic framework in which we will present the cooperative localization problem. We introduce the two classical methods which have been used for localization problems in general and the related work in which they have been adapted for cooperative localization. Based on the shortcomings of these two methods we propose our algorithm. This chapter also addresses the problem of correlations arising from sharing navigation information and the resulting overconfidence in the position estimate. Our method, the IU algorithm, is a general solution for all cooperating navigation algorithms which modifies their update step to keep all vehicle positions decorrelated and thereby preventing overconfidence.

Chapter 4: Intra-Vehicle Geometries for Cooperating AUVs
The specific scenario, in which some vehicles serve as dedicated navigation beacons, requires that these vehicles adapt their position in order to maintain an advantageous relative position to all other vehicles for which they provide the information for. We show how the geometry between beacon and receiving vehicles has a strong influence and we propose an algorithm which positions the beacon vehicles optimally based on locally available information.

Chapter 5: Experiments
This chapter shows the results of the experiments we carried out to validate our cooperative navigation algorithm using different surface and underwater platforms. We also compare the performance of our algorithm with the classical approaches.

Chapter 6: Conclusion
The last chapter shows the direction for future research and summarizes the contribution of this thesis.

Appendix A: Coordinate Systems
The appendix establishes the coordinate system used throughout the thesis.

Chapter 2

AUVs: Communication and Navigation Capabilities

2.1 Underwater Navigation

2.1.1 Navigation Sensors

This section gives an overview of the sensors commonly used in underwater vehicles. It outlines their particular characteristics and shows how several sensors are used jointly to determine a vehicle's position. It is important to note that our cooperative navigation approach does not replace any of these instruments, but adds a "virtual" sensor by combining the measurements of the physical sensors from its own and other vehicles.

Depth Sensor

All submersible vehicles are outfitted with a pressure sensor which allows them to determine their absolute depth with high accuracy and a high update rate. As a result all other underwater navigation systems are only used to resolve the 2D position, (i.e. longitude and latitude) and all underwater vehicle related localization problems are stated in 2D.

Magnetic Compass

Like the pressure sensor, a compass is part of the basic navigation sensor suite of every underwater vehicle as it is an inexpensive and low-power device. It provides the 3D-vector of the local magnetic field. Before computing the heading from the magnetic field vector it is necessary to carefully calibrate the compass each time the vehicle's area of operation changes, as the difference between the orientation of the 3D magnetic field vector and the direction of true north (called "variation") depends on the geographic location. In addition to the spatially slow variation, there are highly localized "magnetic anomalies". The compass output is also affected by its position

in the vehicle as electrical currents create magnetic fields which cannot be discerned from the Earth's magnetic field.

Global Positioning System (GPS)

The GPS is able to provide absolute position information for outdoor land robots as well as Unmanned Aerial Vehicles (UAVs), but the strong absorption of electromagnetic waves by sea-water prohibits the use of the GPS by submerged AUVs. Nonetheless, almost all underwater vehicles today are equipped with a GPS receiver as it can be used to get a position fix before the start of the mission or during intermittent surfacings.

Flow meter

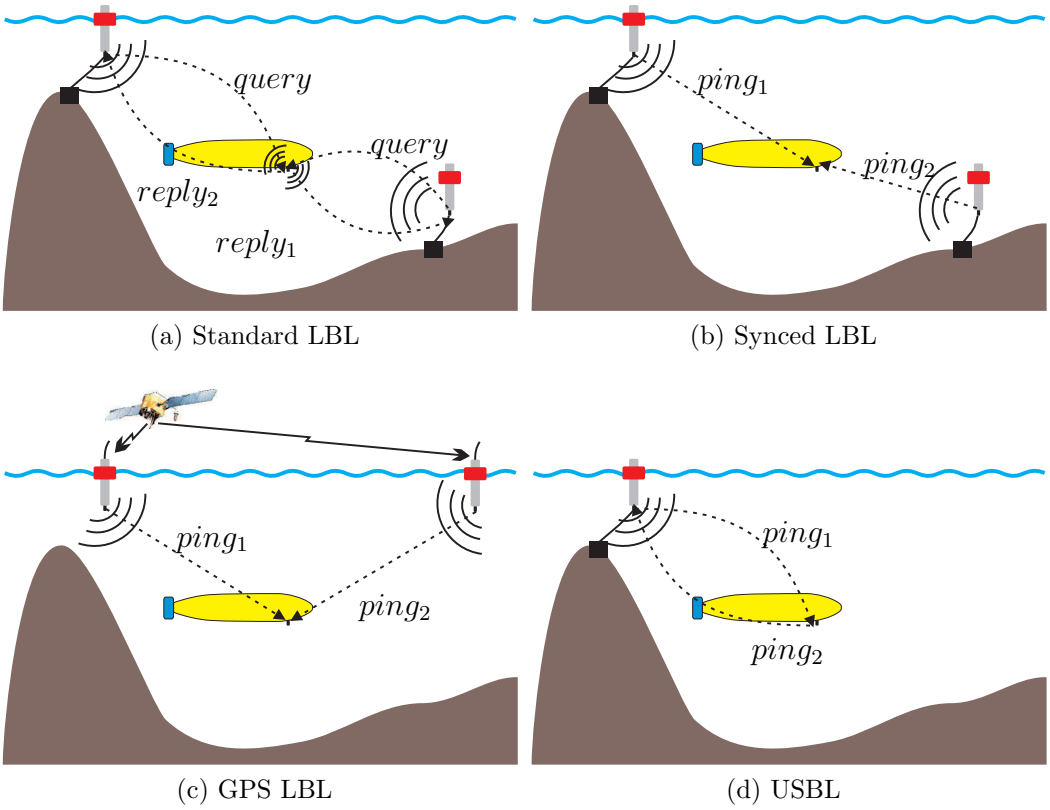
A flow meter consists of a tube, usually mounted in line with the main vehicle axis and is open to the surrounding water on both sites. It contains a propeller which is spun by the water running through the tube as the vehicle moves. A sensor attached to the propeller determines the rotational speed which can be converted into flow speed and thereby giving an indication of the vehicle's speed relative to the water body.

Beacon Techniques

The most commonly used way to obtain absolute position information underwater is through the use of beacons. These beacons are at known locations and the AUV obtains the range and/or bearing to several of these and then calculates its position through trilateration or triangulation. Based on the location of the transceivers we can identify three different baseline systems.

Standard Long Baseline (LBL): A typical LBL-configuration is shown in figure 2-1a. Two or more beacons are deployed around the perimeter of the area in which the AUV will operate. These beacons are anchored and float on the surface or, particularly in deeper water, a few meters above the sea floor. Each unit listens to acoustic query pings on a common receive channel. After receiving a query ping from an AUV, each unit waits for a unique-specific Turn-Around Time (TAT) t^{TAT} and then sends out a reply ping on its individual transmit channel. The AUV then receives the reply pings. The transmit channel as well as the TAT are different for each unit. A unique TAT ensures that two beacons will not interfere by transmitting at the same time and by using different transmit frequencies the beacons provide a way for the AUV to identify from which unit a reply ping was sent. The time difference Δt_i between sending out the query ping and receiving a reply can then be used to determine the One-Way Travel Time (OWTT) t_i^{owtt}.

$$t_i^{owtt} = \frac{\Delta t_i - t_i^{TAT}}{2}$$



(a) Standard LBL (b) Synced LBL

(c) GPS LBL (d) USBL

Figure 2-1: Beacon-based underwater localization techniques.

The distance d_i between a beacon i and the AUV is then given by

$$d_i = \frac{c}{t_i^{owtt}}$$

where the speed of sound c is either a pre-programmed value or measured on-board. Using range measurements to several beacons and the beacon positions stored in the vehicle before deployment, the AUV can now trilaterate its position.

The maximum possible distance between the AUV and a beacon as well as the localization accuracy depend on the the frequency band used for query and reply pings. Long-range LBL-systems using the 12 kHz band work over distances as long as 10 km [93] and can provide provide an absolute position with an error between 1 m and 10 m. Short-range LBL systems using frequencies around 300 kHz band can achieve sub-centimeter precision, but the maximum range is limited to 100 m [93]. The indicated errors assume that large outliers have been filtered out. These outliers, which can be seen in figure 2-2, are due to reflections from temperature and salinity discontinuities which are further

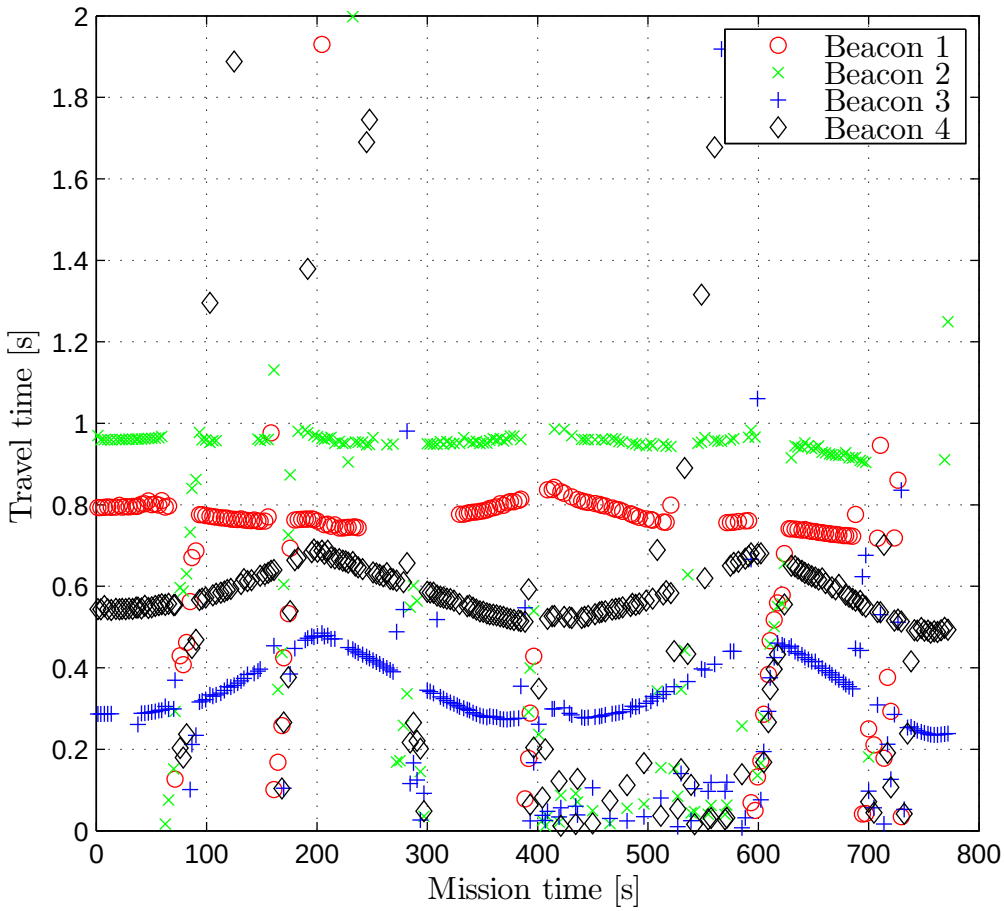


Figure 2-2: Time-of-flight obtained from four LBL beacons. The plot shows significant outliers for all beacons, particularly between 400 s and 600 s. This data was obtained during the GOATS2002 experiment.

explained in section 2.2.2.

LBL variants: Standard LBL systems such as the one described above are not well suited for large groups of AUVs because only one vehicle at a time can query the beacon network and get a position update. Thus the position update interval increases with the number of vehicles. Newer LBL systems, like the one recently developed by *ACSA* [83] and shown in figure 2-1b, have synchronized clocks in the beacons and the AUV transceiver units. The beacons broadcast a ping containing a unique identifier at fixed time intervals. When the AUV receives this ping, the beacon's known broadcast schedule and the synchronized clock's time ensure that the vehicle knows when a ping was sent and can directly compute the OWTT. The synchronized clocks thereby eliminate the need for query pings and allow all vehicles within range of the beacons to get a range

(a) DVL

(b) REMUS AUV with up-(arrow) and downward looking DVL

Figure 2-3

to the broadcasting beacon. As a result the ping interval is independent of the number of vehicles relying on the beacon network.

Another improvement over conventional LBL is the system depicted in figure 2-1c. Building on the setup in figure 2-1b the beacons now transmit their GPS position along with the unique identifier. As with the system described previously, the vehicles do not need to query the beacons. With the position of the beacons embedded in the ping the beacons can float freely and it is not necessary to store their coordinates in the AUV before deployment.

Ultra-Short Baseline (USBL): Another variant of beacon based navigation systems is USBL (figure 2-1d). Here the beacon is of the same kind as in a standard LBL system, but the transceiver on the AUV contains several receiving elements which are very close to each other. After querying the beacons the reply ping is captured by all receiving elements. The phase difference between the signals coming from the different receiving elements allows the AUV to compute a bearing to the beacon. Combined with the beacon position stored in the AUV and the distance d obtained from the OWTT, the vehicle can compute its absolute position using only a reply from a single beacon.

Modern beacon-based systems such as the ones shown in figures 2-1b, 2-1c and 2-1d significantly decrease the pre-deployment effort when compared to early beacon-based systems such as the standard LBL. However all beacon-based systems confine the operating area of the vehicles to a polygon of beacons or, as in the case of USBL, to the coverage radius of a single beacon. Thus beacon-based navigation is only feasible for operating areas of $\mathcal{O}(10\ \mathrm{km}^2)$ in size.

Doppler Velocity Log (DVL)

A DVL (figure 2-3a) is a device which typically has four transceiver units that emit acoustic pulses. When a DVL is used for navigation purposes it is usually mounted to a vehicle such that the transceivers are facing downward. If the DVL is close enough

to the bottom, the transceiver will receive the reflected pulses ("bottom-lock") and as the transceivers are mounted at an angle with respect to the sea floor plane, the received pulses will be subject to a Doppler-shift if the vehicle is moving. Combining the measured Doppler-shifts from all 4 sensors with the built-in roll, pitch and heading sensors the DVL can then compute the vehicle's 3D-speed vector $\boldsymbol{v}_v = [\dot{x}, \dot{y}, \dot{z}]$ in a world-referenced frame.

The maximum distance between the DVL-unit and the sea floor depends on the operating frequency of the transceivers. Low-frequency (150 kHz) DVL can obtain bottom-lock for ranges up to 500 m, while a high-frequency DVL (1200 kHz) requires less than 30 m.

The ranges indicated above can only be obtained under ideal conditions. A soft sea floor or vegetation can absorb most of the energy of the incoming pulse and thereby significantly decrease the maximum range. Another option is to mount the DVL in an upward looking configuration such that the acoustic pulses are reflected at the water/air interface ("surface-lock"). Then, the vehicle measures its speed relative to the water surface, but this strategy may introduce errors in the case of significant surface currents. Figure 2-3b shows a REMUS 100-AUV with a double-DVL configuration. If bottom-lock cannot be obtained with the downward-looking DVL the vehicle tries to determine its speed using the upward looking unit. Recent developments greatly increased the accuracy of DVL-systems and errors as low as 0.2 % (1200 kHz) or 1 % (150 kHz) can be obtained.

Attitude Heading Rate Sensor (AHRS)

An AHRS unit typically consists of a 3-axis linear acceleration sensor as well as a 3-axis gyroscope and a heading sensor (magnetic compass). Combining the measurements from these sensors, the AHRS can compute the 3 linear and 3 angular velocities and accelerations (rates) as well as the attitude and heading from the windowed-average of the linear acceleration sensor readings and the compass.

Inertial Navigation System (INS)

The sensors of an INS are the same as those of the AHRS described above. In addition to the AHRS, the INS uses information from absolute position sensors (such as GPS or LBL) and integrates the rate sensor readings to compute the actual position. This process is called Dead-Reckoning (DR). As the linear and angular acceleration sensors are subject to noise, the position derived from these sensors in the absence of GPS or LBL is subject to a cumulative error and the obtained position will drift with respect to the true position. The drift (error) e between the vehicle's true position $\boldsymbol{x}_{true}$ and the position obtained with DR $\boldsymbol{x}_{DR}$ are expressed as "drift over time" or "drift over distance traveled"

$$e = \frac{\|\boldsymbol{x}_{true} - \boldsymbol{x}_{DR}\|_2}{\Delta t} \qquad \text{or} \qquad e = \frac{\|\boldsymbol{x}_{true} - \boldsymbol{x}_{DR}\|_2}{\Delta x}$$

Typically the heading and rate sensors of an INS are less noisy than those of a

comparably cheap AHRS which decreases the problem of accumulated drift. An INS which fits into the hull of an AUV shows typical drift rates of 1 km/h [46]. The exact performance of the most precise INS available, the ones found in nuclear submarines, remains classified, but is expected to be $\mathcal{O}(0.01$ km/h).

2.1.2 Sensor Fusion

The sensors described in 2.1.1 can be divided into three groups.

1. **Absolute position sensors**
   The output of these sensors is the absolute position of the vehicle in a global reference frame $\boldsymbol{x} = [x, y, z]$.GPS and LBL are examples of absolute position sensors. Visually Augmented Navigation (VAN) is a method presented by Eustice [31]. He shows that an AUV continuously taking pictures of the sea floor while traveling can use these pictures to bound the navigation error introduced by the DR-sensors. As a result this method can be considered an absolute position sensor. Walter uses a similar technique to bound the navigation error by using the output of an imaging sonar navigating along a ship hull for inspection [90]. A special case of an absolute position sensor is the pressure sensor as it resolves the position of an underwater vehicle for the z-dimension only.

2. **DR-sensors**
   The output of these sensors is a speed vector $\boldsymbol{u}^V = \left[v_x^V, v_y^V, v_z^V\right]$ for a DVL or v_x^V for a flow meter, or an acceleration vector, $\boldsymbol{a}^V = \left[a_x^V, a_y^V, a_z^V\right]$ in the case of the AHRS.

3. **Attitude and heading sensors**
   Compass or Fiber-Optic Gyroscopes (FOGs) provide the heading/yaw of our vehicle. Combining this information with the 3D gravity vector, which is provided by the AHRS, we obtain the orientation of the vehicle in space described by the yaw θ, pitch ψ and roll ϕ angle (see appendix A for the definition of the axes and angles).

The Main Vehicle Computer (MVC) collects all incoming information from the sensors and constantly computes the pose, the position and attitude of the vehicle. This process is called *sensor fusion* and is the essential part of every navigation algorithm. The following section describes a basic DR algorithm. More sophisticated algorithms are introduced in section 3.4. The basic DR algorithm is shown in algorithm 1 and explained below.

Initialization

Before being able to rely on DR-sensors the navigation algorithm needs to be initialized with an absolute position $\boldsymbol{x}_0$ from an appropriate sensor (line 4). Usually this is a GPS which is used on the surface before the vehicle submerges.

1: **repeat**
2: wait
3: **until** $\boldsymbol{x}_0$ is available
4: $\boldsymbol{x}(t) = \boldsymbol{x}_0$
5: **loop** {for each $d\tau$}
6: $\boldsymbol{u}^V_{\text{AHRS}}(t) = \int_t^{t+d\tau} \boldsymbol{a}^V_{\text{AHRS}}(t)d\tau$
7: $\boldsymbol{u}(t) = \boldsymbol{R} \cdot \left(u^V_{\text{AHRS}}(t) + \boldsymbol{u}^V_{\text{DVL}}(t)\right)$
8: $\boldsymbol{x}(t+1) = \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \boldsymbol{x}(t) + \int_t^{t+d\tau} \begin{bmatrix} u \\ v \\ 0 \end{bmatrix} d\tau + \begin{bmatrix} 0 \\ 0 \\ z \end{bmatrix}$
9: **if** absolute position measurement $\boldsymbol{x}_{\text{abs}}(t)$ is available **then**
10: $\boldsymbol{x}(t) = \boldsymbol{x}_{\text{abs}}(t)$
11: **end if**
12: **end loop**

Algorithm 1: Simple DR which uses an initial position $\boldsymbol{x}_0$ and acceleration and speed measurements to estimate the vehicles's actual position $\boldsymbol{x}(t)$.

Dead Reckoning

Acceleration sensors such as an AHRS provide a vehicle referenced acceleration vectorn $\boldsymbol{a}^V_{\text{AHRS}}(t)$ which can be integrated to obtain the vehicle-referenced velocity vector (line 6).

$$\boldsymbol{u}^V_{\text{AHRS}}(t) = \int_t^{t+d\tau} \boldsymbol{a}^V_{\text{AHRS}}(t)d\tau$$

Other sensors such as the DVL or the flow meter directly provide the speed vector $\boldsymbol{u}^V(t)$ in vehicle coordinates. We combine the speed vector $\boldsymbol{u}^V_{\text{DVL}}(t)$, with the speed vector from the AHRS $\boldsymbol{u}^V_{\text{AHRS}}(t)$ computed in the previous step. The information from the attitude sensors is then used to compute $\boldsymbol{R}$, the rotation matrix which converts the joint vehicle-referenced velocity vector to a world-referenced velocity vector (line 7).

$$\boldsymbol{u}(t) = \boldsymbol{R} \cdot \boldsymbol{u}^V(t) = \boldsymbol{R} \cdot \left(u^V_{\text{AHRS}}(t) + \boldsymbol{u}^V_{\text{DVL}}(t)\right)$$

Integrating the world-referenced velocity vector combined with the initial position leads to the dead-reckoned position. Usually only the x and y-coordinate are obtained using dead-reckoning, and the z-component (depth) is readily available from the pressure sensor (line 8).

$$\boldsymbol{x}(t+1) = \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \boldsymbol{x}(t) + \int_0^{t} \begin{bmatrix} u \\ v \\ 0 \end{bmatrix} d\tau + \begin{bmatrix} 0 \\ 0 \\ z \end{bmatrix}$$

Reinitialization

Whenever information from an absolute position sensor $\boldsymbol{x}_{\text{abs}}(t)$ is available, the navigation algorithm sets the current position to the obtained position ($\boldsymbol{x}(t) = \boldsymbol{x}_{\text{abs}}(t)$).

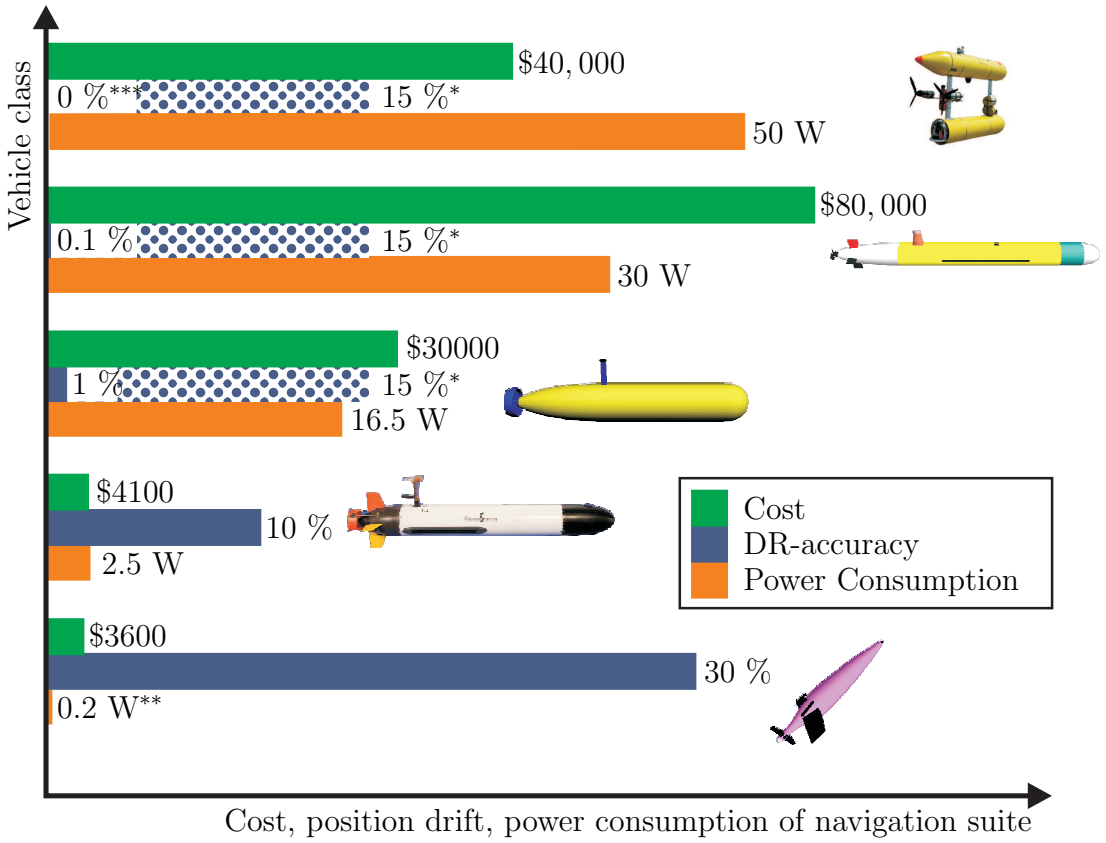


Figure 2-4: Navigation accuracy, power consumption and price of various AUV sensor suites (bottom to top): **1** Glider with compass and attitude sensor **2** Low-cost AUV with compass, attitude sensor and flow meter **3** Medium-range AUV with INS, DVL and LBL **4** High-end AUV with FOG-based INS, DVL and LBL **5** Special-task AUV with with INS, DVL and VAN.
* Drift in mid water-column when DVL cannot obtain bottom or surface lock.
** Assuming a 10 % duty cycle during which the navigation sensors are powered.
*** Assuming that the vehicle was close enough to the sea floor throughout the entire mission to take pictures and revisit places.

2.1.3 State-of-the-Art in Underwater Vehicle Navigation

Every underwater vehicle contains a subset of the navigation sensors described in 2.1.1. Which sensors are used depends on the navigation accuracy required for the mission as well as the available power, space and the cost constraints. Figure 2-4 shows five typical configurations.

Glider with very low power sensor suite

Gliders must operate for extended periods of time without being able to recharge their batteries. As a result power consumption is the limiting factor for the selection of navigation sensors, and the navigation suites of a glider usually consists of a GPS, an AHRS and a pressure gauge. While submerged the glider uses the AHRS combined with a vehicle model to estimate its heading and forward velocity and dead-reckon its position. The high noise and the unobservable variables in the vehicle model lead to a very high drift of 30 % or even more if strong currents are present. On the surface the vehicle resets its position estimate using GPS.

Low-cost AUV sensor suite

Low-cost AUVs such as the *IVER* use a flow meter to obtain a measurement of their forward speed u_x^V. This information combined with an AHRS leads to a significant improvement of the navigation accuracy when compared to that of a glider.

Standard AUV sensor suite

The standard AUV adds a DVL to the list of sensors. When the DVL is able to obtain bottom lock a very accurate vehicle-referenced velocity vector $\boldsymbol{u}^V$ is available and the navigation accuracy improves by an order of magnitude. Drift rates as low as 1 % of the distance traveled can be obtained with a well calibrated magnetic compass. Standard AUVs operating in a confined area are often outfitted with an LBL system. When operating within the polygon established by the position of the LBL beacons, the position drift will remain bounded.

High-end AUV

The dominant source of error in the standard AUV sensor suite described above, is introduced during the transformation of vehicle-referenced velocities to world-referenced velocities as a result of errors in the heading measurements. Replacing the simple magnetic compass with a FOG improves the navigation by two orders of magnitude (0.1 % of the distance traveled). When the DVL is not able to obtain bottom lock all of the vehicles described so far can only rely on the vehicle's linear acceleration sensors to obtain velocities. Due to the large noise introduced by these sensors the navigation accuracy decreases dramatically.

Special-task AUV using VAN

The special-task AUV has the same sensors as the standard AUV. Additionally, it uses a bottom-looking camera to take a series of pictures of the sea-floor. When revisiting a point it has taken a picture of before, it is able to recognize that fact and the navigation algorithm is able to reset the drift. As a result the drift is bounded. This technique has been described by Eustice in [31]. Figure 2-5 shows two consecutive camera images with corresponding features. This method however requires the AUV

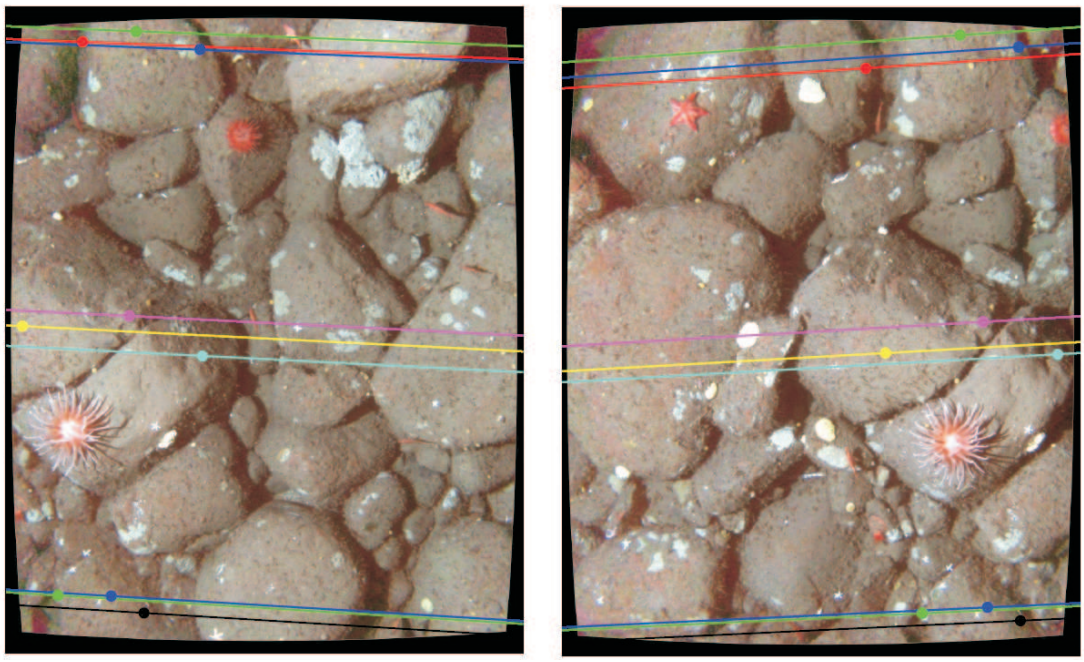


Figure 2-5: Two consecutive pictures after being processed by the VAN algorithm. From the several hundred features identified in each picture, only nine, marked by the colored dots have correspondences in both pictures and fit within the epipolar constraints. Figures courtesy of Ryan Eustice.

to revisit points and stay close enough to the sea floor (less than 10 m) to acquire the pictures.

2.2 Underwater Communication

2.2.1 Technologies

Communication is vital for any collaborative effort such as cooperative navigation. This section briefly describes communication based on electromagnetic waves (RF and optical), the most common mode for untethered data exchange for land and air based systems. As their applicability under water is limited to small niches, the second part of this section addresses acoustic communication the most commonly used channel under water.

Radio Communication

Using electromagnetic waves, particularly in the Ultra-High Frequency (UHF) spectrum around 900 MHz or 2.4 GHz, is the most common way of wireless communication for land robots and UAVs. Radio communication enables these vehicles to communicate over distances from a few meters to anywhere on the Earth, when using satellite

communication, at comparably high speeds (kbps to Mbps).

Due to the strong attenuation of radio waves under water, radio communication is only used in two niches in the realm of submersible vehicles.

1. **Super Low Frequency (SLF) for long-range communication**
   The US-American *Seafarer* and the Russian *ZEVS* system each consist of a single base station, transmitting at 76 Hz and 82 Hz respectively. The antenna structures of the base stations are up to 90 km [61] in size while the submarines tow a very long antenna. This setup allows one-way communication from the shore station to submerged submarines anywhere in the world with data rates $\mathcal{O}$(bits/min). Due to the size and power requirements of the transmitter, the submarines cannot accommodate these. Due to the low data rate provided by this communication channel it is only used to signal a submarine to come to the surface to initiate communication through a satellite.

2. **Low Frequency (LF) for short range communication**
   Recently, several transceivers have been developed which operate in the LF-band (30 kHz to 300 kHz) and can be used for two-way AUV communication. Schill et al. developed a transceiver which is small enough to fit into a very small AUV and is capable of data rates up to 8 kbps and communication ranges up to 10 m [39]. Rhodes et al. developed a similar system which is now commercially available and suitable for mid-size AUVs [92].

While SLF communication is not feasible for AUV-communication the LF-band is a viable option for short range ($\leq$10 m) and medium data rates ($\approx$30 kbps). Unlike optical communication it does not require the two transceivers to be aligned and it is not as bandwidth-constrained and susceptible to background noise as acoustical communication.

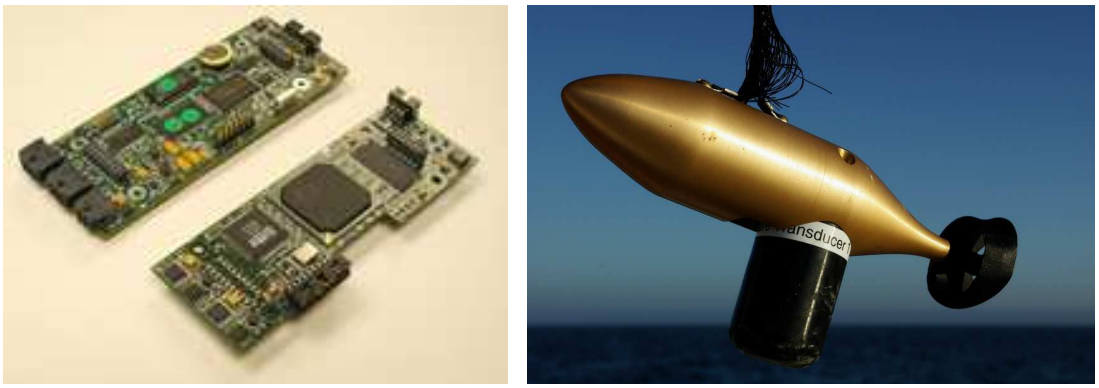
Optical communication

Like radio communication, electro-magnetic waves in the visible spectrum are strongly attenuated under water. Additionally, scattering from suspended particles further decreases the maximum possible range. Schill [39] and Vasilescu [89] both implemented a low-cost optical modem using powerful LEDs as transmitters and photo-diodes, achieving data rates around 50 kbps over distances of 3 m. If the two transceivers can be perfectly aligned, the maximum distance can be increased up to 10 m by adding focusing lenses to the transmit LED.

By using a steered laser beam as a transmitter and a photo-multiplier as a receiver Farr et al. [33] hope to increase the distance to 100 m and the transmission rate to 10 Mbps.

Acoustic Communication

Given the range restrictions of electromagnetic waves in the optical and RF spectrum, acoustic communication is the only available technology today for underwater communication over longer ranges.



(a) WHOI micromodem (b) Modem transducer mounted to towfish

Figure 2-6: The WHOI micromodem. Left: the base board and an additional DSP board to process high data-rate message. Right: the transducer (black cylinder) mounted into a towfish.

2.2.2 Acoustic Communication Constraints for Underwater Vehicles

Acoustic communication has been the subject of research for over three decades. Due to the advent of low-power Digital Signal Processors (DSPs) acoustic modems suitable for deployment in AUVs have been developed and are now commercially available.

The acoustic modem which we will refer to in this thesis, the *micromodem*, has been developed by the Woods Hole Oceanographic Institution. Figure 2-6a shows the base board (top) and the additional DSP. The transducer for transmitting and receiving is either directly mounted to the vehicle's hull (figure 5-1a) or attached to a towfish (figure 2-6b). The transducer consists of a piece of ceramic which in transmit mode expands and contracts and creates a transversal pressure wave which travels through the water body. The incoming pressure wave excites the transducer of the receiving modem which creates an electric signal that is interpreted by the modem.

Synchronized Data Transmission

An important feature provided by the WHOI *micromodem* is its ability to synchronize data transmission to an external signal. A timing board with a very precise oscillator can provide a Pulse Per Second (PPS) signal. This oscillator is synchronized to the global clock signal of the GPS when the vehicle is on the surface. During data transmission the modem ensures that the start of the message is synchronized to the PPS signal. A receiving modem on a different vehicle which also has access to the globally synchronized PPS signal can determine the Time-of-Flight (TOF) by registering the time at which the first symbol of the incoming message was received. The PPS feature can only resolve the TOF within one second, but by embedding a time stamp in the data message longer TOF can be resolved. The syncing of

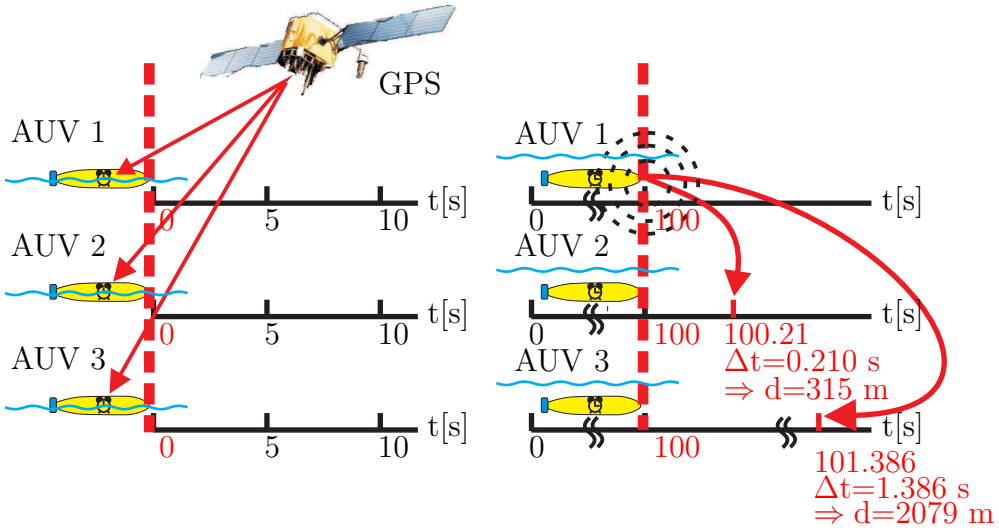


Figure 2-7: Range estimate from globally synchronized data transmission. On the surface all vehicles synchronize their clock to the global GPS-time. When AUV 1 transmits a message it is synchronized to the full second (here t=100 s). AUV 2 receives the message 0.210 s later and uses he measured speed of sound to calculate the distance. AUV 3 receives the message 1.386 s after it was broadcast and uses the time stamp in the message to de-alias.

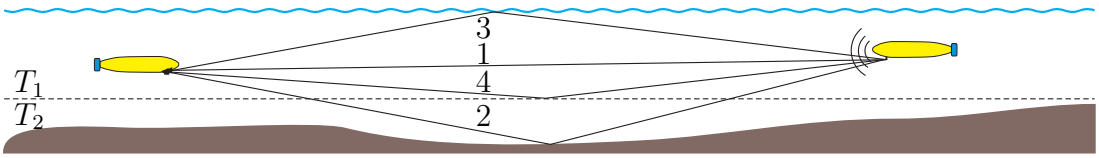


Figure 2-8: Multi-path propagation: in addition to the direct path (1) there are longer paths due to reflection on sea bottom (2), sea surface (3) and a temperature discontinuity (4) (thermocline).

the vehicle's local oscillator and the time-synchronized transmission is illustrated in figure 2-7.

The Acoustic Communication Channel

The speed of sound in water is around 1500 m/s, which is significantly lower than the transmission speed in RF or cable based communication. Additionally the speed depends on the depth as well as the water's temperature and salinity. As a result, discontinuities in temperature and salinity, as well as the sea floor and the air-water interface reflect the pressure waves. For a given physical separation of the transmitter and receiver there often will exist several paths with different lengths along which the pressure wave travels. Figure 2-8 shows four different paths, all with different lengths.

Effects of the Channel on the Communication Performance

The properties of the acoustic communication channel adversely affect data transmission in several ways.

Due to the slow transmission speed, the different path lengths in figure 2-8 will cause interference. For example a symbol s_0 transmitted at time t_0 traveling along path 1 in figure 2-8 will be received at time t_2. Then a symbol s_1 transmitted at time t_1 also traveling along path 1 will collide at the receiver at time t_3 with a "delayed copy" of symbol s_0 which traveled along path 2.

The modem's carrier frequency is directly related to the size of the transducer.Thus frequencies below $\approx$5 kHz are infeasible as the required transducer would be too large for most AUVs. As the attenuation of sound underwater increases linearly with frequency the upper bound of the spectrum which still provides useful communication ranges is approximately 40 kHz. To avoid interference as described above, acoustic modems often cycle through frequency slots transmitting consecutive symbols in different frequency bands. This makes modem communication more robust by avoiding interference, but further limits the usable bandwidth. Due to these bandwidth constraints, data rates for acoustic communication will be limited to $\mathcal{O}$(10 kbps).

Another source of interference is the presence of background noise. Surface waves and marine mammals as well as the noise caused by the vehicle's propulsion and navigation system often occupy the same part of the spectrum as the one used by the acoustic modem.

Chapter 3

Cooperative Localization

3.1 Probabilistic State Estimation

The goal of any localization technique is to maintain an estimate of the vehicle's current pose using information obtained from vehicle sensors and control commands issued to the actuators of the vehicle. As the readings from real sensors are noisy and the effect of a control command cannot be predicted perfectly, the noise needs to be modeled accordingly and the resulting uncertainty in the pose estimate needs to be represented.

3.1.1 State Representation

In the most generic case of a vehicle operating in 3D-space, such as AUVs and UAVs, the state is treated as a vector of random variables comprised of the vehicle's pose which is its position in a (global) reference frame $[x, y, z]^T$ and the orientation. For the rest of the discussion we assume that the orientation is given in Euler angles (ϕ, ψ, θ). The pose vector at time t is then $\boldsymbol{x}(t) = [x, y, z, \phi, \psi, \theta]^T$. Beside the pose, the state vector can also contain the first and second time derivatives of the pose vector. Appendix A illustrates the pose vector of a typical AUV.

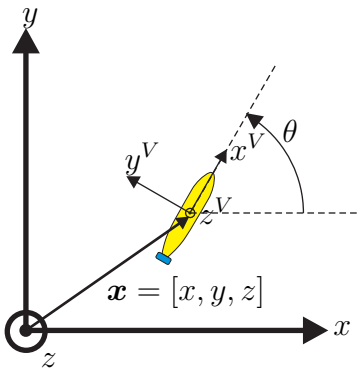
3.1.2 Motion Model

To evaluate the effects of control inputs $\boldsymbol{u}_t$[1] on the pose vector, we need a motion model that can predict a future pose $\boldsymbol{x}_{t_2}$ based on the current pose $\boldsymbol{x}_{t_1}$ and the current control inputs $\boldsymbol{u}_{t_1}$. The continuous-time model for the vehicle state's speed and rate components is

$$\dot{\boldsymbol{x}}_t = f(\boldsymbol{x}_t, \boldsymbol{u}_t) \tag{3.1}$$

The function $f(\cdot)$ in equation 3.1 is usually non-linear and can be very complex. It depends on the vehicle's

[1]For the remainder of section 3.1 we denote the time dependency by a subscript, i.e. $\boldsymbol{x}_t$=$\boldsymbol{x}(t)$



$$\dot{\boldsymbol{x}}_t = \left[u_t, \dot{\theta}_t\right]$$

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} u_t\, cos(\theta_t) \\ u_t\, sin(\theta_t) \\ \dot{\theta} \end{bmatrix}$$

Figure 3-1: Simple AUV motion model

- shape
- size
- weight
- actuators
- operating environment (air, water, vacuum, ...)

The more complex the model, the more accurately it can represent the vehicle dynamics and provide a better prediction of the future pose, but obtaining such a model requires detailed knowledge of the structure as well as the parameters listed above. A large body of literature exists specifically to provide models for AUVs [36],[45],[44],[70],[53].

Figure 3-1 shows a simple kinematic 2D-model for an AUV in which the control input is given by the commanded forward speed $\dot{x}_t = u_t$ and rudder setting leading to a turning rate $\dot{\theta}_t$.

Equation (3.1) represents an ideal motion model. As even the most complex model, however, cannot fully represent a real robot operating in a real environment, uncertainties will remain. In an underwater scenario, these could be currents which cannot be observed. Therefore we need to include a noise term w_t that accounts for these uncertainties. Equation (3.1) then becomes

$$\dot{\boldsymbol{x}}_t = f(\boldsymbol{x}_t, \boldsymbol{u}_t, \boldsymbol{w}_t) \tag{3.2}$$

As the state estimation is normally carried out on a digital computer, we need to operate in discrete time space. We choose a sampling period ΔT small enough such that $\boldsymbol{x}$ and $\boldsymbol{u}$ and can assumed to be constant within that interval. In a real system, the frequency $f_u = \frac{1}{\Delta T}$ is usually the same as the update frequency of the control loop. High-frequency components not captured by f_u are modeled as noise and contribute to $\boldsymbol{w}$. The sampling period is constant and known and we represent the time t_k simply by the index k with $t_k = k \cdot \Delta T$. The discrete version of (3.2) is then

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_{t(k+1)} = f(\boldsymbol{x}_k, \boldsymbol{u}_{k+1}, \boldsymbol{w}_k) \tag{3.3}$$

3.1.3 Measurement Model

Each robot usually carries various sensors that provide information about its own state (proprioceptive) as well as about the environment (exteroceptive). Many of the sensors used on AUVs are described in chapter 2.1.1. Measurements provided by these sensors are denoted z_k^i, where i is the sensor number. We combine the measurements from all sensors at time k into the measurement vector $\boldsymbol{z}_k$. While the proprioceptive measurements only depend on the state of the robot $\boldsymbol{x}_k$, the exteroceptive measurements also depend on the state of the observed feature. We use $\boldsymbol{m}_k^i$ as a description of the feature. A detailed explanation of the structure of $\boldsymbol{m}_k^i$ will be presented in section 3.2.2. As a result the, usually non-linear, function g representing all sensors is dependent on the state of the robot $\boldsymbol{x}_k$ and all observed features $\boldsymbol{m}_k$.

$$\boldsymbol{z}_k = g(\boldsymbol{x}_k, \boldsymbol{m}_k) \tag{3.4}$$

As every real sensor measurement is subject to error (with the characteristics often provided by the sensor's manufacturer), the error needs to modeled. This requires an additional sensor noise term $\boldsymbol{v}_k$ such that the true sensor model becomes

$$\boldsymbol{z}_k = g(\boldsymbol{x}_k, \boldsymbol{m}_k, \boldsymbol{v}_k) \tag{3.5}$$

3.1.4 Probabilistic Representation

In order to solve the problem of localization in the presence of model errors as well as noisy control and measurement data, we choose a probabilistic formulation. The pose $\boldsymbol{x}_t$ as well as the control inputs $\boldsymbol{u}_t$ and the measurement $\boldsymbol{z}_t$ are modeled as random variables. The evolution of the pose $\boldsymbol{x}_t$ is now modeled in two steps. First, the *state transition probability* and second, the *measurement probability.*

State Transition Probability

The state transition probability models the effect of the control inputs $\boldsymbol{u}_t$ on the robot state $\boldsymbol{x}_t$. As the state $\boldsymbol{x}_t$ is stochastically generated from the state $\boldsymbol{x}_{t-1}$ its distribution can be expressed by the conditional probability $p(\boldsymbol{x}_t|\boldsymbol{x}_{0:t-1}, \boldsymbol{z}_{1:t-1}, \boldsymbol{u}_{1:t})$. By properly defining the state, we can enforce conditional independence [84] such that

$$p(\boldsymbol{x}_t|\boldsymbol{x}_{0:t-1}, \boldsymbol{z}_{1:t-1}, \boldsymbol{u}_{1:t}) = p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}, \boldsymbol{u}_t) \tag{3.6}$$

If (3.6) applies, $\boldsymbol{x}_t$ is *complete* [84].

Measurement Probability

Similarly, the the stochastic measurement model can be expressed as a conditional probability $p(\boldsymbol{z}_t|\boldsymbol{x}_{0:t}, \boldsymbol{z}_{1:t-1}, \boldsymbol{u}_{1:t})$ and if $\boldsymbol{x}_t$ is complete we have

$$p(\boldsymbol{z}_t|\boldsymbol{x}_{0:t}, \boldsymbol{z}_{1:t-1}) = p(\boldsymbol{z}_t|\boldsymbol{x}_t) \tag{3.7}$$

3.1.5 Bayes Filter

A general algorithm to track the new state $\boldsymbol{x}_t$ based on a posterior distribution $\boldsymbol{x}_{t-1}$, a control input $\boldsymbol{u}_t$ and observations (measurements) $\boldsymbol{z}_t$ is the Bayes filter. It implements the state evolution

$$p(\boldsymbol{x}_t|\boldsymbol{u}_{t-1}, \boldsymbol{z}_{t-1}) \xrightarrow{\boldsymbol{u}_t, \boldsymbol{z}_t} p(\boldsymbol{x}_{t+1}|\boldsymbol{u}_t, \boldsymbol{z}_t)$$

as a two step process, the *predict* and *update* step shown in algorithm 2.

1: **for** all t **do**
2: $\quad p(\boldsymbol{x}_t|\boldsymbol{u}_t, \boldsymbol{z}_{t-1}, \boldsymbol{x}_{t-1}) = \int p(\boldsymbol{x}_t|\boldsymbol{u}_{t-1}, \boldsymbol{x}_{t-1}) p(\boldsymbol{x}_{t-1}) d\boldsymbol{x}_{t-1}$
3: $\quad p(\boldsymbol{x}_t|\boldsymbol{u}_t, \boldsymbol{z}_t, \boldsymbol{x}_{t-1}) = \eta p(\boldsymbol{z}_t|\boldsymbol{x}_t) p(\boldsymbol{x}_t|\boldsymbol{u}_t, \boldsymbol{z}_t)$
4: **end for**

Algorithm 2: The Bayes filter

In the *predict* step (line 2) the filter processes the control input $\boldsymbol{u}_t$. The probability distribution over the state $\boldsymbol{x}_{t-1}$ is updated by integrating (summing in the discrete case) over the state $\boldsymbol{x}_{t-1}$ times the probability that the control input $\boldsymbol{u}_t$ applied to the motion model causes a transition to $\boldsymbol{x}_t$.

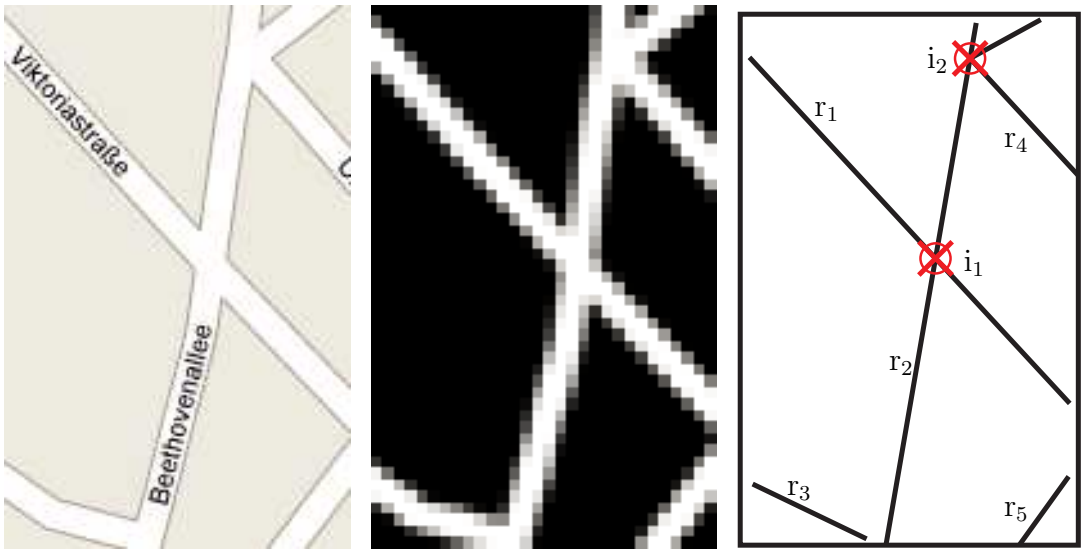
The *update* step (line 3) multiplies the hypothesis of the prior state distribution $\boldsymbol{x}_t$ by the probability that the measurement $\boldsymbol{z}_t$ may have been observed. As the integral over this updated posterior state distribution may not integrate to 1, the normalization factor η is applied [84].

Note that there does not need to be an update step for every prediction step. If no measurements are available, the *update* step will be omitted.

The Bayes algorithm iterates recursively over the *predict* and *update* step. It needs an initial distribution $\boldsymbol{x}_0$. If the initial position is perfectly known, the distribution over $\boldsymbol{x}_0$ is a single realization of the state vector with a probability of 1 or a uniform distribution over the state space if it is perfectly unknown but finite.

3.2 Cooperative Navigation in the Context of SLAM

The problem of localization can be viewed in the more general context of Simultaneous Localization and Mapping (SLAM). As a large body of literature addresses the SLAM problem, by phrasing CN in the context of SLAM, we can leverage the results of research in that area to develop effective localization methods. We will describe the



(a) Human-readable Map (b) Grid-based Map (c) Feature-based Map

Figure 3-2: Three representations of the same environment. A human readable street map (3-2a) and its grid-based (3-2b) and feature-based (3-2c) parametrization.

general formulation of the SLAM problem and a particular representation which is equivalent to the problem of CN. A thorough description of the various aspects of the SLAM problem is beyond the scope of this thesis, but a detailed introduction as well as various implementations of a solution are presented by Thrun [84], Walter [90], Olson [65], Eustice [31] and Frese [38].

3.2.1 General Formulation of the SLAM Problem

The objective of SLAM is to enable a mobile robot to use its sensors to build a map of its environment, while at the same time localizing itself within this map.

As the robot uses noisy sensor readings to build a local map, the map $\boldsymbol{M}$ itself has to be expressed in a probabilistic framework. SLAM addresses this problem by adding the map to the estimated posterior.

$$p(\boldsymbol{x}_{t+1}, \boldsymbol{M} | \boldsymbol{u}_t, \boldsymbol{z}_t) \tag{3.8}$$

3.2.2 Map Representation

To represent the environment sensed by the robot, various parametrizations are possible. The two most common representations are *grid*-based and *feature*-based. Figure illustrates how the same human-readable map (fig. 3-2a) can have the two representations described below (fig. 3-2b and fig. 3-2c).

Grid-based representation

In the most common type of grid-based representation, the environment is discretized into a finite number of cells. The map $\boldsymbol{M}$ then becomes a set of n grid cells, where each element m_i is a binary occupancy value indicating if this cell is occupied (blocked) or free (traversable) and $p(m_i)$ indicates the probability of the cell m_i being occupied. Grid maps assume the probabilities $p(m_i)$ to be independent [84].

$$\boldsymbol{M} = \{m_0, \ldots, m_i, \ldots, m_n\} \qquad m_i = \begin{cases} 1 & \text{“occupied”} \\ 0 & \text{“free”} \end{cases}$$

Figure 3-2b shows a grid-based map representation. The gray-scale value of each cell encodes $p(m_i)$.

Feature-based representation

Feature-based representations parameterize the environment into a set of landmarks, where the type of extracted landmarks is usually constrained by the sensor's capabilities. The map $\boldsymbol{M} = \{\boldsymbol{m}_0, \ldots, \boldsymbol{m}_i, \ldots, \boldsymbol{m}_n\}$ is then a set of vectors $\boldsymbol{m}_i$ as each feature can have several parameters describing it such as position, orientation and color. Each element m_i^j of $\boldsymbol{m}_i = \left[m_i^1, \ldots, m_i^j, \ldots, m_i^n\right]^T$ is a random variable for parameter j.

A special case of a feature is a *point feature* which is assumed to be infinitely small in size and is represented by its position and a unique identifier. The feature vector $\boldsymbol{m}_i$ for a point feature consists of a distribution over its position and the unique id with $p(m_i^j = \mathrm{id}) = 1$.

In figure 3-2c two kind of features were extracted from by the robot, the roads $\mathrm{r}_i = \{\mathrm{r}_1, \ldots, \mathrm{r}_5\}$ and the intersections $\mathrm{i}_i = \{\mathrm{i}_1, \mathrm{i}_2\}$. The intersections are represented as *point features*.

While CN does not explicitly track a feature map $\boldsymbol{M}$ like SLAM does, the following sections will illustrate how the concept of localization in a feature-based map is an adequate description of the CN problem. Also, the problems encountered in feature-based maps, such as the possible correlations of their pose [90], have their correspondence in the CN problem and need to be addressed.

3.3 Probabilistic Representation of CN

After introducing the concept of probabilistic state estimation, its general solution, the Bayes filter, and map representations, we now phrase CN as a localization problem in a feature-based map. The nomenclature introduced in this section will be used throughout the following discussion.

Figure 3-3 shows a set of 5 vehicles $\boldsymbol{V}_i = \{1, \ldots, 5\}$ in the CN setting. All vehicles maintain a distribution of the estimate of their state $\boldsymbol{x}_i$[2]. During a broadcast,

[2]For the remainder of the thesis the subscript will denote a vehicle, i.e. $\boldsymbol{x}_3$ represents the state of vehicle 3.

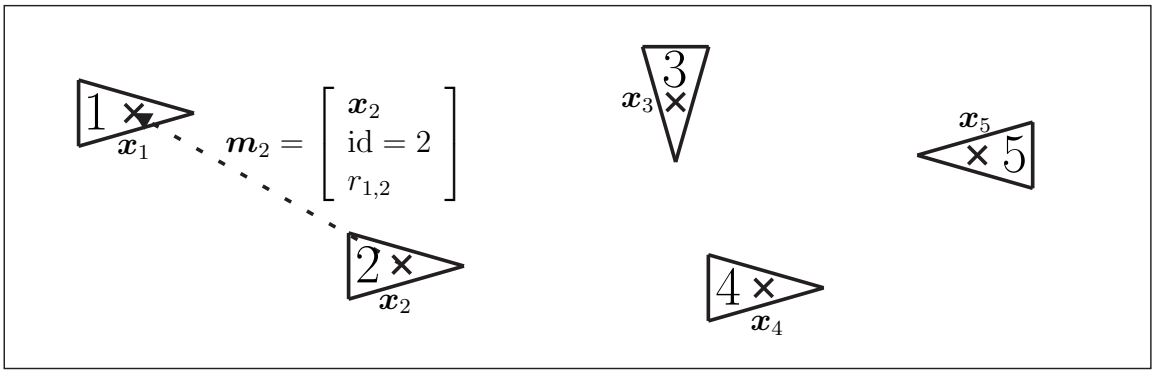


Figure 3-3: A set of 5 vehicles in a CN setting, with each vehicle i maintaining a distribution over its state $\boldsymbol{x}_i$ and obtaining measurements from other vehicles which consist of the state estimate, the unique id of the other vehicle and the range between the two vehicles.

a vehicle (v_2 in fig. 3-3) sends a parametrization of this distribution along with its unique id. The receiving vehicles (only v_1 in fig. 3-3) receive this broadcast and also obtain an intra-vehicle range. Each receiving vehicle treats the reception as the exteroceptive measurement of a known feature represented by the feature vector $\boldsymbol{m}$. At each reception the probability distribution over the transmitting vehicle's pose (here $\boldsymbol{x}_2$) is contained in the data packet and the range to the transmitting vehicle $\boldsymbol{z}_{1,2} = r_{1,2}$ is obtained from onboard sensors (e.g. modem). As a result, no vehicle needs to add a feature map $\boldsymbol{M}$ to its state vector, unlike vehicles performing SLAM where $\boldsymbol{M}$ is part of the posterior (3.8). Just as in SLAM however, measurements can become dependent and we may need to store and transmit additional information in the feature vector in order to integrate the measurements such that these correlations are accounted for. Section 3.6 addresses the problem of dependent measurements.

3.4 Localization Algorithms

In this section, we will introduce the two most popular implementations of the Bayes filter, the *Kalman* and the *Particle* filter, which are widely used for localization and in the SLAM context. The main difference between the filters is how the distribution of the state is represented. The different representations of the distribution also lead to different implementations of the *predict* and *update* step.

We will also examine the computational requirements and the amount of information that needs to be transmitted in the CN scenario outlined in 3.3. The latter is particularly important as the low bandwidth of the acoustic communication channel severely constrains the amount of data that can be transmitted in the $\boldsymbol{m}$.

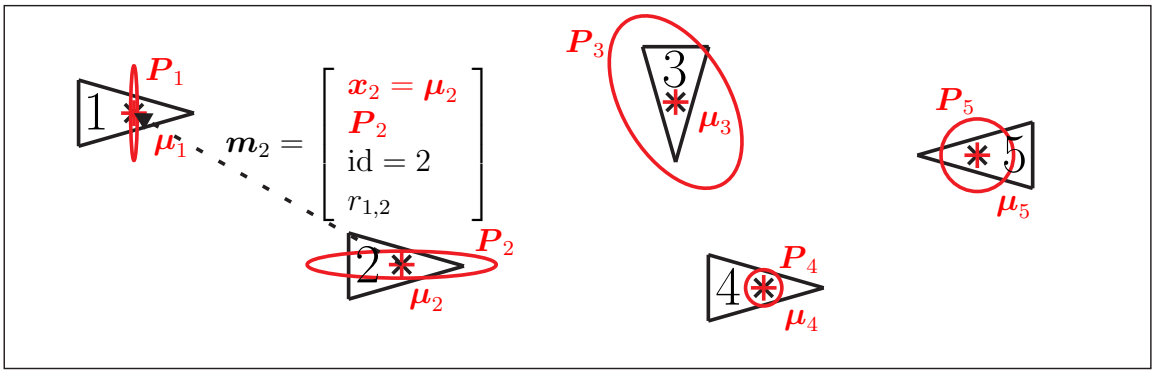


Figure 3-4: A set of 5 vehicles performing CN using an EKF. Each vehicle i maintains the distribution over its state (red) through a mean vector $\boldsymbol{\mu}_i$ and the associated covariance matrix $\boldsymbol{P}_i$. This information, along with the unique id is broadcasted to other vehicles.

3.4.1 Extended Kalman Filter

The original Kalman Filter (KF) [51] requires the function $f(\cdot)$ (3.3) and $g(\cdot)$ (3.5) to be linear, but as even the simple motion model shown in figure 3-1 is non-linear we use the EKF which linearizes equations $f(\cdot)$ and $g(\cdot)$ [81].

The EKF [47] filter is a member of the family of *Gaussian* filters and the probability density over the state $\boldsymbol{x}$ is assumed to be normally distributed. The Gaussian distribution

$$p(\boldsymbol{x}) = \frac{1}{\sqrt{2\,\pi\ |\boldsymbol{P}|}} \exp\left(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^T \boldsymbol{P}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})\right)$$

can be fully described by its mean and covariance. The state $\boldsymbol{x}$ is characterized by a mean vector $\boldsymbol{\mu} \in \mathbb{R}^n$, with the same dimensionality n as the vector $\boldsymbol{x}$, and the covariance matrix $\boldsymbol{P} \in \mathbb{R}^n \times \mathbb{R}^n$ which is positive and semidefinite [84]. As a result only uni-modal distribution centered around $\boldsymbol{\mu}$ can be modeled.

When applying the EKF to solve the problem of CN, we assume that all n vehicles of the set of participating vehicles $\boldsymbol{V}_i = \{1, \ldots, i, \ldots, n\}$ maintain a vector which consists of the mean vector $\boldsymbol{x}_i(k) = [x_i(k), y_i(k), z_i(k)]^T = \boldsymbol{\mu}_i(k) = [\mu_{xi}(k), \mu_{yi}(k), \mu_{zi}(k)]^T$ that contains the an estimate of their position at time k, as well as $\boldsymbol{P}_i$

$$\boldsymbol{P}_i(k) = \begin{bmatrix} {\sigma_{xx}}^2(k) & {\sigma_{xy}}^2(k) & {\sigma_{xz}}^2(k) \\ {\sigma_{yx}}^2(k) & {\sigma_{yy}}^2(k) & {\sigma_{yz}}^2(k) \\ {\sigma_{zx}}^2(k) & {\sigma_{zy}}^2(k) & {\sigma_{zz}}^2(k) \end{bmatrix}$$

the covariance matrix describing the uncertainty associated with that estimate.

In the following description the line numbers refer to algorithm 3.

Prediction

Whenever vehicle $i = 1$ obtains proprioceptive measurements $\boldsymbol{u}_1(k)$ from its dead-reckoning sensors, $\boldsymbol{\mu}_1(k)$ and $\boldsymbol{P}_1(k)$ are propagated[3]

$$\begin{aligned} \overline{\boldsymbol{\mu}}_1(k+1) &= g(\boldsymbol{u}_1(k), \boldsymbol{\mu}_1(k)) && (3.9) \\ \overline{\boldsymbol{P}}_1(k+1) &= \boldsymbol{G}_1(k+1)\boldsymbol{P}_1(k)\boldsymbol{G}_1^T(k+1) \\ &\quad +\boldsymbol{Q}_1(k+1) && (3.10) \end{aligned}$$

where $\boldsymbol{Q}_1(k+1)$ is a matrix where the elements contain the variances of the motion model ($\boldsymbol{w}$ in eq. (3.1)) which is modeled as zero-free Gaussian noise and $\boldsymbol{G}_1(k+1)$ is the Jacobian containing the partial derivatives of g (line 3 and 4).

$$\left.\frac{\partial g(\boldsymbol{u}_1(k+1), \boldsymbol{x}_1(k))}{\partial \boldsymbol{x}_1(k)}\right|_{\boldsymbol{x}_1 = \overline{\boldsymbol{\mu}}_1(k+1)}$$

Update

If vehicle 1 receives a broadcast from vehicle 2 at k that contains $\overline{\boldsymbol{\mu}}_2(l)$ and $\overline{\boldsymbol{P}}_2(l)$ together with an intra-vehicle range measurement $r_{1,2}(k)$ (line 6), it uses this information to update its estimate of its own position as follows:

First, it computes what the predicted range $z_{1,2}(k)$ between the two vehicles would be, based on their estimated position.

$$z_{1,2}(k) = \|\overline{\boldsymbol{\mu}}_1(k) - \overline{\boldsymbol{\mu}}_2(k)\|_2 \tag{3.11}$$

The difference between the predicted measurement and the measured distance $z_{1,2}(k) - r_{1,2}(k)$ represents the innovation (line 7).

The covariance matrix of vehicle 1 and vehicle 2 are combined (line 8) into

$$\overline{\boldsymbol{P}}_{1,2}(k+1) = \begin{bmatrix} \overline{\boldsymbol{P}}_1(k+1) & 0 \\ 0 & \overline{\boldsymbol{P}}_2(k+1) \end{bmatrix}. \tag{3.12}$$

Note that $\overline{\boldsymbol{P}}_1(k+1)$ and $\overline{\boldsymbol{P}}_2(k+1)$ are assumed to be deindependent ($\overline{\boldsymbol{P}}_{1,2}(k+1)$ is diagonal). This is not generally true and if the non-zero off-diagonal elements of $\overline{\boldsymbol{P}}_{1,2}(k+1)$ are ignored, the EKF can become overconfident and diverge. As keeping track of these elements in CN is very difficult, however, we propose a method in section 3.6 which keeps $\overline{\boldsymbol{P}}_1(k+1)$ and $\overline{\boldsymbol{P}}_2(k+1)$ independent.

We compute the Jacobian $\boldsymbol{H}_{1,2}(k+1)$ that contains the derivatives of the range measurement with respect to the position of vehicle 1 and 2 (time index k omitted on matrix components) (line 9).

$$\boldsymbol{H}_{1,2}(k+1) = \begin{bmatrix} \frac{\partial r}{\partial \overline{\mu}_{x_1}} & \frac{\partial r}{\partial \overline{\mu}_{y_1}} & \frac{\partial r}{\partial \overline{z}_1} & \frac{\partial r}{\partial \overline{\mu}_{x_2}} & \frac{\partial r}{\partial \overline{\mu}_{y_2}} & \frac{\partial r}{\partial \overline{z}_2} \end{bmatrix}$$

[3] $\overline{\boldsymbol{\mu}}$ and $\overline{\boldsymbol{P}}$ denote the state *after* the predict step, but *before* the update step.

Using the residual covariance (line 10) and the variance

$$\boldsymbol{S}_{1,2}(k+1) = \boldsymbol{H}_{1,2}(k+1)\overline{\boldsymbol{P}}_{1,2}(k+1)\boldsymbol{H}_{1,2}^{T}(k+1) + \sigma_r^2$$

and σ_r associated with the exteroceptive (range) sensor we compute the Kalman gain (line 11)

$$\boldsymbol{K}_{1,2}(k+1) = \overline{\boldsymbol{P}}_{1,2}(k+1)\boldsymbol{H}_{1,2}^{T}(k)\boldsymbol{S}_{1,2}^{-1}(k+1)$$

that represents a weighting factor for how much the measurement will affect the updated position. Using the innovation $z_{1,2}(k) - r_{1,2}(k)$ and the Kalman gain, the updated position estimate is

$$\boldsymbol{\mu}_1(k+1) = \overline{\boldsymbol{\mu}}_1(k+1) + \boldsymbol{K}_{1,2}(k+1)\Big(z_{1,2}(k) - r_{1,2}(k)\Big) \tag{3.13}$$

(line 12) and the combined covariance is

$$\begin{aligned} \boldsymbol{P}_{1,2}(k+1) &= \begin{bmatrix} \boldsymbol{P}_1(k+1) & \boldsymbol{P}_{12}(k+1) \\ \boldsymbol{P}_{21}(k+1) & \boldsymbol{P}_2(k+1) \end{bmatrix} \\ &= \Big(\boldsymbol{I}_{6\times 6} - \boldsymbol{K}_{1,2}(k+1)\boldsymbol{H}_{1,2}(k)\Big)\overline{\boldsymbol{P}}_{1,2}(k+1) \end{aligned} \tag{3.14}$$

from which we can extract the updated covariance estimate for vehicle 1 $\boldsymbol{P}_1(k+1)$. Note that we also obtain an updated estimate for the position and covariance of vehicle 2 $\boldsymbol{P}_2(k+1)$ and $\boldsymbol{\mu}_2(k+1)$.

Characteristics

Under the assumption that the initial state error as well as all noises entering into the system have a Gaussian distribution and that the underlying model is linear, the KF is the optimal estimator in the sense that it minimizes the Minimum Mean Squared Error (MMSE) [10]. As a result and because of its simplicity, tractability and robustness, the EKF is the most common algorithm used today for tracking and estimation. For the particular case of underwater CN, several key assumptions necessary to guarantee optimality in the MMSE-sense cannot be made and can lead to very large position estimation errors.

Non-linear Motion and Measurement Model Even the most basic motion model, represented by f in (3.3) which maps body velocities into a global coordinate frame has non-linear components (see fig. 3-1). Similarly the measurement model g (3.5) used in the update step (3.11) is non-linear. These non-linearities can lead the EKF to be unstable [48].

Non-Gaussian Noise The sensors used for underwater navigation often have very non-Gaussian noise characteristics. Particularly the various components contributing to the error in a heading measurement derived from a magnetic com-

Require: $\boldsymbol{\mu}_1(0)$,$\boldsymbol{P}_1(0)$

$\boldsymbol{\mu}_1(0), \boldsymbol{P}_1(0)$

loop {do this at each time step k}

$\overline{\boldsymbol{\mu}}_1(k+1) = g(\boldsymbol{u}_1(k), \boldsymbol{\mu}_1(k))$

$\overline{\boldsymbol{P}}_1(k+1) = \boldsymbol{G}_1(k+1)\boldsymbol{P}_1(k)\boldsymbol{G}_1^T(k+1) + \boldsymbol{Q}_1(k+1)$

if measurement $\boldsymbol{m}_2(k)$ is available **then**

$$\boldsymbol{m}_2(k) = \begin{bmatrix} \boldsymbol{\mu}_2(k) \\ \boldsymbol{P}_2(k) \\ \text{id} = 2 \\ r_{1,2} \end{bmatrix}$$

$z_{1,2}(k) = \|\overline{\boldsymbol{\mu}}_1(k) - \overline{\boldsymbol{\mu}}_2(k)\|_2$

$$\overline{\boldsymbol{P}}_{1,2}(k+1) = \begin{bmatrix} \overline{\boldsymbol{P}}_1(k+1) & 0 \\ 0 & \overline{\boldsymbol{P}}_2(k+1) \end{bmatrix}$$

$$\boldsymbol{H}_{1,2}(k) = \begin{bmatrix} \frac{\partial r}{\partial \overline{\mu}_1} & \frac{\partial r}{\partial \overline{y}_1} & \frac{\partial r}{\partial \overline{z}_1} & \frac{\partial r}{\partial \overline{\mu}_2} & \frac{\partial r}{\partial \overline{y}_2} & \frac{\partial r}{\partial \overline{z}_2} \end{bmatrix}$$

$\boldsymbol{S}_{1,2}(k+1) = \boldsymbol{H}_{1,2}(k)\overline{\boldsymbol{P}}_{1,2}(k+1)\boldsymbol{H}_{1,2}^T(k) + \sigma_r$

$\boldsymbol{K}_{1,2}(k+1) = \overline{\boldsymbol{P}}_{1,2}(k+1)\boldsymbol{H}_{1,2}^T(k)\boldsymbol{S}_{1,2}^{-1}(k+1)$

$\boldsymbol{\mu}_1(k+1) = \overline{\boldsymbol{\mu}}_1(k+1) + \boldsymbol{K}_{1,2}(k+1)\Big(z_{1,2}(k) - r_{1,2}(k)\Big)$

$$\begin{aligned} \boldsymbol{P}_{1,2}(k+1) &= \begin{bmatrix} \boldsymbol{P}_1(k+1) & \boldsymbol{P}_{12}(k+1) \\ \boldsymbol{P}_{21}(k+1) & \boldsymbol{P}_2(k+1) \end{bmatrix} \\ &= \Big(\boldsymbol{I}_{6\times 6} - \boldsymbol{K}_{1,2}(k+1)\boldsymbol{H}_{1,2}(k)\Big)\overline{\boldsymbol{P}}_{1,2}(k+1) \end{aligned}$$

end if

end loop

Algorithm 3: EKF algorithm for a vehicle 1 initialized with a pose $\boldsymbol{\mu}_1(0)$ and associated uncertainty $\boldsymbol{P}_1(0)$, moving in 2D by executing control commands and occasionally receiving range/position pairs $\boldsymbol{m}_2(k)$ from another vehicle 2.

pass (see section 2.1.1) are hard to fit to a Gaussian noise model. For exteroceptive measurements Olson et al. show that for LBL beacons (see chapter 2.1.1) Gaussian noise is a very poor approximation due to the large number of outliers [67].

Uni-modal probability distribution As the probability distribution for the EKF is only modeled by the mean and covariance of a single Gaussian, the Kalman filter can only track a single hypothesis of the state vector. If the initial state $\boldsymbol{\mu}(0)$ is not known and needs to be represented by a uniform distribution over a finite state space or the initial information about the state contains ambiguity, the resulting distributions cannot be tracked by an EKF.

Computation and Bandwidth Requirements

The predict step of the EKF is computationally efficient for very large state vectors, but the computation and memory requirement for the update step grows with $\mathcal{O}(n^2)$

where n is the size of the state vector. Fortunately the state vector for underwater CN is small and updates are very infrequent $\mathcal{O}(0.1\,\text{Hz})$. As a result, the computational requirements on a vehicle performing EKF-based CN are very small. Additionally the compact formulation of the state distribution leads to a message size of only a few bytes, which is well suited for the low-bandwidth channel.

3.4.2 Particle Filter

The Particle Filter [26] is a member of the family known as *Monte Carlo* methods and fundamentally differs from the EKF in the way the probability distribution over the state is represented. The distribution is represented by a fixed number (n) of samples from the distribution. Each sample $\boldsymbol{c}_i$ of set $\boldsymbol{C} = \{\boldsymbol{c}_1, \ldots, \boldsymbol{c}_i, \ldots, \boldsymbol{c}_n\}$ consists of an instantiation of the state vector $\boldsymbol{x}_i$ and an associated weight w_i.

$$\boldsymbol{c}_i = [\boldsymbol{x}_i, w_i] \qquad i = [1, \ldots, n]$$

The probability distribution is recursively updated by a two step process. First the motion model is applied, then the measurement update. At each time step every particle is affected.

Several variations of the algorithm exist and we show one possible implementation particularly addressing the CN scenario outlined in 3.3. Here each vehicle maintains its own particle filter only to track its own state.

The following section explains the individual steps and the line indications refer to algorithm 4.

Initialization

The set $\boldsymbol{C}(0)$ which we use to initialize our PF is generated by drawing n samples $\boldsymbol{x}_i$ with $i = [1, \ldots, n]$ from the distribution representing the initial assumption about our state $\mathcal{X}$ (line 2). The distribution function can be completely arbitrary. We can sample from a uniform distribution over the entire state space, as long as it is finite, in case we are ignorant of our initial state, or all n samples are the same instantiation of the state vector in case our initial state is perfectly known. Independent of the distribution function from which we are sampling from, we assign an equal weight to all particles $w_i = \frac{1}{n} \quad \forall i$ (line 3).

Prediction

For a single prediction step, the motion model is applied to each particle c_i individually.

In order to update a single particle we draw a sample from the individual distribution of each variable from the control space $\mathcal{U}$ (line 8) and apply the motion model using these samples (line 9). Note that the distribution over state variables can be different for each component of the state vector and arbitrary.

Update

If a measurement $\boldsymbol{m}^j(k) = \left[\boldsymbol{C}^j(k), \mathrm{id} = j, r_{j(k)}\right]$, as shown in figure 3-5, is available we evaluate the function $e(\boldsymbol{x}_i(k), \boldsymbol{m}^j(k))$ for each particle. The function $e(.)$ computes a likelihood of each particle $\boldsymbol{x}_i(k)$ belonging to the distribution represented by $\boldsymbol{C}^j$ transformed along r_j.

A possible implementation of $e(\boldsymbol{x}_i, \boldsymbol{m}^j(k))$ is to compute the weighted average $\boldsymbol{\mu}^j(k)$ of all particles $\boldsymbol{c}_i^j$ from the distribution $\boldsymbol{C}^j(k)$

$$\boldsymbol{\mu}^j(k) = \sum_{i=1}^{n} \boldsymbol{x}_i^j(k)\, w_i^j(k) \tag{3.15}$$

and assume a normal distribution of the range error with $\tilde{r} = \mathcal{N}(0, \sigma_r)$. The likelihood then becomes

$$w_i(k) = e(\boldsymbol{x}_i(k), \boldsymbol{\mu}(k)^j) = p(\boldsymbol{x}_i(k)) = \frac{1}{\sqrt{2\,\pi\,\sigma_r}} \exp\left(-\frac{1}{2}\frac{(\|\boldsymbol{x}_i - \boldsymbol{\mu}^j\|_2 - r_j)^2}{\sigma_r}\right) \;\forall\; \boldsymbol{x}_i(k)$$

Note that only the weights $w_i(k)$ are updated, not the particle positions. As the sum of all weights in the set does not necessarily add up to 1 the particle weights are renormalized to enforce $\sum_{i=1}^{n} w_i = 1$ (line 14).

Resample

Each update reevaluates for all particles how likely the hypothesis is that the vehicle's state is $\boldsymbol{x}_i$ as represented by the particle $\boldsymbol{c}_i$. This likelihood is represented by the weight w_i. After a series of updates, the distribution of weights across the particles can become very uneven which means that the set $\boldsymbol{C}_i$ has very strong hypotheses (high weight) and many very weak hypotheses (low weight). The goal of the particle filter, to track the distribution of the state through a set of instantiations, is best achieved if all samples have very similar weights.

To maintain an even distribution of the particle weights we determine the number of "effective particles" $N_{\text{eff}} = [0, 1]$ with

$$N_{\text{eff}} = \frac{1}{n\ \sum_{i=1}^{n} w_i^2}$$

which is a measure of how well the weights are distributed. If N_{eff} drops below a threshold γ, we resample by building a new set $\boldsymbol{C}(k)$ by drawing n samples from $\boldsymbol{C}(k-1)$ where each particle $\boldsymbol{c}_i(k-1)$ is drawn with a likelihood $w_i(k-1)$ (line 20). The new set $\boldsymbol{C}(k)$ now represents single strong particles from $\boldsymbol{C}(k-1)$ by a number of equally weighted particles in $\boldsymbol{C}(k)$ while very weak particles might not be drawn at all, they "die" (line 17). This is a problem particular to the PF.

1: **for** $i = 1{:}n$ **do** {initialize all particles}
2: $\quad \boldsymbol{x}_i(0) = \begin{bmatrix} x_i(0) & = & \text{sample}(\mathcal{X}_x) \\ y_i(0) & = & \text{sample}(\mathcal{X}_y) \\ \theta_i(0) & = & \text{sample}(\mathcal{X}_\theta) \end{bmatrix}$
3: $\quad w_i(0) = \frac{1}{n}$
4: $\quad c_i(0) = [\boldsymbol{x}_i(0), w_i(0)]$
5: **end for**
6: **loop** {do this at each time step k}
7: $\quad$ **for** $i = 1{:}n$ **do** {predict}
8: $\quad\quad \boldsymbol{u}_i(k) = \begin{bmatrix} u_i(k) & = & \text{sample}(\mathcal{U}_u) \\ \theta_i(k) & = & \text{sample}(\mathcal{U}_\theta) \end{bmatrix}$
9: $\quad\quad \boldsymbol{x}_i(k) = \begin{bmatrix} x_i(k) & = & x_i(k-1) + u_i(k)\, cos(\theta_i(k)) \\ y_i(k) & = & x_i(k-1) + u_i(k)\, sin(\theta_i(k)) \\ \theta_i(k) & = & \theta_i(k) \end{bmatrix}$
10: $\quad\quad c_i(k) = [\boldsymbol{x}_i(k), w_i(k-1)]$
11: $\quad$ **end for**
12: $\quad$ **if** measurement $\boldsymbol{m}(k)$ is available **then**
13: $\quad\quad$ **for** $i = 1{:}n$ **do** {update}
14: $\quad\quad\quad w_i(k) = e(\boldsymbol{x}_i, \boldsymbol{m}(k))$
15: $\quad\quad$ **end for**
16: $\quad\quad$ **for** $i = 1 : n$ **do** {normalize particle weights}
17: $\quad\quad\quad w_i(k) = w_i(k) \frac{1}{\sum_{i=1}^{n} w_i(k)}$
18: $\quad\quad$ **end for**
19: $\quad\quad$ **if** $\mathrm{N}_{eff} < \gamma$ **then** {resample if necessary}
20: $\quad\quad\quad \boldsymbol{C}(k) = \text{resample}(\boldsymbol{C}(k-1))$
21: $\quad\quad$ **end if**
22: $\quad$ **end if**
23: **end loop**

Algorithm 4: PF algorithm for a vehicle initialized with a pose sampled from $\mathcal{X}$, moving in 2D by receiving control commands $\boldsymbol{u}_1(k)$ sampled from $\mathcal{U}$ and occasionally receiving range/position pairs.

Characteristics

By representing arbitrary probability distributions over the state and to incorporate non-linear motion and measurement models without linearization, the PF avoids two of the main disadvantages of the EKF. The PF only maintains a finite set of discrete hypotheses about the vehicle's position, unlike the Gaussian distribution in the EKF which is continuous and covers the entire state space. To avoid "particle depletion", the absence of particles near the true solution, the weighting function $e(.)$ (line 14 in algorithm 4) must be carefully chosen. Other techniques add random samples [84] or add additional noise to the samples [34]. It is important to note that while the EKF inherently provides a "best guess" for the actual position (the mean), the PF does not. A control algorithm on a vehicle however, that is tasked to guide the vehicle

to a specific location, normally requires an estimate which is a single instantiation of the state distribution. A simple way to provide a single position estimate which can be used by the control algorithm of the vehicle, is to compute the weighted mean of the distribution over the state space (3.15) along with a weighted covariance (3.16).

$$\sigma(k) = \sum_{i=1}^{n} \left(\boldsymbol{x}_i(k) - \boldsymbol{\mu}(k)\right)^2 \; w_i(k) \tag{3.16}$$

A more complex method first discretizes the space, where the likeliest position is assumed to be, into grid points. It then instantiates a Gaussian distribution on every particle and sums up the contributions of all Gaussians to the set of grid points. The grid point with the the highest combined contribution is the filter's estimate of the actual position [32].

Computation and Bandwidth Requirements

The computational complexity is linear in the number of particles. The computation required to carry out the update and the predict step for each particle depends entirely on the complexity of the motion and measurement model respectively, but with the scenario given in typical underwater-CN application PFs with tens of thousands of particles are feasible. Transmitting the entire state space distribution ($\boldsymbol{C}_2$ in figure 3-5) for CN has been done by Fox et al. [37], but given the low bandwidth available in underwater communication, this is only feasible for a very small number of particles. Therefore, in order to apply the PF-based CN underwater the distribution needs to be parameterized before being transmitted to the other vehicle. If the distribution is parametrized by computing the weighted mean (3.15) and covariance (3.16) the transmitted data is the same as in the EKF case. This is the most compact way to represent the distribution which makes it particularly suited for low bandwidth communication at the expense of being able to model multi-modal distributions. One way to maintain the multi-modality of the transmitted distribution, while maintaining a small set of information which needs to be transmitted, is to represent it as a Gaussian Mixture Model (GMM) as proposed by Merwe and Wan [59].

3.5 Multi-Robot Localization

As AUVs have only recently become reliable enough to allow for multi-vehicle deployments, only a small number of experiments involving more than one vehicle have been carried out. However a much larger body of literature for land vehicles and, to a lesser degree for air vehicles, exists. The following two sections provide an overview over the previous work carried out in the field of cooperative localization for land and air (3.5.1) and underwater vehicles (3.5.2).

We then present our CN-algorithm in section 3.5.3. It is particularly designed to work with the navigation information available on AUVs and the infrequent broadcasts obtained from other vehicles through the underwater communication channel.

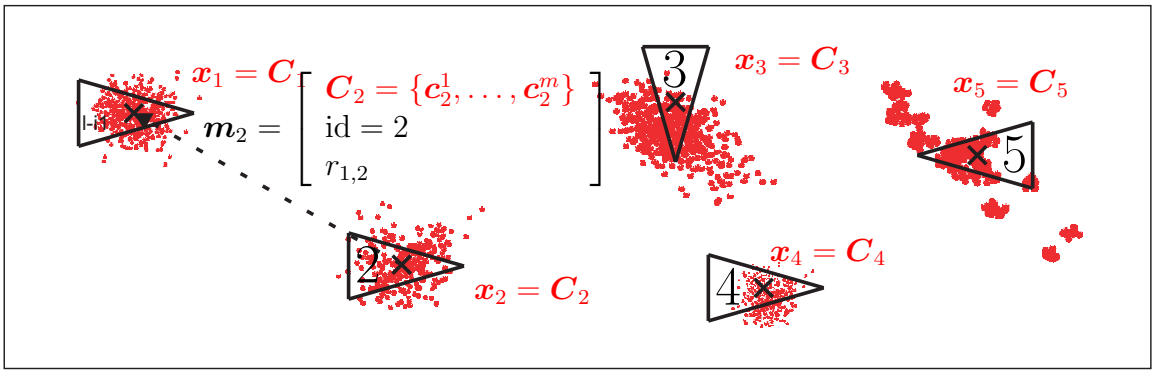


Figure 3-5: A set of 5 vehicles performing CN using a Particle Filter. Each vehicle i maintains the distribution over its state (red) through a set of particles $\boldsymbol{C}_i = \{c_i^1, \ldots, c_i^m\}$. This information, along with the unique id is broadcasted to other vehicles.

3.5.1 Land and Air Vehicles

Roumeliotis et al. have contributed a large body of work to the field of CN using indoor land robots. Early work relies on a central site for data storage and processing [72]. With this setup, the authors make useful insights into the relationship between the number of cooperating robots and the individual position uncertainty. The result is an analytical expression for an upper bound on the growth rate of the overall positioning uncertainty for the group [75]. In another experiment, the central filter that keeps track of the state and covariance of all vehicles is replaced by distributed filters that run on the individual members. Agents now only need to exchange local data, but as both vehicles are required to transmit, this approach does not scale as well as others that rely only on one-way broadcasts [73]. Caglioti et al. [19] also use a distributed filter approach. While they only require one-way data exchange (broadcast), these broadcasts occur very frequently and their method relies on perfect communication as each vehicle is required to receive every broadcast for the successful application of their method.

Nettleton uses a group of UAVs to build a map of observed features locally on each vehicle while relying only on broadcast traffic without the requirement that each vehicle receives all transmissions [62]. While the UAVs do not exchange information for navigation the cooperative map building provides insights into the problem of fusing information from mobile platforms using an unreliable communication channel. The problem of fusing measurements from several sources while properly keeping track of common information has been addressed by Grime [43]. Unlike the work of Roumeliotis et al. Grime and Nettleton's work tracks the information parametrization of the Gaussian rather than the standard form. In the information form, the update step is simply an addition, and joint information, which models co-dependencies among different states, can be subtracted if the communication topology is known. As the update step is computationally inexpensive the information form is well suited to integrate large amounts of information from cooperating vehicles [90].

The previous work presented thus far relies on the KF/EKF, or its inverse, the information filter, to compute an estimate. Fox et al. [37] use a PF to perform cooperative localization. They represent the distribution by a large number of samples rather than as a Gaussian distribution. Section 3.4.2 outlines the advantages of the PF over the KF. However transferring the distribution consisting of many particles between vehicles requires a comparably fast communication channel which is available on land with RF communication, but the slow acoustic communication channel limits the applicability of this approach underwater. Additionally Fox's approach also requires (half-)duplex communication such that broadcast-based approaches cannot be used, which further increases the necessary bandwidth.

3.5.2 Underwater Vehicles

Eickstedt and Schmidt [29] proposed deploying two AUVs equipped with an active sonar in which the sources were synchronized to a global clock and were transmitting orthogonal chirp sequences. This enabled one vehicle to use the ping emitted by the other to perform acoustic tomography and bi-static target detection. While no data was exchanged through the acoustic channel, it was one of the first times globally synchronized transceivers could be used for intra-vehicle range measurements underwater. Leonard et al. [57] and Paley et al. [69] used a fleet of gliders to jointly survey a large body of water. When on the surface, the gliders connected to a central computer and offloaded the data they collected during their last dive as well as their actual GPS-derived position. The data and the position information was used by the central computer to calculate an individual track for each glider which allowed optimal data collection. This track was sent to the glider via satellite and each vehicle was unaware of all others. No data was exchanged while they were submerged.

The idea for CN was mostly inspired by earlier work done in single-beacon localization. Single-beacon navigation uses a single LBL beacon instead of a network of three or more as shown in section 2.1.1 to obtain a position estimate. While classic LBL provides an absolute position every time the beacon network is queried, single-beacon navigation requires several range measurements obtained from the same beacon combined with dead-reckoning information collected between range measurements. The earliest work in the field was presented by Scherbatyuk [76]. He used a Least Squares (LSQ) algorithm to combine three or more range readings with dead-reckoning information to solve for a position in the horizontal plane while the depth was provided directly through a pressure gauge. Vaganay et al. used an EKF similar to the one presented in section 3.4.1 to initially solve the "homing problem" in which a vehicle attempts to get as close as possible to a point which is marked by an acoustic beacon [87]. Baccou and Jouvencel later applied that approach to general vehicle navigation [5],[6]. Another EKF-based single beacon approach called *Synthetic Long Baseline (SLBL)* was presented by Larsen [56].

Combining the concept of a mobile LBL-beacon [24], which not only provides a range to the interrogating vehicle but also its position, with the idea of using consecutive range measurements combined with dead-reckoning information led to

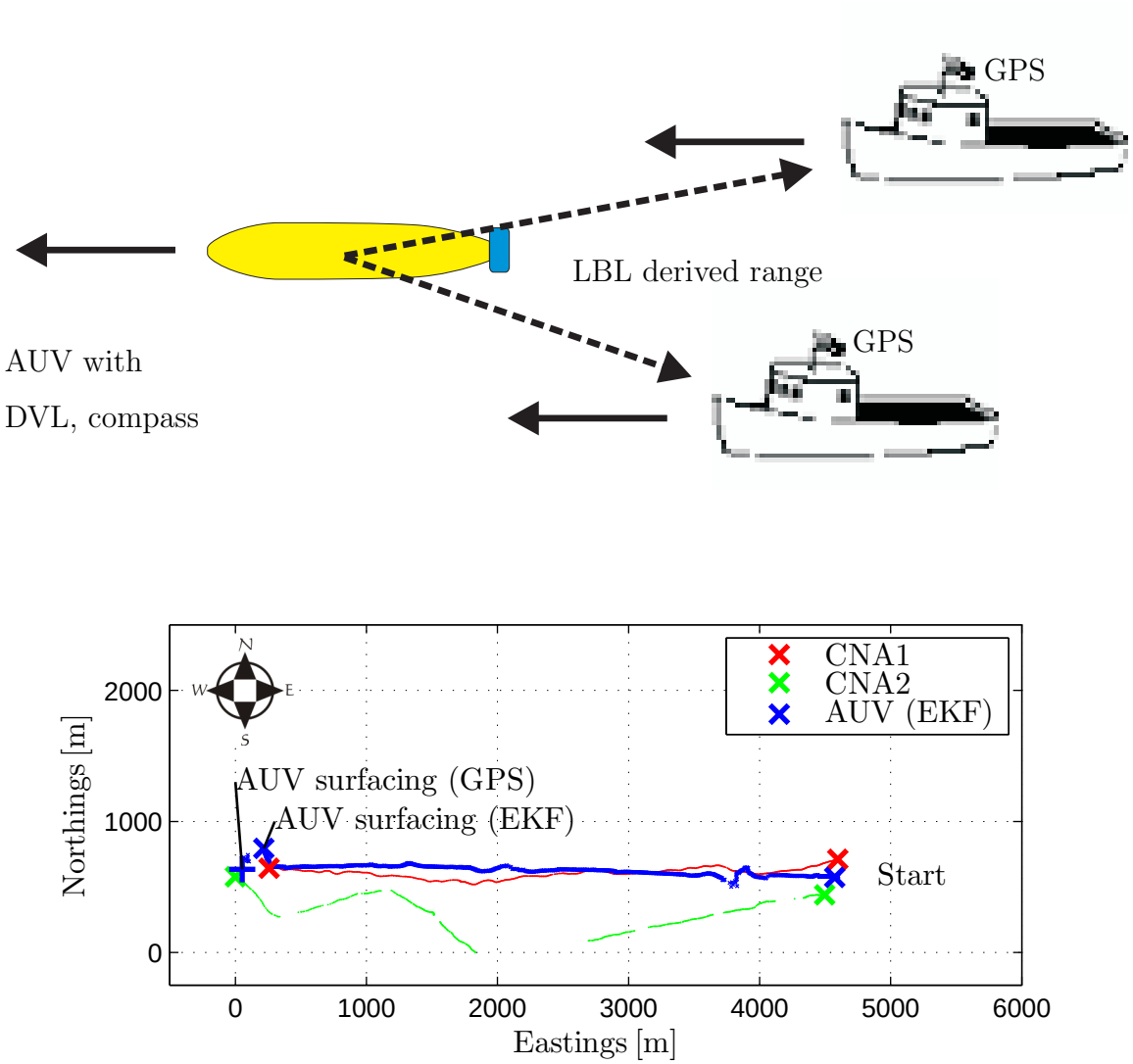


Figure 3-6: Setup and results of the AOFNC2003 MLBL experiment. Top: Setup. Two boats outfitted with GPS and LBL beacons and an AUV (with DVL and compass) interrogating both beacons. Bottom: GPS-track of the boats (CNAs) and the AUV's position estimate using an EKF. Note the difference between the estimated ($\times$) and the true surfacing position ($+$).

the concept of Moving Long Baseline (MLBL). This concept was first expressed by Vagany et al. [88]. They used a single AUV outfitted with an AVTRAK LBL interrogator and two boats, each carrying a conventional LBL beacon (figure 3-6). The AUV was sent on a pre-programmed mission and interrogated both ship-side beacons every 3 seconds. While the AUV logged the TOF/range, the boats, serving as beacons vehicles, logged their GPS-derived position at the time their beacons were interrogated. By combining the log-files from the AUV and the two boats, the pure DR track from the AUV was corrected with the MLBL-algorithm. The algorithm

used is proprietary and no information on how the range/position pairs were fused with the DR track is available.

3.5.3 The CN-Algorithm

Due to the shortcomings of the two classical approaches described in section 3.4 we propose our CN algorithm. The CN algorithm uses consecutive range measurements which it forward propagates to align in time. The intersections of the range circles are hypotheses for possible positions. By combining the latest range measurement with several range measurements forward propagated from the past, the CN algorithm obtains a number of hypotheses for the AUV's position. Using available dead-reckoning information it searches for the most consistent path through the sets of hypotheses generated whenever a new range measurement was received. Unlike the EKF it inherently maintains several hypotheses about its position which are all reevaluated whenever a new range measurement is available. This allows it to recover after a measurement with a large error was received. The individual intersections are similar to the particles in a PF in that they represent hypotheses about the vehicle's position. Unlike the particles in the PF however they are not sampled, but represent the outcome of an intersection. As intersections will always instantiate hypotheses near the true position (assuming a bounded error) this algorithm will not suffer from "particle depletion" close to the correct position which is possible with the PF. The CN algorithm works as follows.

With each successful transmission at time k the AUV receives an estimate of the CNA's position $\boldsymbol{x}^C(k) = [x^C(k), y^C(k)]^T$, the covariance matrix, $\boldsymbol{P}^C(k)$, which accounts for the confidence the CNA has in each component of $\boldsymbol{x}^C(k)$, a depth $z^C(k)$ and the range $r(k)$ between the AUV and the CNA.

$$\boldsymbol{P}^C(k) = \begin{bmatrix} {\sigma_{xx}^C}^2(k) & {\sigma_{xy}^C}^2(k) \\ {\sigma_{yx}^C}^2(k) & {\sigma_{yy}^C}^2(k) \end{bmatrix}$$

The state $\boldsymbol{x}^C(k)$ and the covariance $\boldsymbol{P}^C(k)$ can be a snapshot from the navigation filter running on the CNA or from the GPS in the case that the CNA is at the surface. The range $r(k)$ is directly obtained by the AUV through the PPS-synchronized transmission feature (see section 2.2.2). Many experiments have shown that the error in the range measurement $r(k)$ is only weakly range-dependent and can be modeled as a Gaussian with mean $r(k)$ and a fixed variance σ_r^2.

Furthermore, the AUV builds a matrix $\boldsymbol{D}$ where each entry $\boldsymbol{D}(n, m)$ contains the distance traveled $\boldsymbol{d}_{\overline{n,m}} = [dx_{\overline{n,m}}, dy_{\overline{n,m}}]^T$ between receiving a transmission at $t(n)$ and at $t(m)$ as obtained from proprioceptive measurements as well as the covariance matrix $\boldsymbol{Q}_{\overline{n,m}}$ associated with that measurement.

$$\boldsymbol{Q}_{\overline{n,m}} = \begin{bmatrix} {\sigma_{dx_{\overline{n,m}}}}^2 & 0 \\ 0 & {\sigma_{dy_{\overline{n,m}}}}^2 \end{bmatrix}$$

Figure 3-7 shows how the AUV uses information received at $t(n)$ and $t(m)$ to compute two possible solutions for its position at $t(m)$: The circle with radius $r(n)$

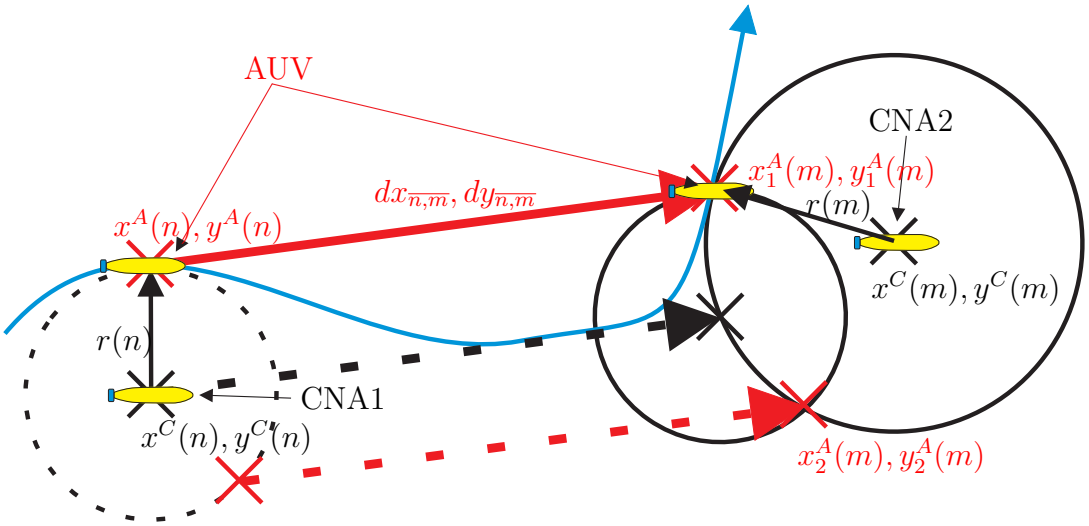


Figure 3-7: Computing two possible positions of the AUV at $t(m)$ $(x_1^A(m), y_1^A(m)$ and $x_2^A(m), y_2^A(m))$ using the dead-reckoning information $dx_{\overline{n,m}}, dy_{\overline{n,m}}$ and the information $x^C(n), y^C(n)$ and $r(n)$ received at $t(n)$ from CNA1 and $x^C(m), y^C(m)$ and $r(m)$ received at $t(m)$ from CNA2.

defines all possible positions at $t(n)$. Shifting the center of this circle by $[dx_{\overline{n,m}}, dy_{\overline{n,m}}]^T$ (dashed black line in figure 3-7) and solving the resulting quadratic equation, we obtain a set $\boldsymbol{X}^A(m)$ of 0, 1 or 2 intersections with the circle around $\boldsymbol{x}^C(m)$ with radius $r(m)$.

$$\boldsymbol{X}^A(m) = \mathcal{F}(\boldsymbol{x}(n)^C, \boldsymbol{x}(m)^C, r(n), r(m), \boldsymbol{d}_{\overline{n,m}}) \tag{3.17}$$

with

$$\boldsymbol{X}^A(m) = \emptyset \quad \text{or} \quad \boldsymbol{X}^A(m) = \boldsymbol{x}_1^A(m) \quad \text{or} \quad \boldsymbol{X}^A(m) = \begin{pmatrix} \boldsymbol{x}_1^A(m) \\ \boldsymbol{x}_2^A(m) \end{pmatrix}$$

Using other values for n $(n = [1, \ldots, m-1])$, we can compute up to $2(m-1)$ solutions for $\boldsymbol{x}^A(m)$. For the upcoming computations we assume that we use q solutions. The Jacobian of the intersection function $\mathcal{F}$ with respect to the measured and transmitted parameters $\boldsymbol{x}^C(n)$, $\boldsymbol{x}^C(m)$, $r(n)$, $r(m)$, $\boldsymbol{d}_{\overline{n,m}}$ is $\boldsymbol{J}_{n,m}$ and can be used to compute $\boldsymbol{P}^A(m)$ the covariance of $\boldsymbol{x}^A(m)$. $\boldsymbol{P}^A(m)$ is given by

$$\boldsymbol{P}^A(m) = \begin{bmatrix} {\sigma_{xx}^A}^2(m) & {\sigma_{xy}^A}^2(m) \\ {\sigma_{yx}^A}^2(m) & {\sigma_{yy}^A}^2(m) \end{bmatrix} = \boldsymbol{J}_{n,m} \boldsymbol{G}_{n,m} \boldsymbol{J}_{n,m}^T \tag{3.18}$$

with

$$\boldsymbol{G}_{n,m} = \begin{bmatrix} {\sigma_{xx}^C}^2(n) & {\sigma_{xy}^C}^2(n) & 0 & 0 & 0 & 0 & 0 & 0 \\ {\sigma_{yx}^C}^2(n) & {\sigma_{yy}^C}^2(n) & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & {\sigma_{xx}^C}^2(m) & {\sigma_{xy}^C}^2(m) & 0 & 0 & 0 & 0 \\ 0 & 0 & {\sigma_{yx}^C}^2(m) & {\sigma_{yy}^C}^2(m) & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & {\sigma_r}^2(n) & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & {\sigma_r}^2(m) & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & {\sigma_{dx}}^2(n,m) & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & {\sigma_{dy}}^2(n,m) \end{bmatrix}$$

and

$$\boldsymbol{J}_{n,m} = \begin{bmatrix} \frac{\partial x^A(m)}{\partial x^C(n)} & \frac{\partial x^A(m)}{\partial y^C(n)} & \frac{\partial x^A(m)}{\partial x^C(m)} & \frac{\partial x^A(m)}{\partial y^C(m)} & \frac{\partial x^A(m)}{\partial r(n)} & \frac{\partial x^A(m)}{\partial r(m)} & \frac{\partial x^A(m)}{\partial dx_{n,m}} & \frac{\partial x^A(m)}{\partial dy_{n,m}} \\ \frac{\partial y^A(m)}{\partial x^C(n)} & \frac{\partial y^A(m)}{\partial y^C(n)} & \frac{\partial y^A(m)}{\partial x^C(m)} & \frac{\partial y^A(m)}{\partial y^C(m)} & \frac{\partial y^A(m)}{\partial r(n)} & \frac{\partial y^A(m)}{\partial r(m)} & \frac{\partial y^A(m)}{\partial dx_{n,m}} & \frac{\partial y^A(m)}{\partial dy_{n,m}} \end{bmatrix}$$

All possible solutions for $\boldsymbol{x}_v^A(m)$ and their respective covariances $\boldsymbol{P}_v^A(m)$ are combined into a matrix $\boldsymbol{S}(m)$, where v is the index for all solutions at time $t(m)$.

$$\boldsymbol{S}(m) = \begin{bmatrix} x_1^A(m) & y_1^A(m) & {\sigma_{1_{xx}}^A}^2(m) & {\sigma_{1_{xy}}^A}^2(m) & {\sigma_{1_{yx}}^A}^2(m) & {\sigma_{1_{yy}}^A}^2(m) \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_v^A(m) & y_v^A(m) & {\sigma_{v_{xx}}^A}^2(m) & {\sigma_{v_{xy}}^A}^2(m) & {\sigma_{v_{yx}}^A}^2(m) & {\sigma_{v_{yy}}^A}^2(m) \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_q^A(m) & y_q^A(m) & {\sigma_{q_{xx}}^A}^2(m) & {\sigma_{q_{xy}}^A}^2(m) & {\sigma_{q_{yx}}^A}^2(m) & {\sigma_{q_{yy}}^A}^2(m) \end{bmatrix}, \; v = [1 \dots q]$$

We also define a position matrix $\boldsymbol{T}(m-q)$ which stores all possible past positions of the AUV, going back to $t(m-q)$, i.e. $\boldsymbol{x}_u^A(m-o)$; $\forall u = [1 \dots q], o = [1 \dots q]$, their respective covariances $\boldsymbol{P}_u^A(m-o)$ and an associated accumulated transition cost $c_u(m-o)$ at $t(m-o)$, where u indexes all possible positions, covariances and costs at $t(m-o)$.

$$
\begin{aligned}
\boldsymbol{T}(m-q) \;=\; & \left[\begin{array}{cccccc}
x_1^A(m-1) & y_1^A(m-1) & {\sigma_{1_{xx}}^A}^2(m-1) & \dots & {\sigma_{1_{yy}}^A}^2(m-1) & c_1(m-1) \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
x_u^A(m-1) & y_u^A(m-1) & {\sigma_{u_{xx}}^A}^2(m-1) & \dots & {\sigma_{u_{yy}}^A}^2(m-1) & c_u(m-1) \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
x_q^A(m-1) & y_q^A(m-1) & {\sigma_{q_{xx}}^A}^2(m-1) & \dots & {\sigma_{q_{yy}}^A}^2(m-1) & c_q(m-1) \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
x_1^A(m-o) & y_1^A(m-o) & {\sigma_{1_{xx}}^A}^2(m-o) & \dots & {\sigma_{1_{yy}}^A}^2(m-o) & c_1(m-o) \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
x_u^A(m-o) & y_u^A(m-o) & {\sigma_{u_{xx}}^A}^2(m-o) & \dots & {\sigma_{u_{yy}}^A}^2(m-o) & c_u(m-o) \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
x_q^A(m-o) & y_q^A(m-o) & {\sigma_{q_{xx}}^A}^2(m-o) & \dots & {\sigma_{q_{yy}}^A}^2(m-o) & c_q(m-o) \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
x_1^A(m-q) & y_1^A(m-q) & {\sigma_{1_{xx}}^A}^2(m-q) & \dots & {\sigma_{1_{yy}}^A}^2(m-q) & c_1(m-q) \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
x_u^A(m-q) & y_u^A(m-q) & {\sigma_{u_{xx}}^A}^2(m-q) & \dots & {\sigma_{u_{yy}}^A}^2(m-q) & c_u(m-q) \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
x_q^A(m-q) & y_q^A(m-q) & {\sigma_{q_{xx}}^A}^2(m-q) & \dots & {\sigma_{q_{yy}}^A}^2(m-q) & c_q(m-q)
\end{array}\right], \\
u \;=\; & [1 \dots q], \\
o \;=\; & [1 \dots q]
\end{aligned}
$$

If a known position $\boldsymbol{x}^A(0)$ (obtained on the surface through GPS) is available in the beginning it can be used to initialize $\boldsymbol{T}(0) = [\boldsymbol{x}^A(0) \;\; c(0) = 0]$. If no initial position is available, the first set of solutions $\boldsymbol{S}(0)$ initializes $\boldsymbol{T}(0)$ and position estimates become available when subsequent information packages are received.

Our cost function $\mathcal{C}_{u,v}(m-o, m)$ computes the cost (inverse of likelihood) of the AUV having traveled from $\boldsymbol{x}_u^A(m-o)$ to $\boldsymbol{x}_v^A(m)$ given $\boldsymbol{x}_u^A(m-o)$, $\boldsymbol{P}_u^A(m-o)$, $\boldsymbol{x}_v^A(m)$, $\boldsymbol{P}_v^A(m)$, $\boldsymbol{d}_{\overline{m-o,m}}$, $\boldsymbol{Q}_{\overline{m-o,m}}$.

This cost is expressed by the "distance" between $(\boldsymbol{x}_u^A(m-o) + \boldsymbol{d}_{\overline{m-o,m}})$, a solution at $t(m-o)$ forward propagated by the dead-reckoning information $\boldsymbol{d}_{\overline{m-o,m}})$, with the associated covariance $(\boldsymbol{P}_u^A(m-o) + \boldsymbol{Q}_{\overline{m-o,m}})$ and $\boldsymbol{x}_v^A$ a solution at $t(m)$ with the associated covariance $\boldsymbol{P}_v^A$. The distance metric used is the Kullback-Leibler divergence given by

$$\begin{aligned}\mathcal{C}_{u,v}(m-o,m) = \frac{1}{2}\Bigg(&\ln\Bigg(\frac{det(\boldsymbol{P}_v^A)}{det(\boldsymbol{P}_u^A(m-o)+\boldsymbol{Q}_{\overline{m-o,m}})}\Bigg)+\\ &+trace\big((\boldsymbol{P}_v^A)^{-1}(\boldsymbol{P}_u^A(m-o)+\boldsymbol{Q}_{\overline{m-o,m}})\big)+\\ &+\big(\boldsymbol{x}_v^A-(\boldsymbol{x}_u^A(m-o)+\boldsymbol{d}_{\overline{m-o,m}})\big)^T(\boldsymbol{P}_v^A)^{-1}\\ &\big(\boldsymbol{x}_v^A-(\boldsymbol{x}_u^A(m-o)+\boldsymbol{d}_{\overline{m-o,m}})\big)-2\Bigg)\end{aligned} \tag{3.19}$$

Using 3.19 we now compute the total cost $c_{u,v}(m-o,m)$ by computing the cost $\mathcal{C}_{u,v}(m-o,m)$ for all q^3 possible transitions from $\boldsymbol{T}(m-q)$ to $\boldsymbol{S}(m)$ and adding the new transition cost $\mathcal{C}_{u,v}(m-o,m)$ to the accumulated cost $c_u(m-o)$.

$$c_{u,v}(m-o,m) = \mathcal{C}_{u,v}(m-o,m)+c_u(m-o) \qquad \forall\ u=[1\dots q], v=[1\dots q], o=[1\dots q] \tag{3.20}$$

We then form a new position matrix $\boldsymbol{T}(m)$

$$\begin{aligned}\boldsymbol{T}(m) &= \begin{bmatrix} x_1^A(m) & y_1^A(m) & {\sigma_{1_{xx}}^A}^2(m) & {\sigma_{1_{xy}}^A}^2(m) & {\sigma_{1_{yx}}^A}^2(m) & {\sigma_{1_{yy}}^A}^2(m) & c_1(m)\\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots\\ x_v^A(m) & y_v^A(m) & {\sigma_{v_{xx}}^A}^2(m) & {\sigma_{v_{xy}}^A}^2(m) & {\sigma_{v_{yx}}^A}^2(m) & {\sigma_{v_{yy}}^A}^2(m) & c_v(m)\\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots\\ x_q^A(m) & y_q^A(m) & {\sigma_{q_{xx}}^A}^2(m) & {\sigma_{q_{xy}}^A}^2(m) & {\sigma_{q_{yx}}^A}^2(m) & {\sigma_{q_{yy}}^A}^2(m) & c_q(m)\end{bmatrix}\\ v &= [1\dots q]\end{aligned}$$

where $c_v(m)$ is the smallest accumulated cost associated with the transition to solution $\boldsymbol{x}_v^A(m)$ from of all q^2 possible positions $\boldsymbol{x}_u^A(m-o)$.

$$c_v(m) = \min_{\forall u}(c_{u,v}(m-o,m)) \qquad \forall\ v=[1\dots q], o=[1\dots q]. \tag{3.21}$$

All solutions $\boldsymbol{x}_v^A(m)$ are now hypotheses for possible positions of the AUV at $t(m)$ and weighted by the associated accumulated transition cost $c_v(m)$. The likeliest position $\boldsymbol{x}_w^A(m)$, i.e. our computed solution for $t(m)$, is the one with the smallest accumulated transition cost

$$\boldsymbol{x}_w^A(m) \text{ with } w \text{ s. t. } c_w(m) = \min_{\forall v}(c_v(m)) \tag{3.22}$$

3.5.4 Example

A single iteration of algorithm 5 is shown in the following example. Figure 3-8 shows a snapshot at $t(33)$ during a cooperative navigation experiment. The AUV (here

Initialize position matrix $\boldsymbol{T}(0) = [\boldsymbol{x}^A(0) \;\; c(0) = 0]$
loop {compute position}
  $m++$
  Wait for new range/position pair $\boldsymbol{x}^C(m)$,$z^C(m)$,$\boldsymbol{P}^C(m)$,$r(m)$ from CNA
  Use $z^C(m)$ to project $\boldsymbol{x}^C(m)$ to a plane at the AUV's depth $z^A(m)$
  for $j = 1$ to q **do** {Calculate intersection solution between now (m) and j steps in the past}
    $n = m - j$
    $\boldsymbol{x}_j^A(m) \leftarrow (3.17)|_{\boldsymbol{x}(n)^C, \boldsymbol{x}(m)^C, r(n), r(m), \boldsymbol{d}_{\overline{n,m}}}$ {Position}
    $\boldsymbol{P}_j^A(m) = \boldsymbol{J}_{n,m}\boldsymbol{G}_{n,m}\boldsymbol{J}_{n,m}^T$ {Covariance}
    $\boldsymbol{S}(m) \leftarrow \boldsymbol{x}_j^A(m), \boldsymbol{P}_j^A(m)$ {Add solution $\boldsymbol{x}_j^A(m)$ and its covariance $\boldsymbol{P}_j^A(m)$ to solution matrix:}
  end for
  for $o = 1$ to q **do** {Iterate through all past time steps}
    for $u = 1$ to q **do** {Iterate through all positions}
      for $v = 1$ to q **do** {Iterate through all solutions}
        $c_{u,v}(m-o, m) \leftarrow c_u(m-o) + (3.19)\Big|_{\boldsymbol{x}_u^A(m-o), \boldsymbol{P}_u^A(m-o), \boldsymbol{x}_v^A(m), \boldsymbol{P}_v^A(m), \boldsymbol{d}_{\overline{m-o,m}}, \boldsymbol{Q}_{\overline{m-o,m}}}$
      end for
      $\boldsymbol{T}(m) \overset{c_v(m)=\min_{\forall u}(c_{u,v}(m-o,m))}{\longleftarrow} [\boldsymbol{x}_v^A(m) \;\; \boldsymbol{P}_v^A(m) \;\; c_v(m)]$
    end for
  end for
  The computed position at $t(m)$ is : $\boldsymbol{x}_w^A(m) = \boldsymbol{x}_v^A(m)$ with w s. t. $c_w(m) = \min_{\forall v}(c_v(m))$
end loop

Algorithm 5: The Cooperative Navigation (CN) algorithm.

simulated by an ASC which also provides GPS for ground-truth) has just received a position/range-pair from the CNA (full circle). This circle intersects with the position/range-pair received at $t(32)$ (dashed circle) and forward propagated by the dead-reckoned distance $\boldsymbol{d}_{\overline{32,33}}$ to $\boldsymbol{x}^C(32')$. It also intersects with other position/range-pairs received at $t(k), (1 \leq k < 32)$ (positions of CNA not shown) forward propagated to $\boldsymbol{x}^C(k')$ by the corresponding dead-reckoned distance $\boldsymbol{d}_{\overline{k,33}}$. All intersections and therefore possible solutions at $t(33)$ are shown with their corresponding accumulated transition cost. The inset in figure 3-8 shows the detailed view near the ground-truth (GPS) position. The computed position at $t(33)$ (marked with a large "X") is the one with the smallest accumulated transition cost selected out of all possible positions $\boldsymbol{x}_v^A(33)$. In this case it is not the one closest to the GPS-derived position.

The complexity to compute a single position is $O(q^3)$ where q is the number of past measurements taken into account. The maximum frequency at which this computation step is invoked is limited by the duration of a data packet transmission. As the

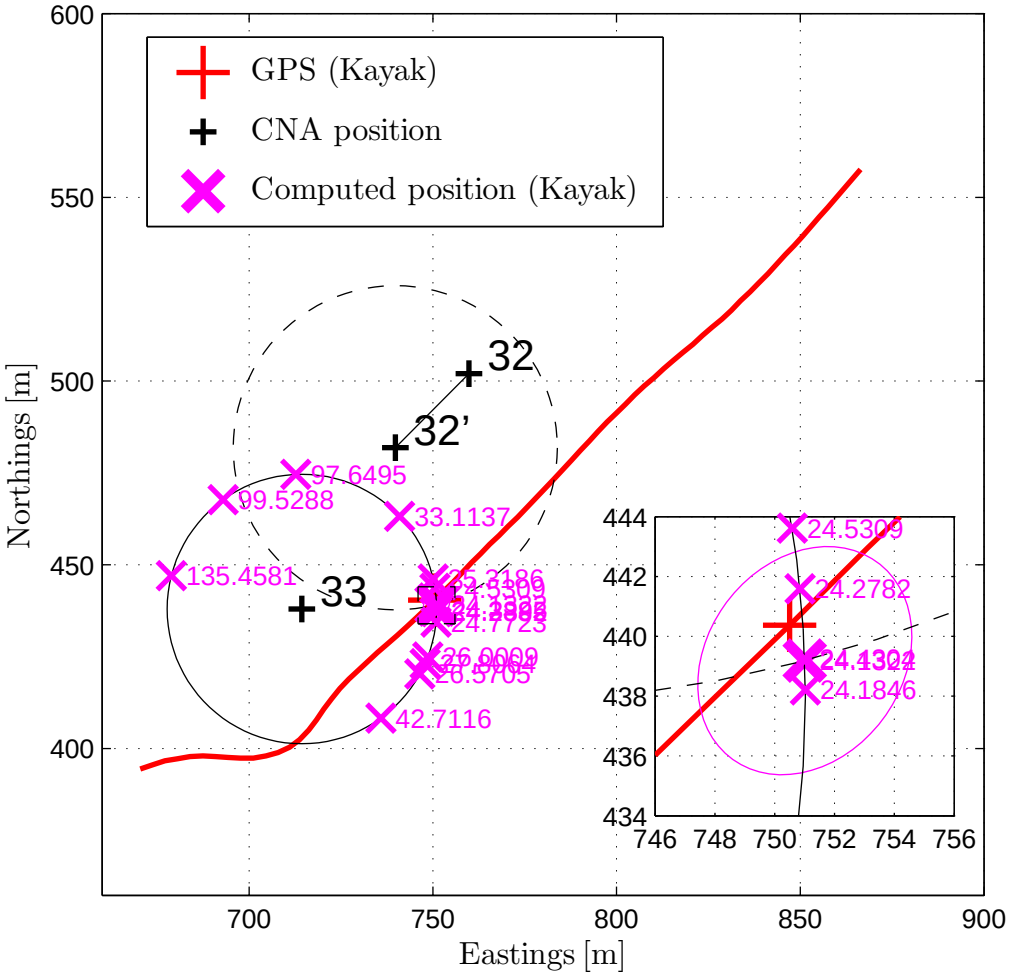


Figure 3-8: All possible solutions for solution #33 with accumulated transition cost; Inset: Detailed view of selected solution and GPS ground-truth.

transmission of a data packet takes 10 s the highest frequency at which algorithm 5 is called is $f_{max} = 0.1$ Hz. For $q \approx 10$ the time to compute a new position is t=0.1 s on a 1 GHz PC. This makes this algorithm well suited to run on the computer of today's AUVs.

3.6 Maintaining Consistency

When robot 1 uses the position estimate of robot 2 to update its own, their position estimates become dependent. This can be observed in the evolution of the covariance

matrix from $\overline{\boldsymbol{P}}_{1,2}(k+1)$ to $\boldsymbol{P}_{1,2}(k+1)$ during the EKF update (eq. (3.14)). In

$$\boldsymbol{P}_{1,2}(k+1) = \begin{bmatrix} \boldsymbol{P}_1(k+1) & \boldsymbol{P}_{12}(k+1) \\ \boldsymbol{P}_{21}(k+1) & \boldsymbol{P}_2(k+1) \end{bmatrix}$$

the elements $\boldsymbol{P}_{12}(k+1)$ and $\boldsymbol{P}_{21}(k+1)$, which were 0 in equation (3.12), indicating that the two positions were not dependent, become non-zero meaning that the two positions are now dependent.

If at a later time robot 2 uses the information from robot 1 to update its own position, $\overline{\boldsymbol{P}}_{2,1}(k+1)$ in (3.12) is not longer a diagonal matrix. If the off-diagonal elements are then assumed to be 0, $\overline{\boldsymbol{P}}_{2,1}(k+1)$, and, after the update $\boldsymbol{P}_{2,1}(k+1)$, will not properly represent the uncertainty of the position estimate $\boldsymbol{\mu}_2(k+1)$. The EKF will become overconfident in the estimate of $\boldsymbol{\mu}_2$ and can diverge [2]. In section 3.6.5 we show an example of EKF-based CN in which the correlations are not taken into account and lead to an overconfident position estimate. Properly modeling the correlations and determining the correct values for $\boldsymbol{P}_{12}(k+1)$ and $\boldsymbol{P}_{21}(k+1)$ requires robot 1 to have exact information about the evolution of $\boldsymbol{x}_2$ which is difficult to achieve in a de-centralized system.

Various approaches have been devised to either properly account for the correlation or to use very conservative uncertainty bounds to avoid overconfidence, but many of the algorithms impose additional requirements which make them unfeasible for many CN-scenarios.

3.6.1 Covariance Intersection

A general approach to the problem of fusing dependent estimates has been proposed by Julier and Uhlmann [49], [50]. Their Covariance Intersection (CI) algorithm fuses two different estimates for a random variable, each represented by their estimated mean and covariance much like the update step in the Kalman filter. The result is a posterior covariance that guarantees consistency under the assumption of Gaussian noise. Arambel et al. present an application of the CI algorithm for a group of space vehicles, in which relative position measurements are communicated in a ring topology [2]. Each of these works has examples of how the state estimator can diverge if estimates are fused with a simple Kalman update without accounting for correlation among the estimates.

A disadvantage of the CI algorithm is that it can only fuse two state estimates. Additionally, unlike the standard Kalman Filter, it cannot perform a partial update such as those that apply to vehicles that only have a range *or* bearing sensor. As a result, robots that only have a bearing sensor, such as a monocular camera, or have only range information from time-of-flight-based techniques cannot participate in a setup which relies on CI for the update of position estimates.

3.6.2 The IU Algorithm

We propose the IU algorithm which ensures that position estimates do not become dependent as outlined in the previous section. The IU algorithm requires that each robot includes additional information within its status broadcast. The receiving robots can then use this information to ensure that the correlations are properly accounted for. The approach does not require centralized data storage and processing as all updates are done locally on each vehicle using only data from the broadcasting vehicle. It does not enforce a particular communication hierarchy or topology and individual members can join and leave the group and do not need any awareness of previous communications or the size of the group. In contrast to other methods, broadcasts within the IU algorithm framework do not need to be received by all participating vehicles as each transmission contains all the information that is required for a position update which accounts for the correlations.

Concept

The basic concept of the algorithm is to maintain a filter for each subset of vehicles from which it received updates and ensure that only information from these vehicles is used to update that particular filter. During broadcast the transmitting vehicle sends the estimates from all filters. On a receiving vehicle the IU algorithm matches the results from the filters of the transmitting vehicles to its own bank of filters and updates them accordingly. It is important to note the that the IU algorithm does not do the update itself, but is an information arbiter. The algorithm is thus able to work with any type of Bayes filter. Figure 3-9 compares the two approaches for a set of three vehicles.

Initialization

For the IU algorithm, each vehicle i now maintains a set $\boldsymbol{\mathcal{X}}_i(k)$ of state estimate vectors together with a set $\boldsymbol{\mathcal{P}}_i(k)$ of associated covariance matrices. As we will explain later, the maximum size of the set is 2^n where n is the total number of vehicles cooperating for navigation.

$$\begin{aligned} \boldsymbol{\mathcal{X}}_i(k) &= \left\{ \boldsymbol{x}_i^1(k), \ldots, \boldsymbol{x}_i^q(k), \ldots, \boldsymbol{x}_i^{2^n}(k) \right\} \\ \boldsymbol{\mathcal{P}}_i(k) &= \left\{ \boldsymbol{P}_i^1(k), \ldots, \boldsymbol{P}_i^q(k), \ldots, \boldsymbol{P}_i^{2^n}(k) \right\} \end{aligned}$$

Before vehicle i receives information from any other vehicle the only contents of $\boldsymbol{\mathcal{X}}_i(k)$ and $\boldsymbol{\mathcal{P}}_i(k)$ are $\boldsymbol{x}_i^1(k)$ and $\boldsymbol{P}_i^1(k)$.

$$\begin{aligned} \boldsymbol{\mathcal{X}}_i(k) &= \left\{ \boldsymbol{x}_i^1(k) \right\} \\ \boldsymbol{\mathcal{P}}_i(k) &= \left\{ \boldsymbol{P}_i^1(k) \right\} \end{aligned}$$

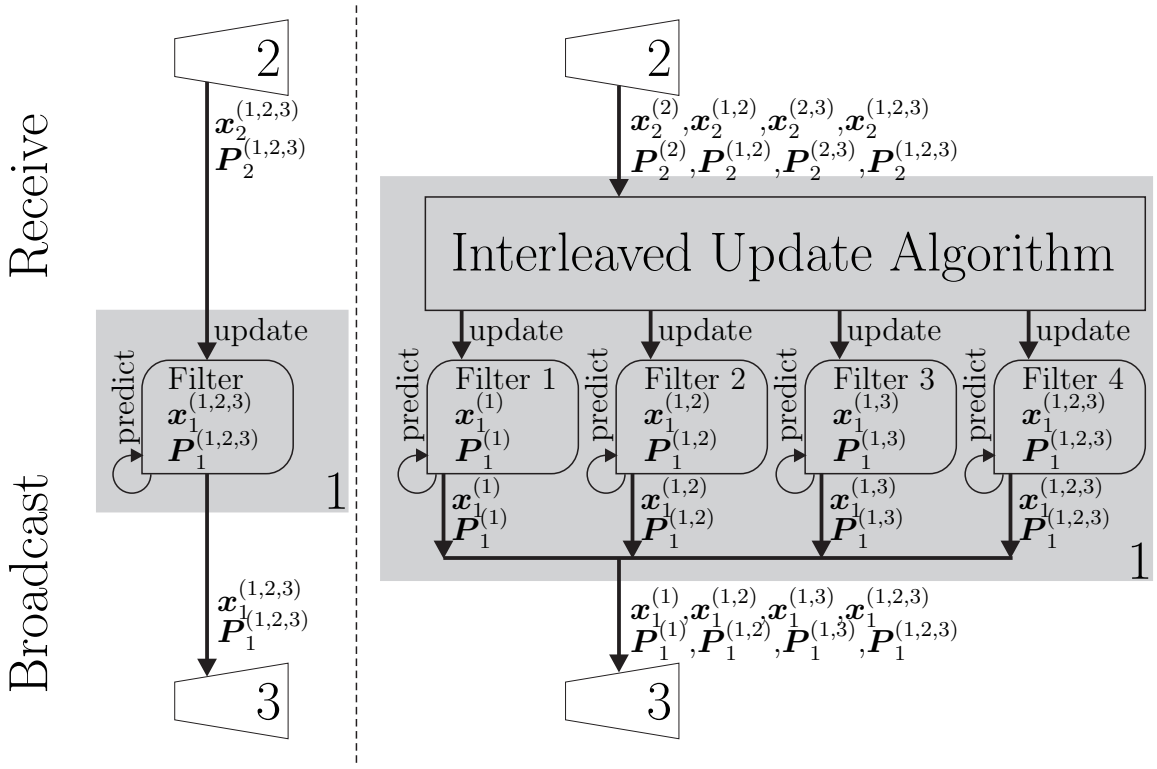


Figure 3-9: Comparison between the standard EKF (left) and the IU update (right). In both cases vehicle 1 (gray box) receives a broadcast from vehicle 2 and broadcasts information to vehicle 3. Left: The standard EKF maintains a single filter which is updated by all other vehicles and therefore dependent to all other vehicles. Right: The IU algorithm maintains a filter for each subset of vehicles such that there is always a filter which is not dependent to at least one vehicle and broadcasts all estimates.

Prediction

Each time vehicle i receives proprioceptive sensor readings it uses the Kalman Filter prediction steps for state and covariance (eq. (3.9) and (3.10)) to propagate $\boldsymbol{x}_i^1(k)$ and $\boldsymbol{P}_i^1(k)$.

$$\begin{aligned} \boldsymbol{x}_i^1(k) &\xrightarrow{(3.9)} \overline{\boldsymbol{x}}_i^1(k+1) \\ \boldsymbol{P}_i^1(k) &\xrightarrow{(3.10)} \overline{\boldsymbol{P}}_i^1(k+1) \end{aligned}$$

First Update

When vehicle i receives a broadcast from vehicle j at time l, it first instantiates a second filter by copying the state and covariance matrix.

$$\begin{aligned} \overline{\boldsymbol{x}}_i^2(l) &= \overline{\boldsymbol{x}}_i^1(l) \\ \overline{\boldsymbol{P}}_i^2(l) &= \overline{\boldsymbol{P}}_i^1(l) \end{aligned}$$

The vehicle also instantiates a matrix $\boldsymbol{T}_i$ in which each row represents a filter and each column represents a vehicle number. The entry in matrix $\boldsymbol{T}_i(q,i)$ is the time when vehicle i was last used to update filter q.

Using the Kalman update equations (eq. (3.13) and (3.14)), we now only update $\overline{\boldsymbol{x}}_i^2(l)$ and $\overline{\boldsymbol{P}}_i^2(l)$. After this update, our sets $\boldsymbol{\mathcal{X}}_i(l)$, $\boldsymbol{\mathcal{P}}_i(l)$ and the matrix $\boldsymbol{T}_i(l)$ are

$$\begin{aligned} \boldsymbol{\mathcal{X}}_i(l) &= \left\{ \boldsymbol{x}_i^1(l), \boldsymbol{x}_i^2(l) \right\} \\ \boldsymbol{\mathcal{P}}_i(l) &= \left\{ \boldsymbol{P}_i^1(l), \boldsymbol{P}_i^2(l) \right\} \\ \boldsymbol{T}_i(l) &= \begin{bmatrix} 0 & \dots & l & 0 & \dots & 0 & 0 & \dots & 0 \\ 0 & \dots & l & 0 & \dots & l & 0 & \dots & 0 \end{bmatrix} \\ & \qquad\qquad \begin{matrix} \uparrow & & \uparrow \\ i & & j \end{matrix} \end{aligned}$$

Subsequent Predictions

The first prediction for vehicle i after the update, propagates both filters using eq. (3.9) and (3.10) to $\overline{\boldsymbol{\mathcal{X}}}_i(l+1)$ and $\overline{\boldsymbol{\mathcal{P}}}_i(l+1)$ and all elements in column i in $\overline{\boldsymbol{T}}_i(l+1)$ are set to $l+1$.

$$\begin{aligned} \overline{\boldsymbol{T}}_i(l+1) &= \begin{bmatrix} 0 & \dots & l+1 & 0 & \dots & 0 & 0 & \dots & 0 \\ 0 & \dots & l+1 & 0 & \dots & l & 0 & \dots & 0 \end{bmatrix} \\ & \qquad\qquad \begin{matrix} \uparrow & & \uparrow \\ i & & j \end{matrix} \end{aligned}$$

Matrix $\overline{\boldsymbol{T}}_i$ therefore keeps track of which vehicles have been used to update a particular filter as well as the age of the updates. Predictions after $l+1$ up to the next update are propagated the same way, both filters are propagated and all elements in column i of $\overline{\boldsymbol{T}}_i$ are set to the actual time. All other columns remain unchanged.

Broadcast

Every time vehicle i sends out a broadcast, the transmitted data consists of $\overline{\boldsymbol{\mathcal{X}}}_i$, $\overline{\boldsymbol{\mathcal{P}}}_i$ and $\overline{\boldsymbol{T}}_i$. By maintaining a state $\overline{\boldsymbol{x}}_i^1$ on vehicle i which is continuously propagated and has not been updated with information from vehicle j, we make sure that a future broadcast from vehicle i received by vehicle j contains a state which is not dependent with vehicle j and can therefore be used by vehicle j for an update.

Subsequent Updates

The general update case in which vehicle i receives a broadcast from j after both vehicles have received broadcasts from various other vehicles and have incorporated those to update their navigation filters unfolds as follows.

We define $\mathcal{S}_i$ as the set of all m vehicle ids from which vehicle i received updates. $\mathcal{S}_i$ not only contains the ids of from which vehicle i has directly received broadcasts, but also those ids which have been propagated to it through other vehicles. The power set 2^{S_i} then contains all 2^m possible subsets of these ids. Each subset

$$\overline{\boldsymbol{\mathcal{A}}}_i^1, \ldots, \overline{\boldsymbol{\mathcal{A}}}_i^q, \ldots, \overline{\boldsymbol{\mathcal{A}}}_i^{2^m} \subseteq (2^{S_i} \cup i) \tag{3.23}$$

then corresponds to a filter maintained in $\overline{\boldsymbol{x}}_i^q$,$\overline{\boldsymbol{P}}_i^q$ which maintains a state that has been updated by the ids in the corresponding subset $\overline{\boldsymbol{\mathcal{A}}}_i^q$ and therefore has correlations with these vehicles. The information about which ids are in the individual subsets is maintained in line q of $\overline{\boldsymbol{T}}_i$ as each line in $\overline{\boldsymbol{T}}_i$ corresponds to a subset of $\overline{\boldsymbol{\mathcal{A}}}_i$.

Similarly there is a set $\mathcal{S}_j$ for all o ids from which vehicle j has received broadcasts.

$$\overline{\boldsymbol{\mathcal{A}}}_j^1, \ldots, \overline{\boldsymbol{\mathcal{A}}}_j^p, \ldots, \overline{\boldsymbol{\mathcal{A}}}_j^{2^o} \subseteq (2^{S_j} \cup j)$$

When vehicle i receives $\overline{\boldsymbol{\mathcal{X}}}_j$,$\overline{\boldsymbol{\mathcal{P}}}_j$ and $\overline{\boldsymbol{T}}_j$ from vehicle j it first adds entries in $\overline{\boldsymbol{\mathcal{X}}}_i$,$\overline{\boldsymbol{\mathcal{P}}}_i$ and $\overline{\boldsymbol{T}}_i$ for all elements of $\overline{\boldsymbol{\mathcal{A}}}_j$ which are not in $\overline{\boldsymbol{\mathcal{A}}}_i$. As a result vehicle i then maintains filters for a new set $\overline{\boldsymbol{\mathcal{A}}}_i$

$$\overline{\boldsymbol{\mathcal{A}}}_i \cup \overline{\boldsymbol{\mathcal{A}}}_j \rightarrow \overline{\boldsymbol{\mathcal{A}}}_i$$

Each filter $\overline{\boldsymbol{x}}_i^q$,$\overline{\boldsymbol{P}}_i^q$ represented by $\overline{\boldsymbol{\mathcal{A}}}_i^q$ is now updated without introducing any additional correlations. This means that $\boldsymbol{\mathcal{A}}_i^q = \overline{\boldsymbol{\mathcal{A}}}_i^q$. To update $\overline{\boldsymbol{x}}_i^q$,$\overline{\boldsymbol{P}}_i^q$ we now find all possible combinations of sets from $\overline{\boldsymbol{\mathcal{A}}}_i$ and $\overline{\boldsymbol{\mathcal{A}}}_j$ such that

$$\overline{\boldsymbol{\mathcal{A}}}_i^g \cup \overline{\boldsymbol{\mathcal{A}}}_j^h \rightarrow \boldsymbol{\mathcal{A}}_i^q \tag{3.24}$$

Each of these combinations represents a possible update for $\overline{\boldsymbol{x}}_i^q$,$\overline{\boldsymbol{P}}_i^q$

$$\overline{\boldsymbol{x}}_i^g \overset{(3.13)\ with\ \overline{\boldsymbol{x}}_i^h}{\longrightarrow} \boldsymbol{x}_i^q \tag{3.25}$$

$$\overline{\boldsymbol{P}}_i^g \overset{(3.14)\ with\ \overline{\boldsymbol{P}}_i^h}{\longrightarrow} \boldsymbol{P}_i^q \tag{3.26}$$

We now select g and h such that $\boldsymbol{P}_i^q$ has the smallest trace of all possible combinations.

$$(g^*, h^*) = \underset{g,h \text{ s.t. } (3.24)}{\operatorname{argmin}} \Big(trace\big(\boldsymbol{P}_i^q\big) \Big) \tag{3.27}$$

Using g^* and h^* determined through eq. (3.27), we use eq. (3.25) to update the state.

$$\overline{\boldsymbol{x}}_i^{g*} \overset{(3.13)\ with\ \overline{\boldsymbol{x}}_i^{h*}}{\longrightarrow} \boldsymbol{x}_i^q$$

Line q in $\boldsymbol{T}_i$ is updated to reflect the age of updates.

$$\begin{aligned} \boldsymbol{T}_i^q(i,u) &= \overline{\boldsymbol{T}}_i^{g*}(g*,u) \qquad \forall u \in \overline{\boldsymbol{\mathcal{A}}}_i^{g*} \\ \boldsymbol{T}_i^q(i,u) &= \overline{\boldsymbol{T}}_j^{h*}(h*,u) \qquad \forall u \in \overline{\boldsymbol{\mathcal{A}}}_j^{h*} \end{aligned}$$

All steps in section (3.6.2) are repeated for all 2^n filters on vehicle i, and all other vehicles that overheard the broadcast, update their local filters accordingly.

3.6.3 Enforcing Constant Set Size

The amount of information which needs to be transmitted during each broadcast, as well as the number of local prediction and update steps, grows with $\mathcal{O}(m^2)$ where m is the size of set $\mathcal{S}_i$ as defined in section 3.6.2. The amount of data which needs to be transmitted per filter however is fairly small if the state is parameterized with a mean and a covariance ($\approx$ 10 bytes) and the update of each filter only requires 4 $[2 \times 6] \cdot [6 \times 6]$ matrix multiplications for a 3D environment where range and heading measurements are available. Assuming a data packet size of 10 kBytes, set sizes up to 30 ids are feasible.

For a large group of cooperating robots with the same level of uncertainty in their proprioceptive measurements, Roumeliotis et al. show in [74] that the uncertainty growth is inversely proportional to the number of robots. Thus the contribution of each additional robot follows a law of diminishing return. This suggests that set sizes of 30 and less are sufficient to obtain an improvement of navigation accuracy which is close to the theoretical maximum obtained when broadcasts of all available vehicles are incorporated.

Based on our available communications bandwidth and available processing cycles we can choose an upper bound b for the size of $\mathcal{S}_i$. If our set size grows larger than b we can incorporate the new broadcast according to section 3.6.2 and then resize $\mathcal{S}_i$ by eliminating the id which contributes the least amount of information. The resize process then consists of two steps. First we determine the vehicle (id) which contributes the least amount of information. Second we remove this id from $\mathcal{S}_i$ and modify $\overline{\boldsymbol{\mathcal{X}}}_i$,$\overline{\boldsymbol{\mathcal{P}}}_i$ and $\overline{\boldsymbol{T}}_i$ accordingly.

Compare

One method to determine the vehicle with id q which contributes the least amount of information is to compare the trace difference between the filter that was only updated by $\{q, i\}$ with the filter that has the dead reckoning result only $\{i\}$.

$$\begin{aligned} q^* &= \underset{q}{\operatorname{argmin}} \Big(trace\big(\boldsymbol{P}_i^g\big) - trace\big(\boldsymbol{P}_i^h\big) \Big) \\ & \quad \forall q \in \mathcal{S}_i, q \neq i \\ & \quad \boldsymbol{P}_i^g \text{ s.t. } \boldsymbol{\mathcal{A}}_g = \{i, q\} \\ & \quad \boldsymbol{P}_i^h \text{ s.t. } \boldsymbol{\mathcal{A}}_h = \{i\} \end{aligned}$$

Table 3.1: Contents of $\boldsymbol{\mathcal{X}}$,$\boldsymbol{\mathcal{P}}$ and $\boldsymbol{T}$ at time $k = 1$

$\overline{\boldsymbol{T}}_1(1)$	$\overline{\boldsymbol{\mathcal{X}}}_1(1)$	$\overline{\boldsymbol{\mathcal{P}}}_1(1)$	$\overline{\boldsymbol{T}}_2(1)$	$\overline{\boldsymbol{\mathcal{X}}}_2(1)$	$\overline{\boldsymbol{\mathcal{P}}}_2(1)$	$\overline{\boldsymbol{T}}_3(1)$	$\overline{\boldsymbol{\mathcal{X}}}_3(1)$	$\overline{\boldsymbol{\mathcal{P}}}_3(1)$	$\overline{\boldsymbol{T}}_4(1)$	$\overline{\boldsymbol{\mathcal{X}}}_4(1)$	$\overline{\boldsymbol{\mathcal{P}}}_4(1)$
1	$\overline{\boldsymbol{x}}_1^1(1)$	$\overline{\boldsymbol{P}}_1^1(1)$	1	$\overline{\boldsymbol{x}}_2^1(1)$	$\overline{\boldsymbol{P}}_2^1(1)$	1	$\overline{\boldsymbol{x}}_3^1(1)$	$\overline{\boldsymbol{P}}_3^1(1)$	1	$\overline{\boldsymbol{x}}_4^1(1)$	$\overline{\boldsymbol{P}}_4^1(1)$

Eliminate

After we determine q^* we remove all filters depending on q^* from our sets $\boldsymbol{\mathcal{X}}_i$,$\boldsymbol{\mathcal{P}}_i$ and obtain our new sets $\boldsymbol{\mathcal{X}}_i^-$ and $\boldsymbol{\mathcal{P}}_i^-$ and our updated matrix $\boldsymbol{T}_i^-$ by removing all lines which have a non-zero entry in column q^*.

$$\boldsymbol{\mathcal{X}}_i \xrightarrow{\boldsymbol{x}_i^h \text{ if } q^* \notin \boldsymbol{\mathcal{A}}_h} \boldsymbol{\mathcal{X}}_i^- \tag{3.28}$$

$$\boldsymbol{\mathcal{P}}_i \xrightarrow{\boldsymbol{P}_i^h \text{ if } q^* \notin \boldsymbol{\mathcal{A}}_h} \boldsymbol{\mathcal{P}}_i^- \tag{3.29}$$

$$\boldsymbol{T}_i \xrightarrow{\boldsymbol{T}_i(g,h)\ \forall g, \text{ with } \boldsymbol{T}_i(g,q^*)=0} \boldsymbol{T}_i^- \tag{3.30}$$

3.6.4 Example

The four frames in figure 3-10 and the tables I through IV show how the sets $\overline{\boldsymbol{\mathcal{X}}}_i$,$\overline{\boldsymbol{\mathcal{P}}}_i$ and the matrix $\overline{\boldsymbol{T}}_i$ evolve over time.

k=1

Up to this point all four vehicles have only used dead-reckoning information so none of their positions are dependent. All sets $\overline{\boldsymbol{\mathcal{X}}}_i$,$\overline{\boldsymbol{\mathcal{P}}}_i$ only contain a single state and covariance matrix.

k=2

Vehicle 1 broadcasts its state $\boldsymbol{x}_1(2)$ which is received by vehicle 2 and 3. Both vehicles instantiate a second filter $\boldsymbol{x}_2^2(2)$,$\boldsymbol{P}_2^2(2)$ and $\boldsymbol{x}_2^3(2)$,$\boldsymbol{P}_3^2(2)$ respectively which are updated with the broadcast and range received from vehicle 1, while the other filter in both vehicles are not.

k=3

Up to k=3 all filters (filter 1 in vehicle 1 and 4, filter 2 in vehicle 2 and 3) are propagated using the Kalman time prediction step. At k=3 the broadcast from 2 is received at 4. As 2 has been previously updated with 1 the set of filters received by 4 contains 2 new ids (1 and 2). Vehicle 4 therefore instantiates 3 additional filters, each containing a possible permutation of $\mathcal{S}_4$ as specified in eq. (3.23).

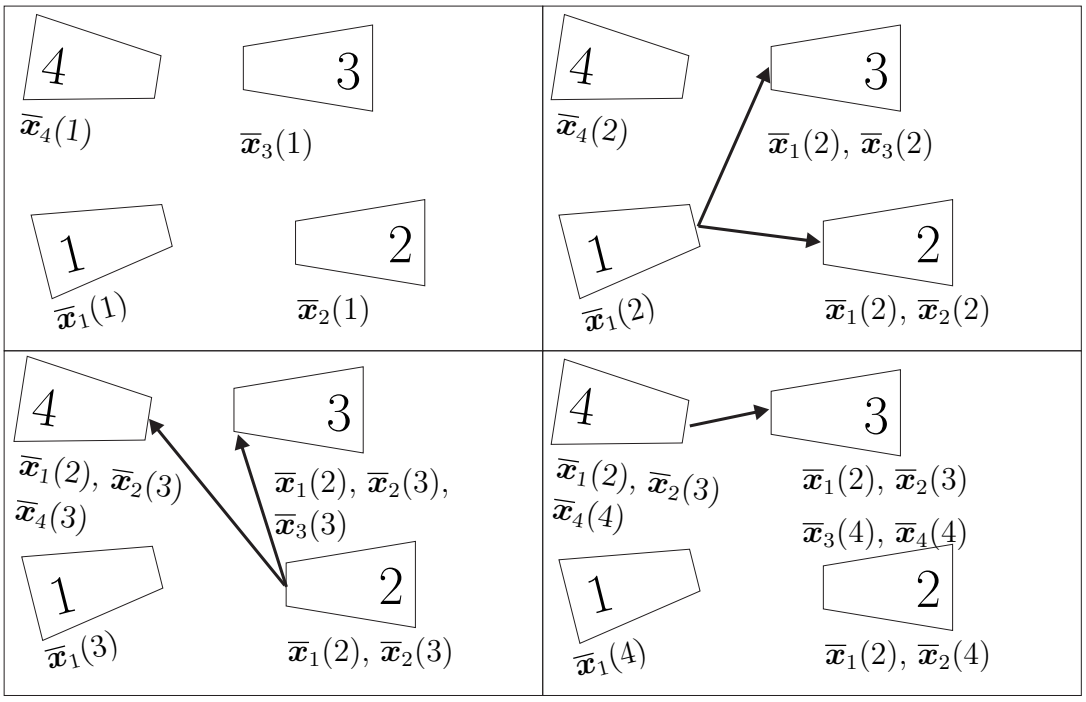


Figure 3-10: Four vehicles exchanging navigation information for Cooperative Navigation from time k=1 (top left) to k=4 (bottom right). The arrows indicate which vehicle broadcasts during a particular time step and which vehicles received the broadcast. Below each vehicle are the states which were used to update this vehicle's various position filters.

Table 3.2: Contents of $\boldsymbol{\mathcal{X}}$,$\boldsymbol{\mathcal{P}}$ and $\boldsymbol{T}$ at time $k = 2$

$\overline{\boldsymbol{T}}_1(2)$	$\overline{\boldsymbol{\mathcal{X}}}_1(2)$	$\overline{\boldsymbol{\mathcal{P}}}_1(2)$
2	$\overline{\boldsymbol{x}}_1^1(2)$	$\overline{\boldsymbol{P}}_1^1(2)$

$\overline{\boldsymbol{T}}_2(2)$	$\overline{\boldsymbol{\mathcal{X}}}_2(2)$	$\overline{\boldsymbol{\mathcal{P}}}_2(2)$
2	$\overline{\boldsymbol{x}}_2^1(2)$	$\overline{\boldsymbol{P}}_2^1(2)$
2 2	$\overline{\boldsymbol{x}}_2^2(2)$	$\overline{\boldsymbol{P}}_2^2(2)$

$\overline{\boldsymbol{T}}_3(2)$	$\overline{\boldsymbol{\mathcal{X}}}_3(2)$	$\overline{\boldsymbol{\mathcal{P}}}_3(2)$
2	$\overline{\boldsymbol{x}}_3^1(2)$	$\overline{\boldsymbol{P}}_3^1(2)$
2 2	$\overline{\boldsymbol{x}}_3^2(2)$	$\overline{\boldsymbol{P}}_3^2(2)$

$\overline{\boldsymbol{T}}_4(2)$	$\overline{\boldsymbol{\mathcal{X}}}_4(2)$	$\overline{\boldsymbol{\mathcal{P}}}_4(2)$
2	$\overline{\boldsymbol{x}}_4^1(2)$	$\overline{\boldsymbol{P}}_4^1(2)$

k=4

At k=4 vehicle 3 receives a broadcast from vehicle 4. After the update vehicle 3 now maintains the maximum set of 8 filters.

3.6.5 Simulation Results

To evaluate the algorithm we set up a simulation with three vehicles. All three vehicles had a very accurate initial position estimate. After the start of the mission they continuously estimated their positions using forward (u) and starboard (v) speed as well as heading measurements (θ) from very noisy sensors. The sensor noise was assumed Gaussian and the standard deviations are show in table 3.5. Using these

Table 3.3: Contents of $\boldsymbol{\mathcal{X}}$,$\boldsymbol{\mathcal{P}}$ and $\boldsymbol{T}$ at time $k = 3$

$\overline{\boldsymbol{T}}_1(3)$	$\overline{\boldsymbol{\mathcal{X}}}_1(3)$	$\overline{\boldsymbol{\mathcal{P}}}_1(3)$
3	$\overline{\boldsymbol{x}}_1^1(3)$	$\overline{\boldsymbol{P}}_1^1(3)$

$\overline{\boldsymbol{T}}_2(3)$	$\overline{\boldsymbol{\mathcal{X}}}_2(3)$	$\overline{\boldsymbol{\mathcal{P}}}_2(3)$
3	$\overline{\boldsymbol{x}}_2^1(3)$	$\overline{\boldsymbol{P}}_2^1(3)$
2 3	$\overline{\boldsymbol{x}}_2^2(3)$	$\overline{\boldsymbol{P}}_2^2(3)$

$\overline{\boldsymbol{T}}_3(3)$	$\overline{\boldsymbol{\mathcal{X}}}_3(3)$	$\overline{\boldsymbol{\mathcal{P}}}_3(3)$
3	$\overline{\boldsymbol{x}}_3^1(3)$	$\overline{\boldsymbol{P}}_3^1(3)$
2 3	$\overline{\boldsymbol{x}}_3^2(3)$	$\overline{\boldsymbol{P}}_3^2(3)$
3 3	$\overline{\boldsymbol{x}}_3^3(3)$	$\overline{\boldsymbol{P}}_3^3(3)$
2 3 3	$\overline{\boldsymbol{x}}_3^4(3)$	$\overline{\boldsymbol{P}}_3^4(3)$

$\overline{\boldsymbol{T}}_4(3)$	$\overline{\boldsymbol{\mathcal{X}}}_4(3)$	$\overline{\boldsymbol{\mathcal{P}}}_4(3)$
3	$\overline{\boldsymbol{x}}_4^1(3)$	$\overline{\boldsymbol{P}}_4^1(3)$
2 3	$\overline{\boldsymbol{x}}_4^2(3)$	$\overline{\boldsymbol{P}}_4^2(3)$
3 3	$\overline{\boldsymbol{x}}_4^3(3)$	$\overline{\boldsymbol{P}}_4^3(3)$
2 3 3	$\overline{\boldsymbol{x}}_4^4(3)$	$\overline{\boldsymbol{P}}_4^4(3)$

Table 3.4: Contents of $\boldsymbol{\mathcal{X}}$,$\boldsymbol{\mathcal{P}}$ and $\boldsymbol{T}$ at time $k = 4$

$\overline{\boldsymbol{T}}_1(3)$	$\overline{\boldsymbol{\mathcal{X}}}_1(4)$	$\overline{\boldsymbol{\mathcal{P}}}_1(4)$
4	$\overline{\boldsymbol{x}}_1^1(4)$	$\overline{\boldsymbol{P}}_1^1(4)$

$\overline{\boldsymbol{T}}_2(4)$	$\overline{\boldsymbol{\mathcal{X}}}_2(4)$	$\overline{\boldsymbol{\mathcal{P}}}_2(4)$
4	$\overline{\boldsymbol{x}}_2^1(4)$	$\overline{\boldsymbol{P}}_2^1(4)$
2 4	$\overline{\boldsymbol{x}}_2^2(4)$	$\overline{\boldsymbol{P}}_2^2(4)$

$\overline{\boldsymbol{T}}_3(4)$	$\overline{\boldsymbol{\mathcal{X}}}_3(4)$	$\overline{\boldsymbol{\mathcal{P}}}_3(4)$
4	$\overline{\boldsymbol{x}}_3^1(4)$	$\overline{\boldsymbol{P}}_3^1(4)$
2 4	$\overline{\boldsymbol{x}}_3^2(4)$	$\overline{\boldsymbol{P}}_3^2(4)$
3 4	$\overline{\boldsymbol{x}}_3^3(4)$	$\overline{\boldsymbol{P}}_3^3(4)$
2 3 4	$\overline{\boldsymbol{x}}_3^4(4)$	$\overline{\boldsymbol{P}}_3^4(4)$
4 4	$\overline{\boldsymbol{x}}_3^5(4)$	$\overline{\boldsymbol{P}}_3^5(4)$
2 4 4	$\overline{\boldsymbol{x}}_3^6(4)$	$\overline{\boldsymbol{P}}_3^6(4)$
3 4 4	$\overline{\boldsymbol{x}}_3^7(4)$	$\overline{\boldsymbol{P}}_3^7(4)$
2 3 4 4	$\overline{\boldsymbol{x}}_3^8(4)$	$\overline{\boldsymbol{P}}_3^8(4)$

$\overline{\boldsymbol{T}}_4(4)$	$\overline{\boldsymbol{\mathcal{X}}}_4(4)$	$\overline{\boldsymbol{\mathcal{P}}}_4(4)$
4	$\overline{\boldsymbol{x}}_4^1(4)$	$\overline{\boldsymbol{P}}_4^1(4)$
2 4	$\overline{\boldsymbol{x}}_4^2(4)$	$\overline{\boldsymbol{P}}_4^2(4)$
3 4	$\overline{\boldsymbol{x}}_4^3(4)$	$\overline{\boldsymbol{P}}_4^3(4)$
2 3 4	$\overline{\boldsymbol{x}}_4^4(4)$	$\overline{\boldsymbol{P}}_4^4(4)$

sensors each vehicle dead-reckoned its position using the EKF prediction equations for state and covariance (3.9),(3.9).

Table 3.5: Variance of sensor noise for the simulated vehicles.

Vehicle	σ_u,σ_v	σ_θ	σ_r
1	0.3 m/s	$10\,^\circ$	$0.1\,m$
2	0.3 m/s	$10\,^\circ$	$0.1\,m$
3	0.2 m/s	$2\,^\circ$	$0.1\,m$

Figure 3-11 shows two typical runs. Past true positions are indicated by small "×" and the actual true position at t=2000 s with a large "×". The vehicle's position estimate is indicated by a large "+" together with the 3σ-covariance ellipse. The enlarged sections in the insets show clearly how for all three vehicles the true position is outside 3σ-bound, i.e. the 99.6 % confidence interval when using the standard EKF-update (figure 3-11 left), but well within it when using the EKF with the IU-algorithm for the update (figure 3-11 right).

Figure 3-12 shows how the the error $\tilde{x}$ in the position estimate for the x component develops over time. As no outside position fix is available to any of the three vehicles

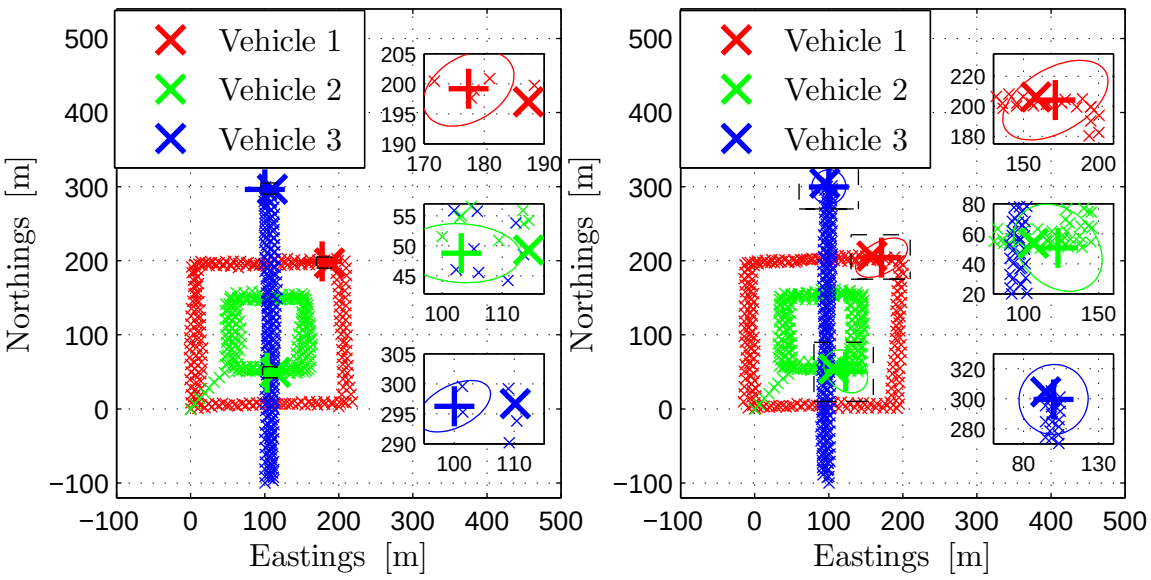


Figure 3-11: True (×) and estimated (+) position with 3σ-covariance ellipse of three vehicles navigating cooperatively using an EKF and naively incorporating all updates (left) or selectively updating using the IU algorithm (right). Note the change of scale between of the enlarged sections on the left and on the right plot.

the error grows. The selective updating used in the IU-algorithm incorporates fewer measurements than the standard EKF and as a result the error grows faster, but the error accumulated through the IU-algorithm is properly accounted for and the error $\tilde{x}$ remains well within the $3\sigma_{xx}$-bound (figure 3-12 right) while the standard EKF algorithm is overconfident. This overconfidence can cause the standard EKF-filter to diverge.

To asses the consistency of the position estimate of standard a EKF vs. that of the IU algorithm we computed the Normalized Estimation Error Squared (NEES) as described in [10] for 20 runs (ten standard EKF and ten IU).

$$\epsilon(k) = \tilde{x}(k|k)^T \, P(k|k)^{-1} \, \tilde{x}(k|k)$$

For each time k we compute the $N = 10$ average NEES $\bar{\epsilon}(k)$.

$$\bar{\epsilon}(k) = \frac{1}{N} \sum_{i=1}^{N} \epsilon^i(k) \tag{3.31}$$

Under the hypothesis H_0 that the filter is consistent and under the linear-Gaussian assumption $N\bar{\epsilon}(k)$ will have a chi-square density with $N\, n_x$ degrees of freedom, where n_x is the dimension of $\boldsymbol{x}$. The hypothesis H_0, that the state estimation errors are consistent with the filter-calculated covariances, also called *chi-square test*, is accepted if

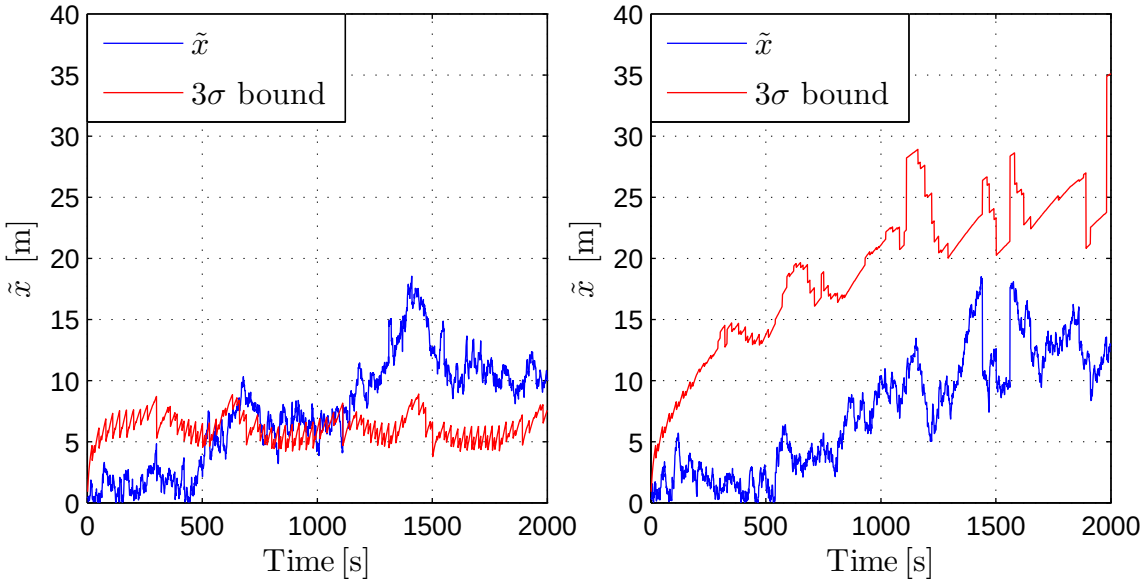


Figure 3-12: Error in x-position (blue) and 3σ bound (red) of vehicle 1 navigating cooperatively using an EKF and naively incorporating all updates (left) or selectively updating using the IU algorithm (right).

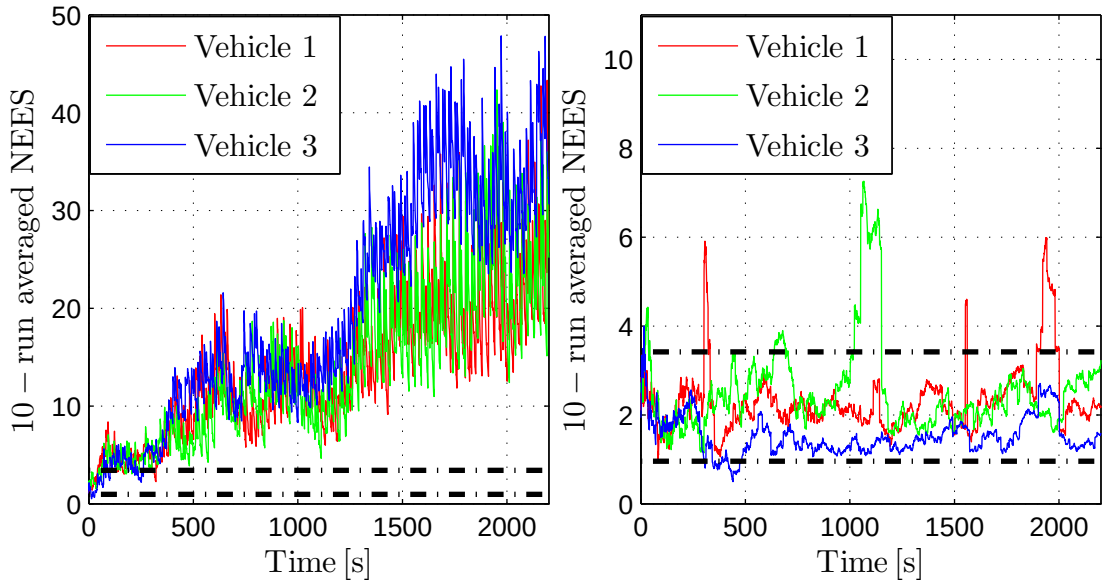


Figure 3-13: Averaged NEES for 10 runs as shown in fig 3-11 using the standard EKF (left) and the EKF with the IU algorithm (right).

$$\bar{\epsilon}(k) \in [r_1, r_2]$$

where the acceptance interval is determined such that

$$P\left\{\bar{\epsilon}(k) \in [r_1, r_2] \,|H_0\right\} = 1 - \alpha$$

The two sided 95 % region for a 20 degree of freedom ($Nn_x = 10 \cdot 2 = 2 = 20$) chi-square distribution is

$$\left[\chi^2_{20}(0.025), \chi^2_{20}(0.975)\right] = [9.60, 34.2] \tag{3.32}$$

Dividing the interval in (3.32) by N we obtain

$$\left[\frac{\chi^2_{20}(0.025)}{N}, \frac{\chi^2_{20}(0.975)}{N}\right] = [0.96, 3.42]\,. \tag{3.33}$$

Figure 3-13 shows the 10-run average NEES according to (3.31) and the boundaries determined in (3.33). For the standard EKF update the NEES quickly grows above the upper bound (figure 3-13 left) and indicates that this approach not only leads to inconsistent results, but that this inconsistency is growing. For the IU algorithm between 5 % and 9 % of the values fall outside the 95 % region which is acceptable [10].

3.6.6 Conclusion

Simulations such as the one in section 3.6.5 show that not properly accounting for the correlations between vehicles can quickly cause the navigation filter to diverge. Our proposed IU algorithm ensures that a filter is only updated with a measurement that it is not dependent to, thus ensuring that the position estimate of all individual filters will not suffer from overconfidence. We also propose a method that maintains an upper bound on the additional cost in computation and communication bandwidth.

Chapter 4

Intra-Vehicle Geometries for Cooperating AUVs

When a heterogeneous group of vehicles exchanges navigation information two scenarios are possible. In the first case, every vehicle broadcasts information and every other vehicle receiving the message incorporates it to improve its own position estimate. Each vehicle can at any point transmit navigation information or receive and incorporate it. We refer to this scenario as "organic cooperation" and it is outlined in section 4.1.1. The second one, described in section 4.1.2 is a more hierarchical approach in which a dedicated set of Communication and Navigation Aid (CNA) vehicles exists. These vehicles maintain a very accurate position estimate throughout the entire mission and broadcast it so that all other vehicles may use this information to improve their position estimate.

In both scenarios, the achievable improvement of the position estimate strongly depends on the relative position of the participating AUVs with respect to each other. However only in the hierarchical scenario in which dedicated beacons in the form of CNAs exist, this geometry can be controlled by positioning the CNAs. This is illustrated in a simulation in section 4.2.1. Before introducing an algorithm for dynamic positioning of CNAs to minimize the navigation uncertainty of all receiving vehicles, in section 4.3.6, we will first discuss previous work related to error metrics in 4.3.2. Various error metrics which can be used to parametrize the position uncertainty are shown in section 4.3.3. To address the problem of dynamically planning optimal beacon positioning we briefly investigate the case of simultaneous trilateration from static beacons in section 4.3.4. Section 4.3.5 explains the assumptions made by our algorithm in controlling the CNAs and 4.3.7 presents the results for two simulated runs.

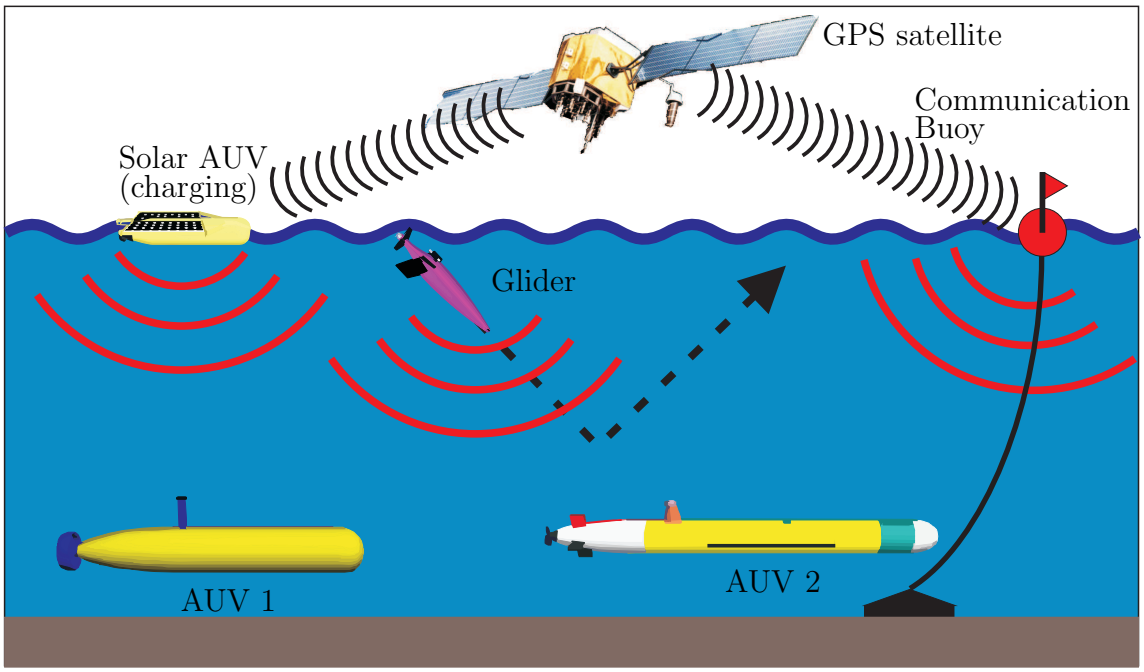


Figure 4-1: Organic CN scenario: AUV 1 and AUV 2 use "beacons of opportunity" such as a recently submerged glider which still has a very good dead-reckoned estimate or a recharging solar AUV or a communications buoy which both have access to GPS.

4.1 Vehicle Task Hierarchies for CN

4.1.1 Organic Cooperation

Figure 4-1 shows a possible scenario of a heterogeneous group of vehicles in which all vehicles can at one point in time participate actively (by transmitting) or passively (by receiving) in cooperative navigation. For example a solar powered AUV, such as the one described by Blidberg [16] and Crimmins [22], can serve as a navigation beacon while recharging its batteries on the surface while also having access to GPS. A glider (see section 1.1.1) as described by Eriksen in [30] penetrates the surface at the end of each dive to obtain a GPS fix, as well as to communicate over satellite. During its time at the surface with access to GPS, the glider can serve as a navigation beacons= for submerged vehicles in the area.

4.1.2 Dedicated Navigation Beacon Vehicles

The concept of dedicated CNA was first proposed in [88] for a mine-hunting scenario (shown in figure 4-2) with the underlying idea that a very small number of CNAs (one or two) with a very accurate estimate of their positions could be used to provide a much larger group of Search, Classify and Map (SCM)-vehicles with navigation information. These SCM-vehicles would be equipped with a special sonar payload to

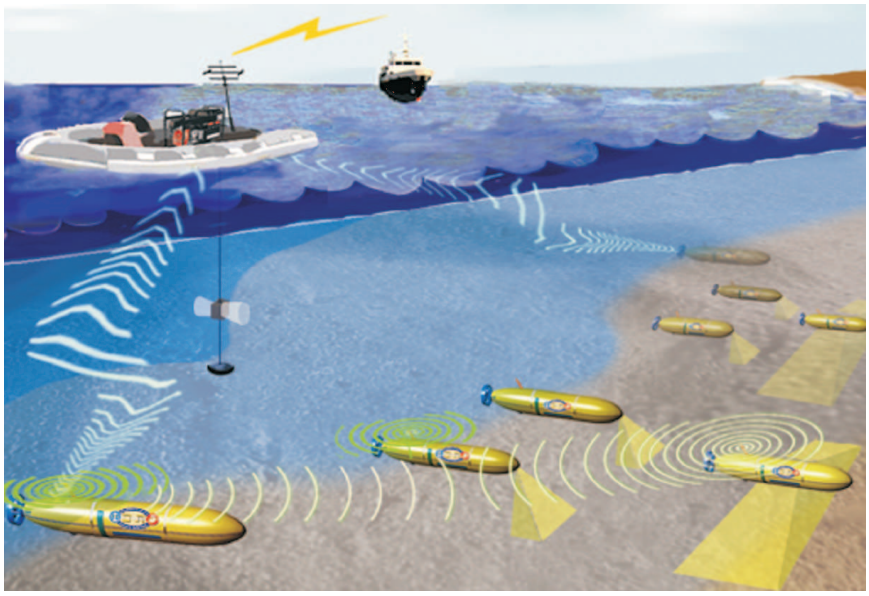


Figure 4-2: Hierarchical cooperation scenario where a small number of CNAs maintains a very accurate position estimate while a much larger group of SCM-vehicles with specialized sonar equipment searches the sea floor for mines. Graphic courtesy of Bluefin Robotics.

detect buried or free-floating mines. The CNAs would be either surface crafts with a permanent access to GPS or AUVs with a very accurate (and expensive) navigation suite. To maintain a bounded uncertainty on their position estimates, these CNAs would move at a very shallow depth and surface for a GPS fix whenever necessary. The SCMs outfitted with much simpler (and cheaper) navigation sensors would be able to maintain a bounded uncertainty on their position estimates without surfacing over the entire duration of the mission.

The sole mission objective of the CNAs is to minimize the overall uncertainty of the SCM vehicles. To accomplish this, their first objective is to maintain a very good estimate about their own position, as in Cooperative Navigation any uncertainty in the CNA's position directly translates into an uncertainty in the SCM's position. In addition, the relative position between the CNA and SCM will also strongly affect the position uncertainty of the SCM as we will show in the following section. Therefore the second objective of the CNA is to adjust its position such that the CNA-SCM geometry is optimal for CN.

4.2 Beacon Geometries for Cooperating AUVs

The hierarchical scenario shown in figure 4-2 allows us to position CNAs as dedicated navigation beacons for our AUV. In this section we examine the effect of the relative position between the CNAs and a single AUV. We ran two simulations in which the relative position between the CNAs and the AUV was the only parameter we changed.

Table 4.1: Variance of sensor noise for the simulated vehicles for geometry comparison (figure 4-3)

Vehicle	σ_u,σ_v	σ_θ	σ_r	Notes
CNA 1	0 m/s	0 °	1 m	CNAs have GPS
CNA 2	0 m/s	0 °	1 m	
AUV	0.2 m/s	10 °	1 m	

4.2.1 Simulation

To illustrate the effect of different geometries between the CNAs and the SCMs on the position accuracy obtainable through CN we simulated two CN missions in this section. The only difference between the missions was the relative position between the CNAs and the SCM[1]. For both simulations we assumed the CNAs to be surface crafts with access to GPS. We simulate the CNAs' access to an absolute position estimate by setting the variance of the measured forward σ_u and starboard speed σ_v as well as the heading variance σ_θ to zero. As a result the CNAs have a constant position uncertainty which only depends on the quality of the GPS signal. The AUV's sensor variances correspond to those of a very simple navigation suite (table 4.1). For both simulated runs all vehicles traveled over a distance of 500 m at a speed of 1 m/s, while maintaining the initial geometry.

4.2.2 Results

Figure 4-3 shows the results for both simulated runs. Each marker "×" shows the position of all vehicles at one point in time and the big markers "✕" mark an arbitrary instant to illustrate the relative position between the vehicles. As all vehicles moved at the same speed and on the same course, this relative position was maintained throughout the entire mission. Every ten seconds one of the two CNAs broadcast its position. The AUV used all of these broadcasts to update its position running an EKF (as described in section 3.4.1) and a PF (as described in section 3.4.2) in parallel. At each time instant marked by "×" the covariance ellipse as well as the particle set at that time instant are shown. The bottom plots for both runs show how the uncertainty evolves by representing the trace of the covariance matrix $\boldsymbol{P}$. In the case of the Particle Filter (PF) the covariance was obtained by computing the weighted variance for the entire particle set as described in equation (3.16).

The two top plots in figure 4-3 show the results for a run with a very disadvantageous relative position between the CNAs and the AUV throughout the entire mission. As a result of the near-collinear configuration the updates received by the AUV only "fixed" the position in the east-west direction, while the uncertainty in the north-south direction continued to grow without bound, as illustrated in the trace

[1]For the remainder of the chapter we refer to the SCM simply as AUV to illustrate that this approach is not just applicable to the specialized mine-hunting scenario

plot of the EKF. For the PF the mission duration is too short to see if the uncertainty of the position estimate, represented by the spread of the particle set, would have kept growing unbounded as in the case of the EKF. The two bottom plots show a run in which the AUV was consistently at the tip of a perpendicular triangle. This geometry led to a small and bounded position uncertainty for both filters.

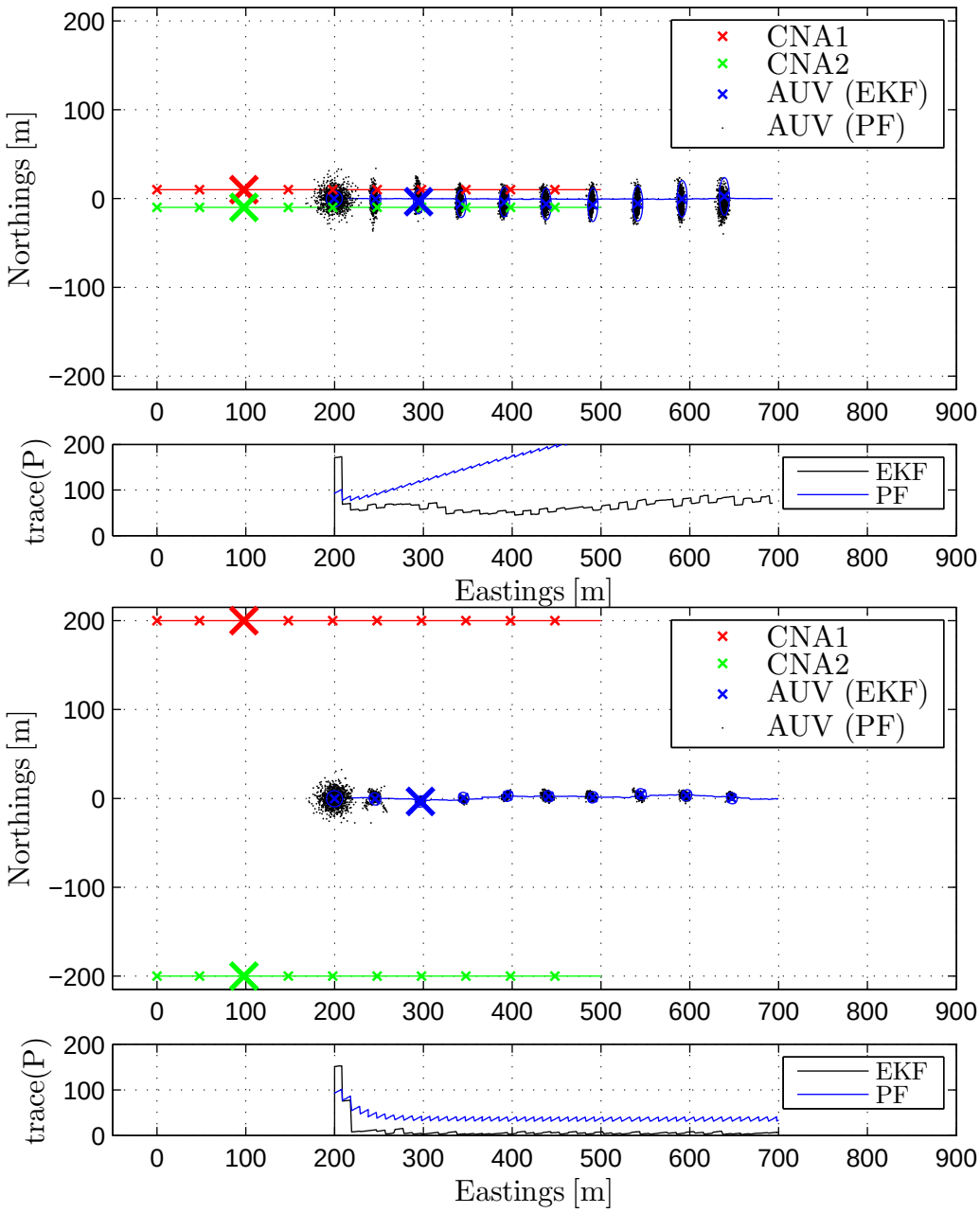


Figure 4-3: Simulated runs with two CNAs and one AUV. Top plots: A near-colinear ("bad") geometry causes the EKF's uncertainty to grow unbounded and the PF's uncertainty to remain high. Bottom plots: A "good" geometry keeps the EKF's and PF's uncertainty bounded.

4.3 Dynamic Positioning for Dedicated Beacon Vehicles

In the previous section, we showed how the intra-vehicle geometry has a great effect on the position accuracy obtainable with CN. In a hierarchical scenario with dedicated CNA-vehicles we want to control the position of the CNAs such that the overall position uncertainty of the participating AUVs is minimized. In this section we present a decentralized algorithm which runs on each CNA. It attempts to minimize the overall position uncertainty by dynamically computing future waypoints based on the position uncertainty of the participating AUVs and the predictable trajectory of other CNAs. We first introduce different metrics which can be used to quantify the position uncertainty and then explain the assumptions made for our algorithm. The remaining sections then present our algorithm and simulation results.

4.3.1 Motivation

Figure 4-3 illustrates how, by just choosing the relative position between CNAs and AUVs one can bound the AUVs' position uncertainty allowing the AUVs to remain submerged during long missions, or how the CNAs can have very little positive effect, requiring the AUVs to surface for a position fix to enforce a bounded uncertainty. This motivated the active positioning of the CNAs.

The special case of a hierarchical CN scenario in which vehicles use *consecutive* range/position pairs from moving beacons combined with dead-reckoning information, has not been the subject of substantial research. This reflects the reality that robots outside the underwater realm can usually obtain range or bearing measurements to several known landmarks at one time and combine them to a position estimate. Results from the large body of research carried out for static trilateration[2] to (certain or uncertain) landmarks can provide insights into the problem of motion planning for dedicated beacon vehicles (CNAs). As range measurements are usually readily available underwater we will focus on trilateration.

4.3.2 Related Work

In particular, GPS navigation relies on trilateration to satellites at known positions, and hence there is a large body of work that addresses the uncertainty of trilateration fixes. The metric most frequently employed is the geometric dilution of precision (GDOP). A single dimensionless number which captures the influence of the geometry on the error of the position estimate. Examples of work that investigates error metrics for GPS trilateration includes Chaffee and Abel [20] and McKay and Pachter [58]. This literature typically assumes precise knowledge of the positions of satellites, and hence the impact of the beacon position error is not analyzed. Early work in multi-robot localization, which used the concept of "portable landmarks", in

[2]Estimating a position is called *trilateration* if *range* measurements to landmarks at known positions are used and *triangulation* if the position is obtained by using *angle* measurements.

which a stationary group of robots would serve as landmarks to a moving group of landmarks, did not take the trilateration and triangulation-error into account [54]. Later, more dynamic scenarios were investigated in which all robots tried to improve their position estimates through range or angle measurements to other robots [74]. Here the uncertainty estimate of the landmarks (other robots) was used implicitly during the position estimate, but no attempt was made to explicitly analyze its magnitude and its dependence on the geometry. In feature-based Simultaneous Localization and Mapping (SLAM), the environment is typically represented in terms of discrete landmarks, and hence in such a formulation uncertainty in the landmark location is incorporated explicitly. Feature-based SLAM from range-only measurements has been addressed by several authors, including Djugash et al. [25], who developed SLAM algorithms for range-only measurements using custom sensor nodes, and Wijk and Christensen [94], who performed SLAM via trilateration using range measurements from ultrasonic sensors. In a SLAM formulation, the position estimate for the vehicle implicitly accounts for the (correlated) uncertainty in the position estimates of the landmarks. As the position of the landmarks in the range-only SLAM research cannot be controlled the topic of actively positioning the beacons to minimize the uncertainty has not yet been addressed.

Trilateration is a central component of methods to calibrate the positions of networks of sensors with range-only measurement capabilities. Moore [60] presents an algorithm which selects landmarks for trilateration such that the resultant geometry ensures a unique solution in the presence of measurement noise.

Research that specifically investigates the error of trilateration- (or triangulation) based localization techniques is fairly sparse. Kelly [52] provides a useful and intuitive insight into the relationship between robot/landmark-geometry and the resulting position accuracy, but he does not take the uncertainty of the landmarks into account. Easton and Cameron [27] explicitly assume noisy landmarks for triangulation-based algorithms, but their method does not consider the effects of strongly skewed error distributions (as shown in figure 4-4) and their effect on the optimal triangulation geometry. As a result, the uncertainty of the triangulated position chosen by their algorithm is affected by the sensor noise, but the point at which triangulation leads to the smallest possible error for given landmark uncertainties does not vary.

The special case where beacon vehicles are actively positioned to improve self-localization of other vehicles has not been the subject of much previous work. A similar case however is investigated by Trawny [86]. Using a single monolithic filter he chooses a path which minimizes the joint covariance matrix for all vehicles. In addition there is a large body of work where a single robot or a group of robots is positioning itself to track a target (Zhou and Roumeliotis [96]) explore an environment (Feder and Leonard [35]) or determine the location of a source (Christopoulos and Roumeliotis [21]). In the underwater domain Paley [68] presents several distributed control mechanisms for efficient ocean sampling. Results from this research can provide insights into the beacon positioning problem.

4.3.3 Metrics for Position Uncertainty

Three metrics are widely used to describe the error ellipse of a bivariate Gaussian distribution with a single parameter [27].

Area of the Covariance Ellipse The area A of the 3σ-error ellipse which is related to the covariance matrix $\boldsymbol{P}$ by $A = \pi * \sqrt{det(\boldsymbol{P})}$.

Ratio of the Semi-minor to the Semi-major Axis The ratio of the semi-minor to the semi-major axis of the error ellipse $\rho = \frac{\sigma_1}{\sigma_2}, (\sigma_1 < \sigma_2)$

Trace of the Covariance Matrix trace($\boldsymbol{P}$).

As these three metrics assume that the distribution of the position estimate is Gaussian, we propose as an additional metric a variation of the Circular Error Probable (CEP). The majority of published research defines the CEP as the probability, $p(R)$, that a single realization of a mean-free, bivariate Gaussian distribution $N(0, \boldsymbol{P})$ is within a circle with radius R [42]. A variation which defines the CEP as the radius $R(p)$ of the circle which contains half the realizations (for $p = 0.5$) of $N(0, \boldsymbol{P})$ is described by Torrieri [85].

$$\iint_R \mathcal{F}(\boldsymbol{x}) d\boldsymbol{x} = 0.5 \qquad R = \{|\boldsymbol{x} - E[\boldsymbol{x}]| \leq \text{CEP}\} \tag{4.1}$$

While most applications which use the CEP assume a bivariate Gaussian for the distribution $\mathcal{F}$ in equation (4.1) the key advantage of this metric is that it can be used for any 2D probability distribution.

There is no closed form solution for $p(R)$ or $R(p)$, but Shnidman provides an efficient algorithm to compute $R(p)$ in case $\mathcal{F}$ is a bivariate Gaussian [78]. Shnidman's algorithm can also be generalized for $p \neq 0.5$, but $R(p = 0.5)$ provides a parameter with an intuitive understanding of an "average error". For $\rho \ll 1$ Shnidman's algorithm can run into underflow problems, but as in this case the bivariate approaches a one-dimensional Gaussian with variance σ_2, we can compute R using the inverse error function

$$R = \sqrt{2} * \sigma_2 * \text{erf}^{-1}(\boldsymbol{P}) \tag{4.2}$$

A function to evaluate the inverse error function is provided in MATLAB and is also described in [95].

4.3.4 Trilateration from Uncertain Static Beacons

To gain insight into the effect of geometry between CNAs and AUVs on the uncertainty of the AUVs' position estimate we examine the case in which two CNAs are held in a fixed position and an AUV can position itself freely in the 2D plane (figure 4-4). The AUV then localizes itself in the plane using two range measurements (one to each CNA)[3]. Both CNAs have uncertainty associated with their positions.

[3]We assume that the ambiguity has been resolved and the correct solution of the two has been selected.

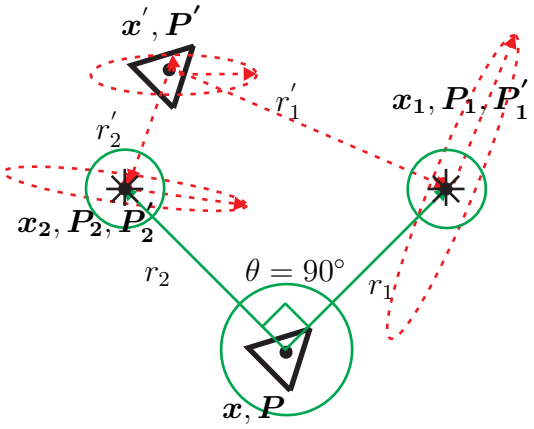


Figure 4-4: Optimal trilateration positions for different error distributions. green, solid: $\sigma_1 = \sigma_2$ and $\boldsymbol{P}_1 = \boldsymbol{P}_2$. red, dashed: $\sigma_1 \ll \sigma_2$ and $\boldsymbol{P}_1 \neq \boldsymbol{P}_2$

We then compute the error associated with the AUV's position estimate. The metric used is the CEP as described in the previous section. We then determine the optimal trilateration point $\boldsymbol{x}_{\text{opt}}$, i.e. the point with the smallest CEP. In the special case in which $\sigma_1 = \sigma_2$ for $\boldsymbol{P}_1$ and $\boldsymbol{P}_2$ the optimal position of the robot is the corner of a right-angled triangle, with the landmarks in the two other corners (green in figure 4-4) as shown by [52]. However in all other cases, in which $\sigma_1 \neq \sigma_2$ for $\boldsymbol{P}_1$ or $\boldsymbol{P}_2$ the optimal position strongly depends on the shape and orientation of the error ellipses (red in figure 4-4). We show in [7] that there is no closed form solution to obtain $\boldsymbol{x}_{\text{opt}}$ even for the simple scenario shown in figure 4-4.

4.3.5 Assumptions

Our algorithm computes the optimal future position of a CNA such that position information broadcast from this position by the CNA will reduce the combined position uncertainty of all AUVs by the largest amount. The algorithm is decentralized and as such only incorporates information which is locally available or overheard through the acoustic channel. Using decentralized algorithms is a key requirement in the underwater domain as the reliable communication channel to a single controller, as required by centralized topologies, is not available.

The metric which is minimized in this version of the algorithm is the sum of the trace differences between the prior and posterior covariances of the AUVs' position estimates. This metric assumes that the navigation algorithm running on all vehicles is an EKF as described in section 3.4.1. The algorithm however can accommodate other Bayes filters and any state representation by modifying line 6 in algorithm 7 and line 6 in algorithm 8 accordingly. Also, the metric which is minimized can be changed to other metrics such as those proposed in 4.3.3 by modifying line 5 in algorithm 9.

The following assumptions are made by the adaptive positioning algorithm:

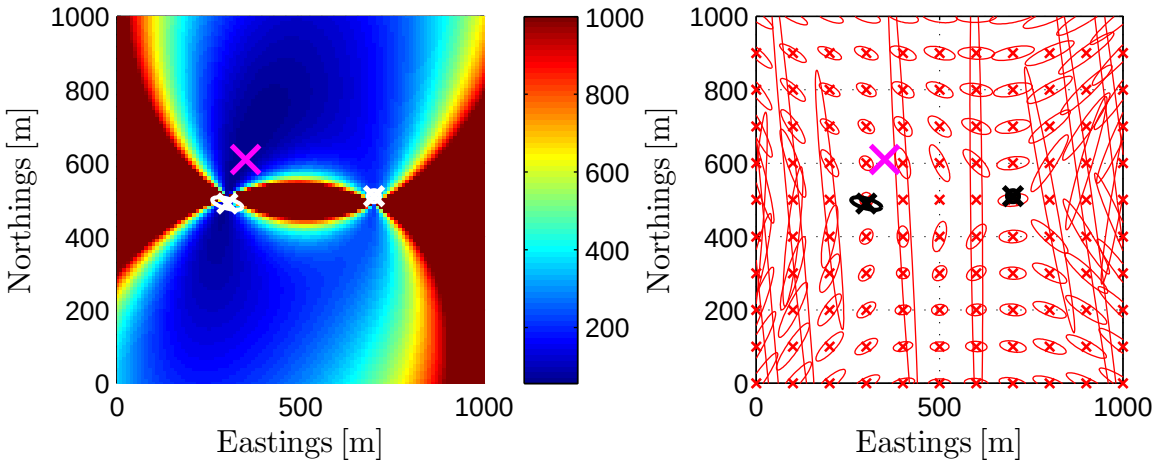


Figure 4-5: CEP and covariance ellipses for trilateration solutions. Left: the CEP for positions obtained through trilateration from the two static and uncertain CNA positions. Right: covariance ellipses for selected trilateration solutions for the same scenario. In both figures the point with the smallest CEP is marked with a magenta "×".

Vehicles

There are two groups of vehicles. A group of AUVs, $\mathcal{A}$, which carry out a mission and a group of CNAs, $\mathcal{C}$, which serve as moving navigation beacons. Optimizing the relative position between CNA and a AUVs is entirely left to the CNAs as it is assumed that each AUV's track is solely controlled by its mission objective. No CNA needs to be aware a priori of all members of the set of participating AUVs and CNAs. The sets $\mathcal{A}$ and $\mathcal{C}$ can be updated dynamically.

Communication

Each member of $\mathcal{A}$ and $\mathcal{C}$ shall be outfitted with an acoustic modem for data transmission and intra-vehicle ranging. As only one vehicle can transmit at any given time, there will be a schedule $\mathcal{S}$ which assigns a time slot during which a vehicle (CNA or AUV) can broadcast a status message. The schedule $\mathcal{S}$ is, either, provided to all vehicles before the mission starts or, in the case of a central communications controller which initializes communication through polling, the vehicles "learn" the schedule as they overhear polling requests. It is assumed that the schedule is repetitive and does not change over a longer period of time such that predictions about the time of future transmissions are possible once $\mathcal{S}$ is known. Each entry in $\mathcal{S}$ consists of a vehicle identification number, i, and a broadcast time, t_i^b, which is relative to the start of the schedule. When a vehicle i broadcasts, its transmission $\boldsymbol{m}_i$ not only contains the actual distribution over its pose estimate $\boldsymbol{x}_i$, but also its course θ_i and speed v_i or

even a short description of the upcoming mission plan. This will enable every other vehicle overhearing this message to compute a short-time prediction of the vehicle's future position. The message also contains a unique vehicle identification number i. Each vehicle also stores the predicted positions of CNAs and AUVs in the according entries in $\mathcal{A}$ or $\mathcal{C}$. While section 3.4 outlines several ways to represent the distribution of the state estimate, we chose to parametrize it through a mean and a covariance $\boldsymbol{P}_i$ as this is the most compact representation and therefore most suitable one for acoustic communication.

Sensors

Optionally, the CNAs may have available to them a sensor table $\mathcal{N}$ which contains a set of tuples, in which each tuple $n_i \in \mathcal{N}$ contains information about the i-th sensor's capabilities. If this information is available to the CNA it can also carry out short-term predictions about the future position and uncertainties of the AUVs and CNAs.

4.3.6 Algorithm

The adaptive positioning algorithm consists of four modules (Algorithm 6, 7, 8 and 9), which are run on each CNA individually when the appropriate conditions are met. Algorithm 7 and 8 both call the function algorithm 9 which computes the optimal CNA position for a given setup of CNAs and AUVs.

Algorithm 6 is run whenever the CNA receives a broadcast from an AUV.

Algorithm 7 is run whenever the CNA receives a broadcast from another CNA.

Algorithm 8 is run whenever the schedule $\mathcal{S}$ indicates that the CNA should broadcast.

Algorithm 9 is a function which computes an optimal future CNA position when the position and associated uncertainties of all CNAs and AUVs have been predicted for this time.

Message Reception from an AUV (Algorithm 6)

When a CNA receives a broadcast from an AUV, a_j, it decodes the message (line 3) and uses it to update its estimate of the future positions and associated uncertainties of a_j up to the next time t_i^b (line 4) at which the CNA is scheduled to broadcast. It achieves this by forward projection using a_j's actual position course and speed (line 5) and the information about a_j's sensor quality which is retrieved from $\mathcal{N}_i(j)$. If the received message $\boldsymbol{m}_j^A(t_0)$ from a_j contains a description of its short term mission plan an even more accurate prediction can be made. For the scenario we use to illustrate the algorithm, all predictions are based on available course and speed information. The functions $g(\cdot)$ and $h(\cdot)$ in line 5 also use the information locally stored in $\mathcal{C}_i$ so as to consider the message broadcasts from all other CNAs which occur between the

current time (t_0) and t_i^b and how they will affect the AUVs' position estimate at the time t_i^b. The updated information about a_j is stored in $\mathcal{A}_i(j, t_i^b)$ (line 6).

Require: $\mathcal{A}_i, \mathcal{C}_i, \mathcal{S}_i, \mathcal{N}_i$

1: **loop**

2: **if** message $\boldsymbol{m}_j^A$ received from AUV $a_j \in \mathcal{A}_i$ **then**

3: $$\boldsymbol{m}_j^A(t_0) = \begin{bmatrix} \boldsymbol{x}_j^A \\ \boldsymbol{P}_j^A \\ v_j^A \\ \theta_j^A \\ j \end{bmatrix}$$

4: $t_i^b = f(t_0, \mathcal{S}_i(i))$

5: $$\begin{aligned} \boldsymbol{x}_j^A(t_i^b) &= g(\boldsymbol{x}_j^A(t_0), v_j^A(t_0), \theta_j^A(t_0), t_i^b, \mathcal{C}_i) \\ \boldsymbol{P}_j^A(t_i^b) &= h(\boldsymbol{x}_j^A(t_0), \boldsymbol{P}_j^A(t_0), v_j^A(t_0), \theta_j^A(t_0), t_i^b, \mathcal{N}_i(j), \mathcal{C}_i) \end{aligned}$$

6: $t_i^b, \boldsymbol{x}_j^A(t_i^b), \boldsymbol{P}_j^A(t_i^b) \rightarrow \mathcal{A}_i(j, t_i^b)$

7: **end if**

8: **end loop**

Algorithm 6: Executed on CNA whenever a message from an AUV is received.

Message Reception from Another CNA (Algorithm 7)

When a message is received from CNA c_j it shall contain a more recent estimate of the CNA's state estimate $\boldsymbol{x}_j^C$, the associated uncertainty $\boldsymbol{P}_j^C$ as well as the actual course and speed (estimates) v_j^C and θ_j^C (line 3). The algorithm then locally emulates the effect that that specific broadcast would have had on the positioning estimate of all AUVs assuming that all AUVs received the message. This is carried out as follows: Firstly, it fetches the predicted position, $\overline{\boldsymbol{x}}_k^A$, and uncertainty estimate, $\overline{\boldsymbol{P}}_k^A$, for the actual time t_0 for each AUV in $\mathcal{A}_i$ from its AUV table (line 5). It then updates the position and uncertainty of each AUV using the Kalman state update (3.13) and the uncertainties using the Kalman covariance update (3.14) (line 6) and then stores the the resultant estimate back into the table $\mathcal{A}_i(k)$ (line 7).

Algorithm 7 then duplicates the decision making process taking place at CNA c_j. Using the communications schedule $\mathcal{S}_i(j)$, it computes the point in time, t_j^b, at which CNA c_j will broadcast again (line 9). Calling the function *compute_opt_CNA_position* (algorithm 9) with the actual position of c_j obtained from $\boldsymbol{m}_j^C(t_0)$ and our local knowledge of the future positions of the AUVs and the CNAs, we can compute the optimal position $\boldsymbol{x}_{j\_\text{opt}}^C(t_j^b)$ for c_j (line 11). If all information transmitted through the acoustic modems was received by all vehicles, then CNAs c_i and c_j will have the same positioning information available and $\boldsymbol{x}_{j\_\text{opt}}^C(t_j^b)$, computed locally by c_j, should be the same location computed by c_i. If not all values were equally shared, c_i and c_j will compute different values, but in the absence of any other information $\boldsymbol{x}_{j\_\text{opt}}^C(t_j^b)$ is the best prediction for c_j's position at t_j^b. Additionally we use the table entry

Require: $\mathcal{A}_i, \mathcal{C}_i, \mathcal{S}_i, \mathcal{N}_i$

1: **loop**
2: **if** message $\boldsymbol{m}_j^C$ received from CNA $c_j \in \mathcal{C}_i$ **then**
3: $\boldsymbol{m}_j^C(t_0) = \begin{bmatrix} \boldsymbol{x}_j^C \\ \boldsymbol{P}_j^C \\ v_j^C \\ \theta_j^C \\ j \end{bmatrix}$
4: **for all** $a_k \in \mathcal{A}_i$ **do**
5: $\mathcal{A}_i(k, t_0) \rightarrow \overline{\boldsymbol{x}}_k^A(t_0), \overline{\boldsymbol{P}}_k^A(t_0)$
6: $\overline{\boldsymbol{x}}_k^A(t_0) \overset{(3.13), \boldsymbol{x}_j^C, \boldsymbol{P}_j^C}{\longrightarrow} \boldsymbol{x}_k^A(t_0)$
$\overline{\boldsymbol{P}}_k^A(t_0) \overset{(3.14), \boldsymbol{P}_j^C, \mathcal{N}_i(k)}{\longrightarrow} \boldsymbol{P}_k^A(t_0)$
7: $\boldsymbol{x}_k^A(t_0), \boldsymbol{P}_k^A(t_0) \rightarrow \mathcal{A}_i(k, t_0)$
8: **end for**
9: $t_j^b = f(t_0, \mathcal{S}_i(j))$
10: $\boldsymbol{x}_{j\_\text{opt}}^C(t_j^b) \leftarrow$ *compute_opt_CNA_position*$\big(t_j^b, \boldsymbol{x}_j^C(t_0), \mathcal{A}_i(t_j^b), \mathcal{C}_i(t_j^b)\big)$ {Alg. 9}
11: $\boldsymbol{P}_j^C(t_j^b) = h(\boldsymbol{x}_j^C(t_0), \boldsymbol{P}_j^C(t_0), \boldsymbol{x}_{j\_\text{opt}}^C(t_j^b), \mathcal{N}_i(j))$
12: $t_j^b, \boldsymbol{x}_{j\_\text{opt}}^C(t_j^b), \boldsymbol{P}_j^C(t_j^b) \rightarrow \mathcal{C}_i(j, t_j^b)$
13: **end if**
14: **end loop**

Algorithm 7: Executed on a CNA whenever a message from another CNA is received.

for c_j's sensor noise characteristics $\mathcal{N}_i(j)$ to predict the future position uncertainty at $\boldsymbol{x}_{j\_\text{opt}}^C(t_j^b)$ (line 11). The new estimate about c_j's future positions is updated in $\mathcal{C}_i(j, t_j^b)$ (line 12).

CNA broadcast (Algorithm 8)

When the actual time, t_0, matches its scheduled broadcast time, t_i^b, CNA c_i first broadcasts a message $\boldsymbol{m}_i^C(t_0)$ containing its actual position estimate $\boldsymbol{x}_i^C$, associated covariance $\boldsymbol{P}_i^C$ as well as its actual course θ_i^C and speed v_i^C (line 3) in a similar manner to that of algorithm 7. First, the effect that this CNA's position broadcast would have on each AUV is modeled, in which it is assumed that each received the latest broadcast $\boldsymbol{m}_i^C(t_0)$ (line 5, 6 and 7). Then using the schedule $\mathcal{S}_i$ the next broadcast time t_i^b is computed (line 9). At this time all available information about the positions of each CNA and AUV at t_i^b (from $\mathcal{A}_i(t_j^b)$ and $\mathcal{C}_i(t_j^b)$) is used to determine the optimal position, $\boldsymbol{x}_{i\_\text{opt}}^C(t_i^b)$ at which the CNA's next broadcast should take place (line 10). The position uncertainty accumulated up to $\boldsymbol{x}_{i\_\text{opt}}^C(t_i^b)$ is predicted based on the actual position and uncertainty, as well as the future position and the sensor noise $\mathcal{N}_i$ (line 11). All updated information is stored in $\mathcal{C}_i(i, t_j^b)$ (line 12).

Require: $\mathcal{A}_i, \mathcal{C}_i, \mathcal{S}_i, \mathcal{N}_i$

1: **loop**
2: **if** $t_0 = t_i^b$ **then**
3: broadcast $\boldsymbol{m}_i^C(t_0) = \begin{bmatrix} \boldsymbol{x}_i^C \\ \boldsymbol{P}_i^C \\ v_i^C \\ \theta_i^C \\ i \end{bmatrix}$
4: **for all** $a_k \in \mathcal{A}_i$ **do**
5: $\mathcal{A}_i(k, t_0) \rightarrow \overline{\boldsymbol{x}}_k^A(t_0), \overline{\boldsymbol{P}}_k^A(t_0)$
6: $\overline{\boldsymbol{x}}_k^A(t_0) \overset{(3.13), \boldsymbol{x}_i^C, \boldsymbol{P}_i^C}{\rightarrow} \boldsymbol{x}_k^A(t_0)$
$\overline{\boldsymbol{P}}_k^A(t_0) \overset{(3.14), \boldsymbol{P}_i^C, \mathcal{N}_i(k)}{\rightarrow} \boldsymbol{P}_k^A(t_0)$
7: $\boldsymbol{x}_k^A(t_0), \boldsymbol{P}_k^A(t_0) \rightarrow \mathcal{A}_i(k, t_0)$
8: **end for**
9: $t_i^b = f(t_0, \mathcal{S}_i)$
10: $\boldsymbol{x}_{i\_\text{opt}}^C(t_i^b) \leftarrow$ *compute_opt_CNA_position*$\big(t_i^b, \boldsymbol{x}_i^C(t_0), \mathcal{A}_i(t_i^b), \mathcal{C}_i(t_i^b)\big)$ {Alg. 9}
11: $\boldsymbol{P}_i^C(t_i^b) = h(\boldsymbol{x}_i^C(t_0), \boldsymbol{P}_i^C(t_0), \boldsymbol{x}_{i\_\text{opt}}^C(t_i^b), \mathcal{N}_i)$
12: $t_i^b, \boldsymbol{x}_{i\_\text{opt}}^C(t_i^b), \boldsymbol{P}_i^C(t_i^b) \rightarrow \mathcal{C}_i(i, t_j^b)$
13: **end if**
14: **end loop**

Algorithm 8: Executed on a CNA whenever it is scheduled to broadcast.

Determining the Optimal CNA Position (Algorithm 9)

This function computes the optimal CNA position for a desired time, t_i^b, assuming that the predicted position of all other CNAs in $\mathcal{C}_i$ and the positions for all AUVs in $\mathcal{A}_i$ are available.

As we showed in 4.3.4 that there is no closed form solution to find the optimal beacon point, we chose a brute-force approach. The function first computes a grid of discrete positions $\boldsymbol{M}$ which could possibly be reached by the CNA before the next broadcast (line 1). The number of grid positions in $\boldsymbol{M}$ depends on the maximum speed of the vehicle, v_{max}, the time between now (t_0) and the next broadcast t_i^b and the spacing of the grid points. As the runtime of the function is *linearly* dependent on the number of grid points, the grid spacing can be varied depending on v_{max}, t_i^b and the available CPU cycles.

For each grid point, $\boldsymbol{x}_p^C$ in $\boldsymbol{M}$, we now compute by how much the overall position uncertainty would be reduced if it would broadcast from this point at t_i^b. It does this by fetching the position $\overline{\boldsymbol{x}}_k^A(t_i^b)$ for each AUV a_k (line 4) and computing the difference between the trace of the prior $\overline{\boldsymbol{P}}_k^A(t_i^b)$ and posterior covariance matrix $\boldsymbol{P}_k^A(t_i^b)$, assuming a Kalman update (3.14) by c_i from position $\boldsymbol{x}_p^C$. The trace differences for all AUVs are summed up and stored in $\boldsymbol{K}$ (line 5). $\boldsymbol{K}$ has the same size as $\boldsymbol{M}$. After the total achievable improvement has been computed for all $\overline{\boldsymbol{x}}_p^C(t_i^b)$, we determine the largest entry in $\boldsymbol{K}$. The position which maps to this entry is the optimal position

Require: $t_i^b, \boldsymbol{x}_i^C, \mathcal{A}_i, \mathcal{C}_i$

1: $\boldsymbol{M} = \left\{\boldsymbol{x}_1^C, \ldots, \boldsymbol{x}_p^C, \ldots, \boldsymbol{x}_q^C\right\} \qquad \forall\ \boldsymbol{x}_p^C\ \ s.t.\ \left\|\boldsymbol{x}_i^C - \boldsymbol{x}_p^C\right\|_2 \leq v_{imax}^C(t_i^b - t_0)$

2: **for all** $\boldsymbol{x}_p^C \in \boldsymbol{M}$ **do**

3: **for all** $a_k \in \mathcal{A}_i$ **do**

4: $\mathcal{A}_i(k, t_i^b) \rightarrow \overline{\boldsymbol{x}}_k^A(t_i^b), \overline{\boldsymbol{P}}_k^A(t_i^b)$

5: $\boldsymbol{K}(p) = \sum_k \text{trace}\left(\overline{\boldsymbol{P}}_k^A(t_i^b) - \boldsymbol{P}_k^A(t_i^b)\Big|_{\overline{\boldsymbol{P}}_k^A(t_i^b) \overset{(3.14),\boldsymbol{x}_p^C,\boldsymbol{P}_i^C}{\rightarrow} \boldsymbol{P}_k^A(t_i^b)}\right)$

6: **end for**

7: **end for**

8: $\boldsymbol{M} \overset{\max(\boldsymbol{K})}{\rightarrow} \boldsymbol{x}_{p\_opt}^C(t_i^b)$

9: **return** $\boldsymbol{x}_{p\_opt}^C(t_i^b)$

Algorithm 9: Compute the optimal position $\boldsymbol{x}_{opt}^C$ for a CNA c_i for a predicted time t_i^b. It assumes that the position and uncertainties for all other vehicles (CNAs and AUVs) are given by $\mathcal{A}_i$ and $\mathcal{C}_i$.

Table 4.2: Sensor noise and maximum speed of the simulated vehicles used in the adaptive positioning simulation (figure 4-6 and 4-7).

Vehicle	σ_u,σ_v	σ_θ	σ_r	v_{max}	Notes	
CNA 1	0 m/s	0°	2 m	1.5 m/s		CNAs have GPS
CNA 2	0 m/s	0°	2 m	1.5 m/s	not in scenario 1	
AUV 1	0.2 m/s	10°	1 m	1 m/s		
AUV 2	0.2 m/s	10°	1 m	1 m/s	not in scenario 1	

$\boldsymbol{x}_{p\_opt}^C$ to which the CNA should move so as to maximally reduce the uncertainty of the AUV set (line 8).

4.3.7 Results

To test this adaptive positioning algorithm we simulat two scenarios. The first scenario (figure 4-6) consists of one AUV and one CNA, in which both vehicles start at the same point and the AUV mission takes it on a straight west-east trajectory for 400 m. The second scenario (figure 4-7) uses two AUVs and two CNAs. All vehicles start at the same point with AUV 1 moving north for 100 m and AUV 2 moving south for 100 m. Both AUVs then move on a west-east trajectory while maintaining their 200 m separation. The simulated sensor noise is equivalent to an AUV with an inexpensive navigation suite. The variances of the sensor noise for both simulations are the same as those used in section 4.1.2 and are shown in table 4.2.

One AUV, One CNA

Figure 4-6 shows the simulation results for the most basic possible CN setup, one CNA and one AUV. Every 60 seconds the CNA broadcasts its position and then computes the optimal position for the next broadcast. As there are no other CNAs present, the CNA only needs to take the effect of its own updates and the vehicles' sensor performance into account. The top plot, at t=20 s, shows the situation directly after the mission commenced. The CNA has just broadcast its position and the position it predicts for the AUV at the next broadcast which is marked with red "+". The semi-transparent circle with radius $r = \Delta t \cdot v_{max} = 60\,\mathrm{s} \cdot 2\,\mathrm{m/s} = 120\,\mathrm{m}$ marks all positions which the CNA could reach at maximum speed. Our algorithm discretizes this circle into grid points with 5 m spacing. It then computes, for each grid point, the position uncertainty which the AUV would have *after* a hypothetical update broadcast by the CNA from this grid position. The difference between the prior and posterior trace of the AUVs' position estimate is represented by the color of the semi-transparent circle. Positions marked blue would lead to a very small decrease in overall uncertainty and positions marked red would lead to a very high overall decrease. The mapping between the absolute value of $\boldsymbol{K}(p)$ and the color is scaled, each time the circle is plotted, to span the maximum color space. Thus we cannot provide a legend which maps colors to absolute values for $\boldsymbol{K}(p)$. The position which corresponds to the maximum of that difference is selected as the future position for he CNA.

As the AUV has a high variance in its heading direction it accumulates the highest uncertainty in the direction perpendicular to the direction it is traveling in. As shown by Zhou and Roumeliotis in [96], the biggest decrease in the trace of the covariance can be achieved if the beacon vehicle is somewhere along the semi-major axis of the AUV's covariance ellipse. Brute-force computation confirms this, by highly favoring positions perpendicular to the direction in which the AUV is traveling, illustrated in dark red, for the first update. At t=72 s (middle plot) the CNA has reached its planned position. The AUV has reached its predicted position and the CNA has transmitted its message and computed a new optimal broadcast position for its new message. As the previous broadcast, at t=70 s, strongly reduced the error in the north-south direction, the along-track error will dominate the position uncertainty and the optimal position is in line with the vehicle traveling. The bottom plot, at t=320 s, shows the vehicles after the fifth broadcast. At this stage a "saw-tooth" pattern has been established, in which the CNA oscillates between the two relative positions which can be seen in the top and middle plot.

Due to the much larger distances that the CNA has to travel in this scenario, compared to those of the AUV, the distance between the CNA and the AUV slowly increases, as reaching the optimal relative position is the CNA's only goal. Section 4.4 addresses how the course and speed of the CNA can be determined if reaching the optimal position is only one of the CNA's objectives. Another objective, for example, would enforce a minimum distance between the vehicles.

Two AUVs, Two CNAs

A more complex CN-scenario is shown in figure 4-7. Here, two CNAs try to jointly optimize their trajectory to improve the position uncertainty for two AUVs. All four vehicles start at the same position and both CNAs broadcast their position every 30 s. After CNA 1 broadcasts its first message, at t=10 s, it determines that the position marked by the blue "+" is the optimal position for its next broadcast. Meanwhile CNA 2 waits until its first broadcast, at t=40 s, and then determines its optimal position for its next broadcast at t=100 s (cyan "+"). When computing the trace difference represented by the semi-transparent circle in the middle plot (the corresponding circle for CNA 1 is not shown as they would overlap), CNA 2 takes the effects of the broadcast from CNA 1 at t=70 s into account, as otherwise it would also head for the optimal position previously computed by CNA 1, leading to a redundant update. Shortly after CNA 2 reaches its computed position, all four vehicles achieve the stable position of a quadrilateral which is maintained throughout the mission and can be seen in the bottom plot. The bottom plot also shows the two predicted positions for both AUVs.

4.4 Optimal Positioning in a Multi-Objective Decision Scenario

The "one AUV, one CNA scenario" depicted in figure 4-6 shows how optimizing the trajectory for the short-term optimal broadcast position alone can lead to a sub-optimal long-term solution as the distance between the vehicles constantly grows until the distance is too long for transmission. Therefore we would also like the dynamic positioning of our CNAs to be influenced by other objectives such as maintaining a minimum distance to all vehicles. If the acoustic propagation conditions are known, choosing the broadcast position such that the transmission loss to all vehicles is minimized could be another possible objective.

The problem of selecting an action for an agent, in our case the speed and course of our CNA, in a situation in which several objectives have to be satisfied has been the subject of extensive research [8], [3]. These methods typically switch between satisfying the different goals individually or perform averaging which does not necessarily lead to the optimal solution as illustrated in [12].

More recently Benjamin developed the Interval Programming (IVP)-method which can compute an optimal solution for a set of piece-wise linear objective functions [13, 14]. This method was implemented in the Mission Oriented Operating Suite (MOOS)-environment, a software suite developed by Newman [64] and used by MIT to control various land, surface and underwater robots. This implementation was tested in several different scenarios and has demonstrated an ASC successfully reaching a waypoint while observing the "rules of the road" [11] and tracking underwater targets with a towed array while ensuring that its maneuvers would not damage the array [15].

The output of our adaptive positioning algorithm can be directly expressed as an

objective function which the IVP method controlling the CNA could combine with other objective functions.

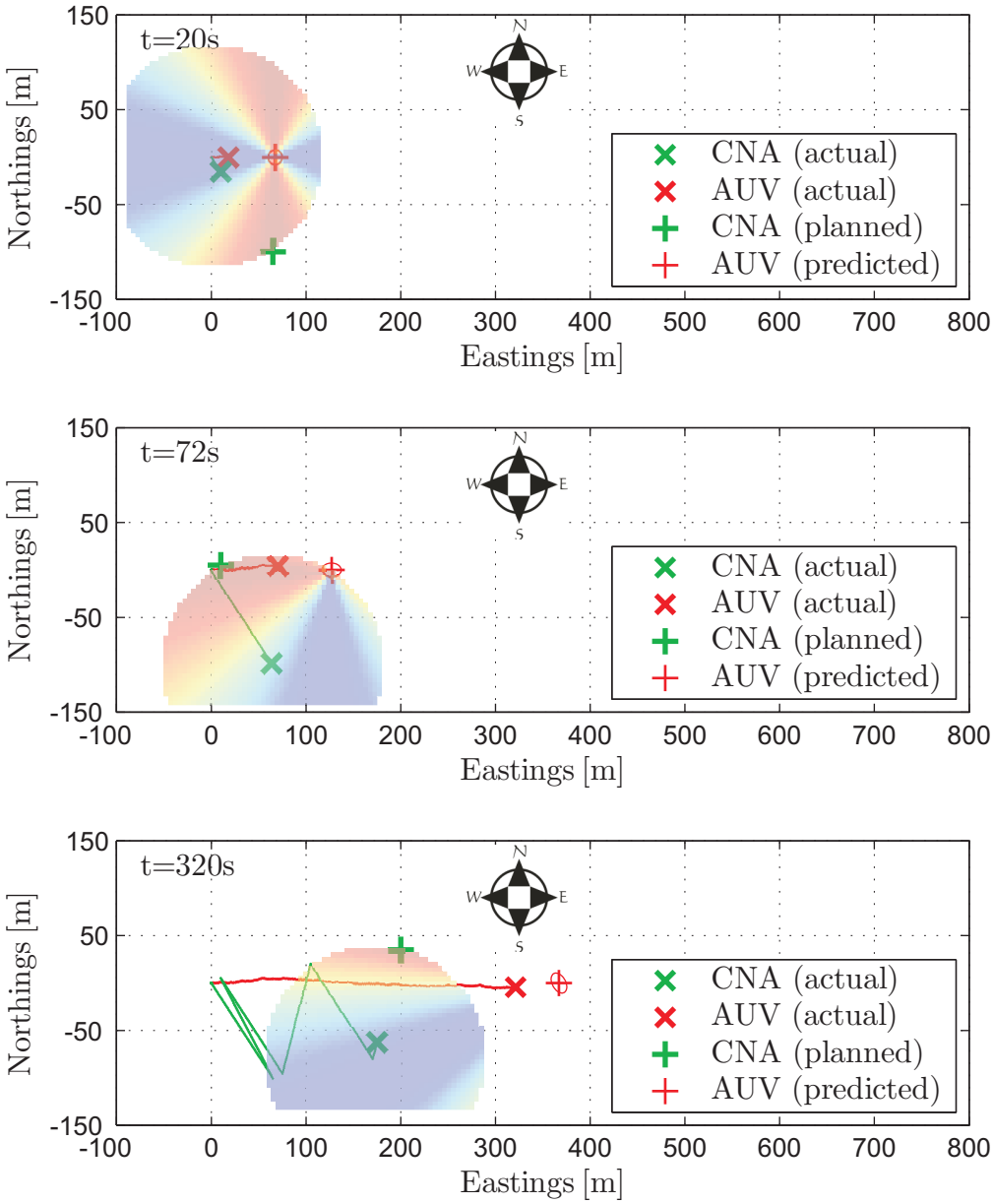


Figure 4-6: One CNA one AUV in an adaptive motion control simulation. The CNA automatically adapts its position to be in a position during the broadcast which minimizes the position uncertainty of the AUV.

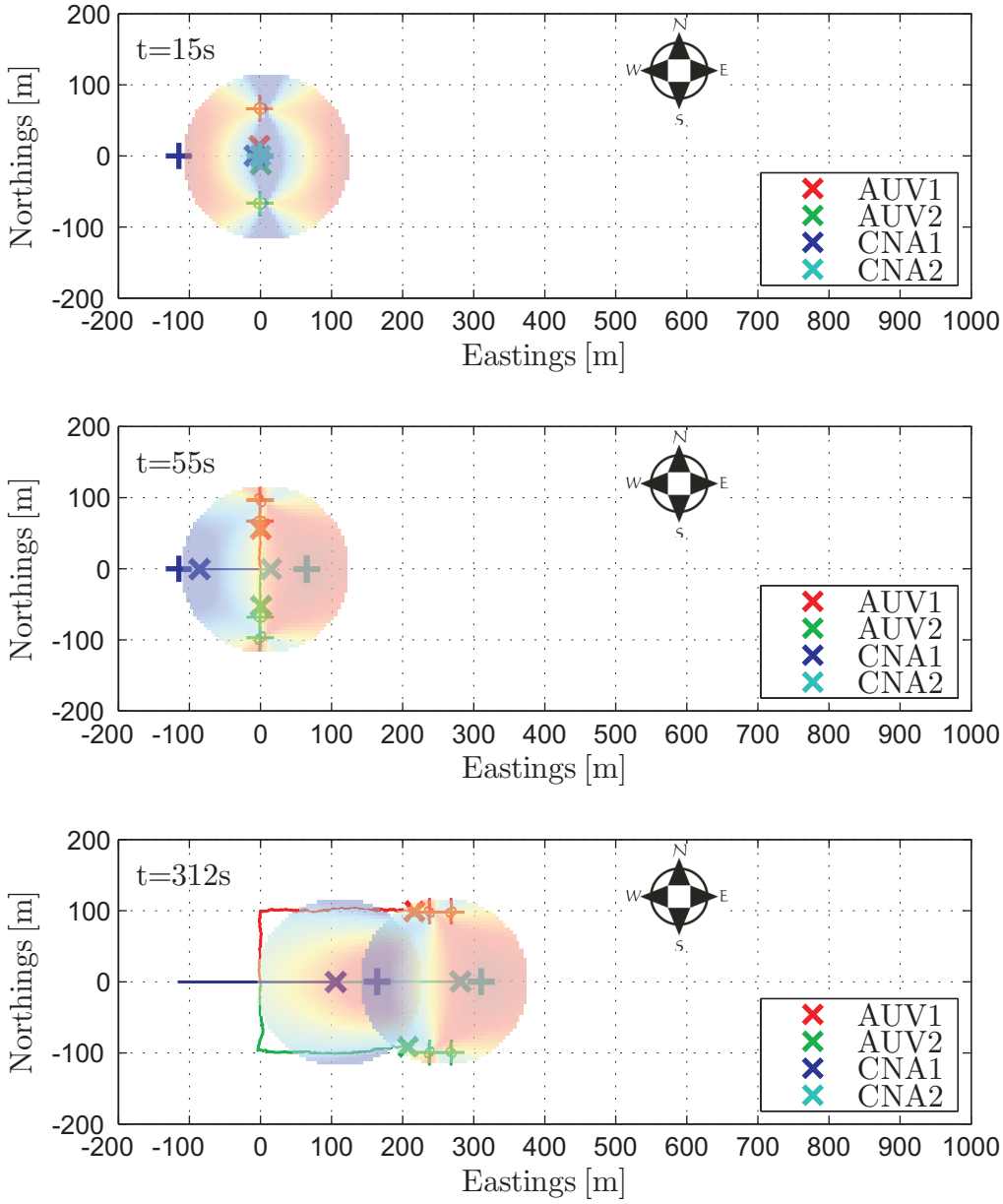


Figure 4-7: Two CNA, two AUV in an adaptive motion control simulation. The CNAs automatically adapt their position to be in a position during the broadcast which minimizes the position uncertainty of both AUVs.

Chapter 5

Experiments

The following chapter shows the results of a series of experiments which were part of the CN research. As the modem's capability to precisely measure the distance between two vehicles outfitted with them is key for all CN experiments we first carried out an experiment to determine the ranging accuracy as well as the maximum obtainable vehicle-vehicle distance. In the following experiments we collected the necessary data to run the localization algorithms outlined in 3.4 in post processing to asses their performance and detect possible failure modes.

The first of these experiments used surface crafts as CNAs such that the algorithms' results could be compared against ground truth (GPS). The second and third experiments involved two types of underwater vehicles using a surface craft as CNA. One was a buoyancy driven glider, the other a propelled AUV. All three vehicles and their capabilities are described in the following sections.

5.1 Modem Ranging Test

5.1.1 Setup

All Cooperative Navigation approaches outlined in this thesis require the participating vehicles to communicate their position and obtain intra-vehicle range measurements. As a result the performance of all CN-algorithms will be strongly dependent on the performance of the WHOI acoustic modem, which will be used for all upcoming CN experiments. In order to asses the accuracy of the modem's ranging capability we set up a test in Lake Grey, ME. with three "SCOUT" ASCs. The ASCs are shown in figure 5-1b and 5-3a and described in [23]. They consist of a commercial kayak hull outfitted with a thruster, a mini-ATX PC, GPS and the WHOI acoustic modem which is also used on the AUVs and glider. The vehicle dynamics of the ASC are comparable to those of a mid-sized AUV. By using only the acoustic modem to exchange information and estimate ranges between the two vehicles, we have applied the same restrictions which are encountered in an AUV-only scenario while at the same time we are able to compare the algorithm's navigation performance against the "true" GPS position. Figure 5-1a shows the modem transducer strapped onto the hull. This was the configuration used during the Lake Grey test. Having the

(a) Modem transducer strapped to the hull

(b) Two kayaks moored on the pier while the a third one moves and determines its distance to the moored ones using the acoustic modem

Figure 5-1: Setup for modem test on Lake Grey, ME, October 2004

transducer closely coupled to the vibrating hull and close to the water surface limited the maximum possible range to 500 m. As a result, all future experiments with the ASCs had the transducer mounted into a towfish which was hanging about 2 m below the keel (figure 5-3b).

During the first experiment, two of the ASCs ("Bobby" and "Charlie") were moored in the open water at a 100 m distance while the third kayak ("Andy") first moved away and then toward the two ASCs in an east-west direction (top plot in figure 5-2) . The distance between "Andy" and both other ASCs was between 100 m and 700 m. During the entire run "Andy' alternated between sending query pings to "Bobby" and "Charlie". If a reply was received, Andy was able to obtain the TOF.

5.1.2 Results

The moving ASC ("Andy") was able to reliably obtain ranging information to both vehicles for a distance of up to 400 m. During a short transition period between 400 m and 500 m around half of the ranging attempts were successful, but no ranging was possible beyond 500 m (figure 5-2 center plots). The fact that the maximum obtainable range was surprisingly low was mostly attributed the mounting of the transducer which, as a result, was later changed to the towfish configuration (figure 5-3b).

The two center plots in figure 5-2 show the range between "Andy" and "Bobby" as determined by the modem ("∘") and by the GPS. We first note that the 109 intra-vehicle range measurements ("Andy"-"Bobby": 60; "Andy"-"Charlie": 49) do not contain a single outlier. If the range between the vehicles as derived by the GPS position is taken as ground-truth the relative range error is around 1 %.

We concluded that the range accuracy obtained in this experiment was sufficient for Cooperative Navigation, but that different transducer mounting was necessary if the ASCs were to be used as CNAs.

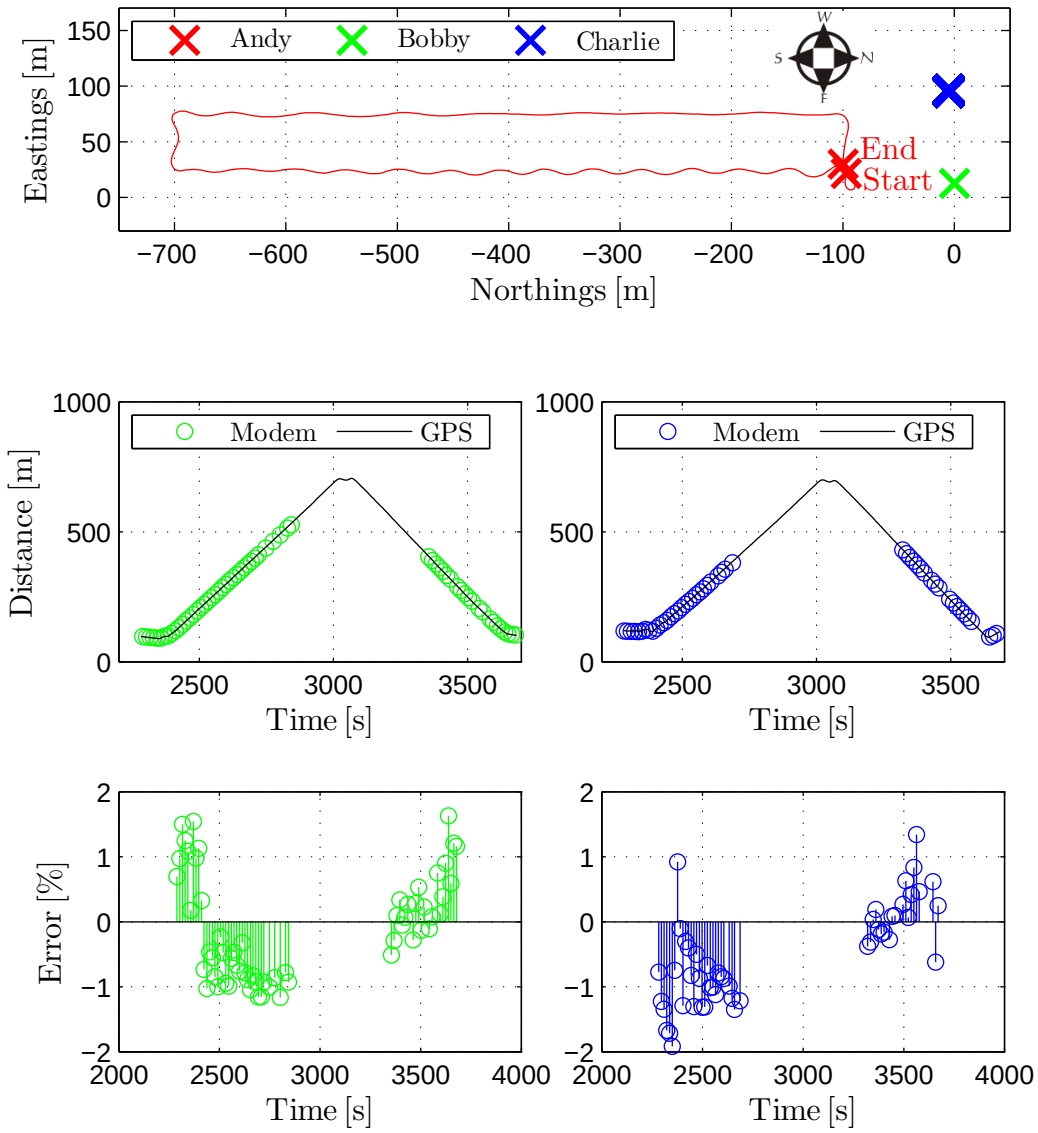


Figure 5-2: Modem range test. One kayak ("Andy") traveled on a U-shaped mission while ranging to the two other moored kayaks ("Bobby" and "Charlie"). The top plot shows the trajectory of Andy, the bottom plots show the range as determined by the acoustic modem and the ground truth (GPS).

(a) Three kayaks navigating co-operatively

(b) Towfish with modem transducer

5.2 CN Using Surface Crafts Only

5.2.1 Setup

To obtain a data set which also contained ground-truth for CN in the form of GPS measurements, three ASCs as described in the previous section were set up to run in formation along a trackline while broadcasting their position information over the acoustic modem. Each ASC in the formation was able to participate actively, by sending information, and passively by computing its position estimate based on the information obtained from the other two, but the results are only shown for one ASC of the formation. In this case two kayaks act as the "CNAs" while the other kayak acts as the "AUV". In the setup shown in figure 5-3a the center kayak ran a preprogrammed mission using its GPS for navigation. The other two kayaks followed in a predetermined formation in order to stay within range of the acoustic modems. The position/range-pairs obtained from the two CNAs over the acoustic modem were logged by the AUV-kayak and the algorithm was used to compute position fixes in post-processing. The data set was obtained during AUVFest 2005 in Seattle, WA.

5.2.2 Results

Post-processing the data logged on the ASC acting as a surrogate for an AUV we computed the position estimate whenever a broadcast from any of the two CNAs was successfully received. Figure 5-3 shows the GPS track of the ASC and the computed positions with their associated error ellipses. The tracks of the CNAs are not shown. Figure 5-4 shows the error of the computed position, the distance between the computed and the GPS position. When comparing the GPS derived distance with the modem derived one we noticed larger errors for measurements obtained at position #6, #15 and #16 which then led to larger errors in the computed position.

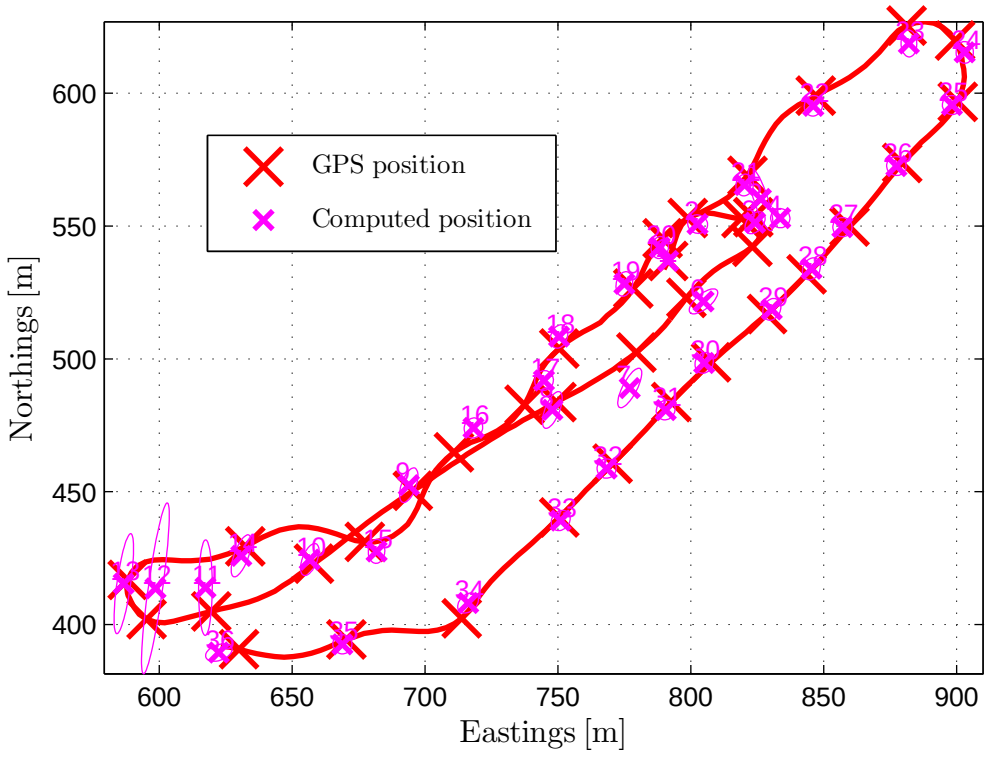


Figure 5-3: GPS tracks of CNA (ASC) and computed positions

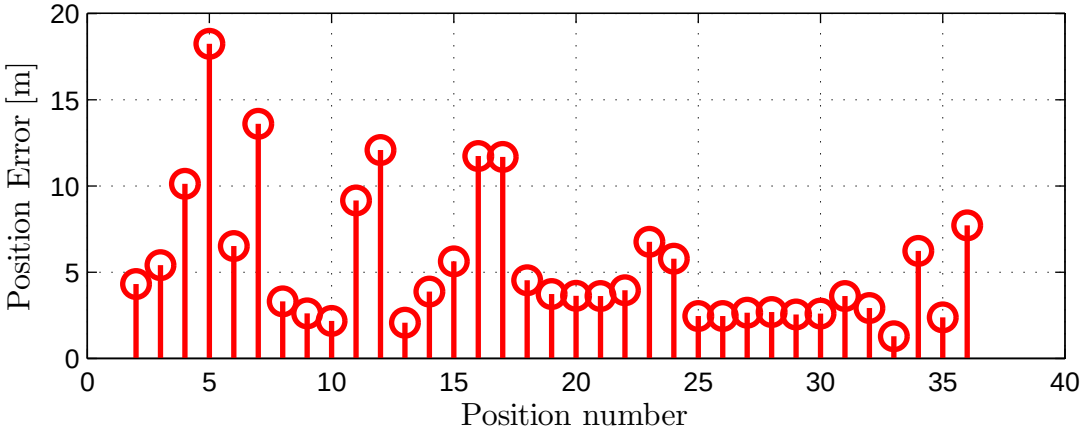
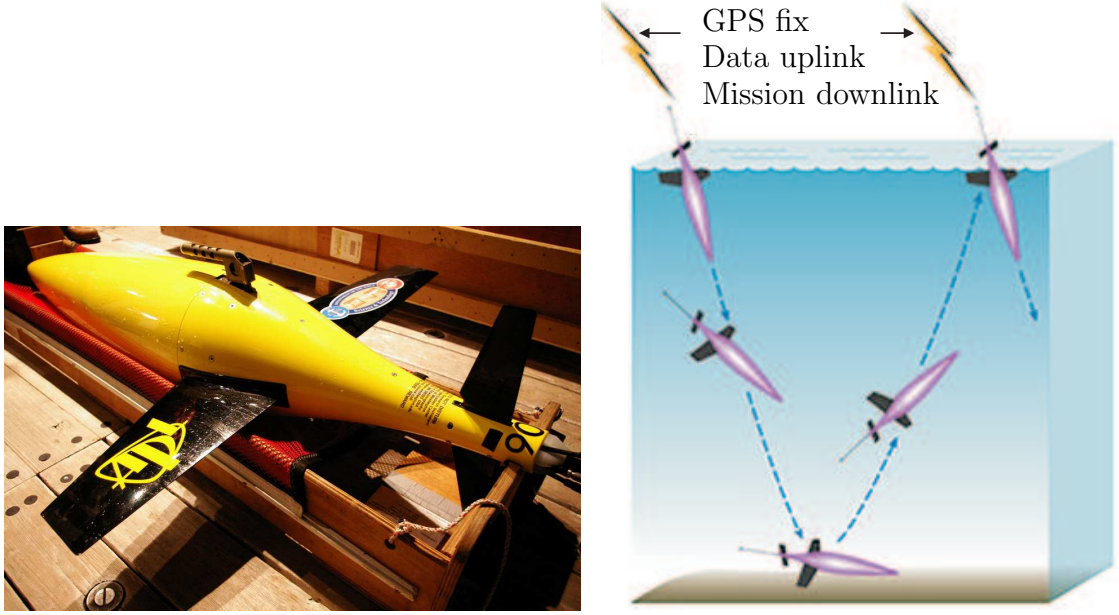


Figure 5-4: Distance between GPS derived position and computed position



(a) Seaglider on deck after recovery

(b) Seaglider dive profile. Illustration courtesy of *APL-University of Washington*

Figure 5-5: University of Washington - Applied Physics Lab's *Seaglider*

5.3 ASCs and an underwater glider

5.3.1 Setup

The second CN experiment which took place during the MB06 experiment in Monterey Bay, CA in August 2006 involved two ASCs as described in the previous sections and an underwater glider operated by the Applied Physics Lab of the University of Washington (APL-UW) (see section 1.1.1 for a detailed description of a glider). The low power consumption ($\approx$ 1 W) makes for very long duration missions which can last up to half a year. While on the surface, the glider can reset its navigation using a GPS, but during the dive the very small power budget only allows for very simple navigation sensors such as a depth sensor and a compass. The information from these sensors together with a vehicle model is used to compute dead-reckoning navigation information. The position estimate derived from these sensors can drift at a rate of up to 30 % of distance traveled, especially when underwater currents are present. As a result the drift rate can lead to a large cumulative navigation error during a dive which can typically last up to several hours. This makes a glider particularly suited for cooperative navigation as in a scenario with several gliders, a surfaced glider with access to GPS could provide navigation information for every submerged glider within communication range. While the power consumption of an acoustic modem is very high during transmission ($\approx$ 20 W), only a small number of these transmissions would occur while the glider is on the surface which takes place about every 2 h. In receive mode the power consumption drops to 0.1 W. As a result an acoustic modem would

only add about 10-15 % to a glider's power budget. During the MB06 experiment a modem was added to a glider for the first time. As the modem was only capable of logging information and did not have access to the glider's main vehicle computer (which provides the dead-reckoning information), on-board processing was not possible. The ASCs measured the range to the glider and by combining the logs from the kayaks, the glider's Main Vehicle Computer and the glider's log of the modem traffic it was possible to compute post-processed solutions of the glider's positions. The shallow water of Monterey Bay prohibited dives deeper than 30 m. As the distance traveled in horizontal direction during a single dive is directly proportional to the maximum achievable depth, the depth limit only allowed for transects which were about 100 m long. The main goal of the experiment was to demonstrate the feasibility of glider communication for navigation purposes. Future experiments will involve longer and deeper dives leading to longer transsects.

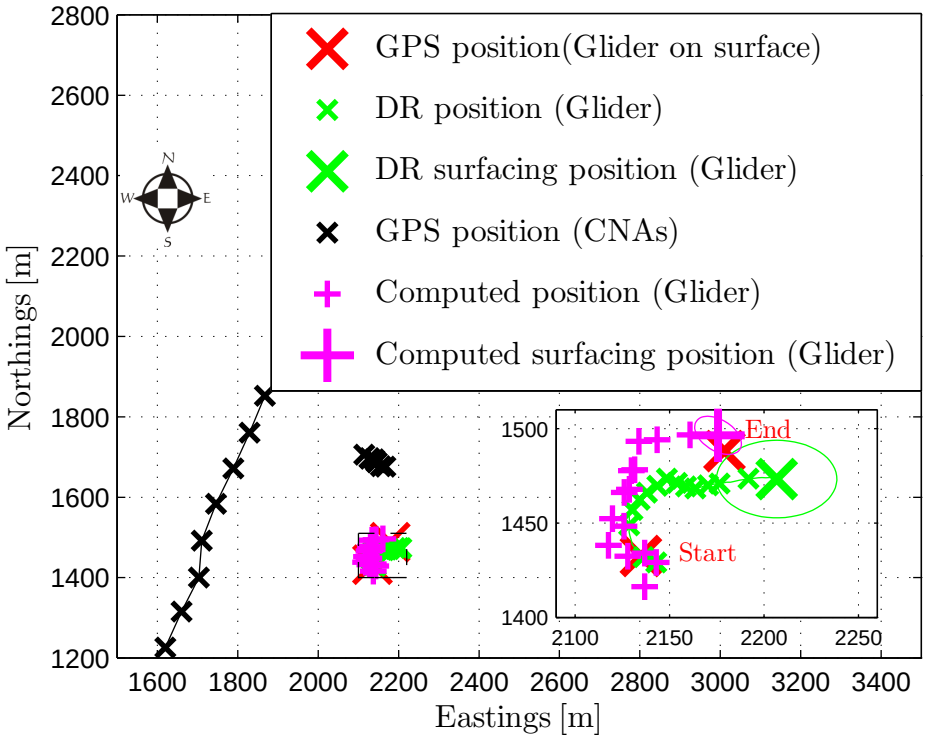
5.3.2 Results

As described earlier, the shallow depth of Monterey Bay only allowed for shallow diving depths and, as a result, very short transects of the glider. Figure 5-6a shows the positions of the two ASCs (acting as CNAs) as well as the dead-reckoned and computed positions of the glider. The inset shows a detailed view of the glider track (dead-reckoned and computed). The GPS fixes mark the last GPS derived position before the glider submerged as well as the first one after it surfaced. Figures 5-6b through 5-6d show the depth, the dead-reckoned position and the computed position of the glider as well as the associated uncertainties for different time instances during the mission. Due to the short transect the cumulative error of the dead-reckoned position is not significantly above the uncertainty of the computed position, however the computed position just before surfacing is much closer to the GPS surfacing position than the dead-reckoned one (figure 5-6d). Future experiments involving longer dives with transects of several kilometers in length should lead to significant differences between the dead-reckoned and the computed position.

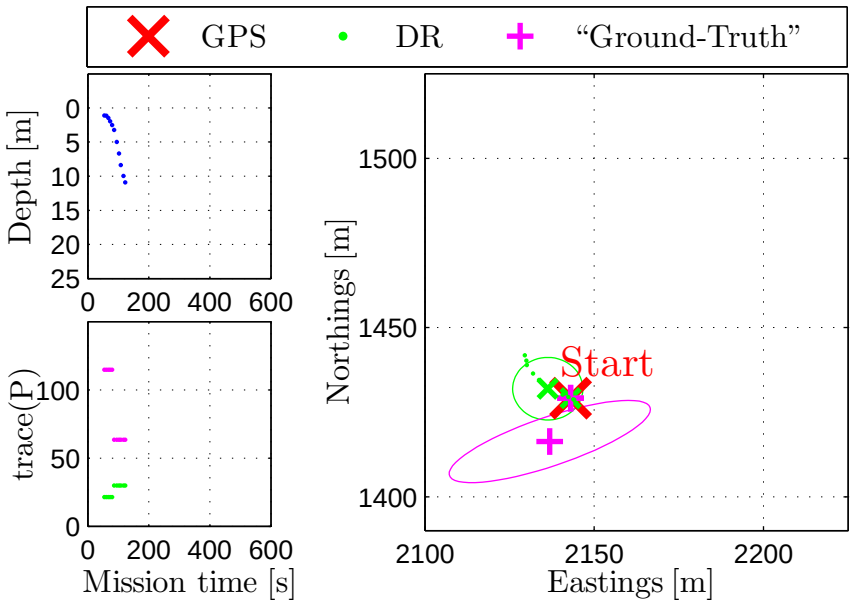
5.4 ASCs and an AUV

5.4.1 Setup

During a demonstration at the *Naval Surface Warfare Center (NSWC)* in Panama City, FL, USA in December 2006, two ASCs and a Bluefin 12" AUV (figure 5-7) ran several missions in which the ASCs acted as CNAs and followed the AUV while sending their GPS-derived positions over the acoustic modem. The AUV also obtained distances to the transmitting ASC and stored both information for post processing. Ground truth was not directly available, but by post-processing (provided by Bluefin)

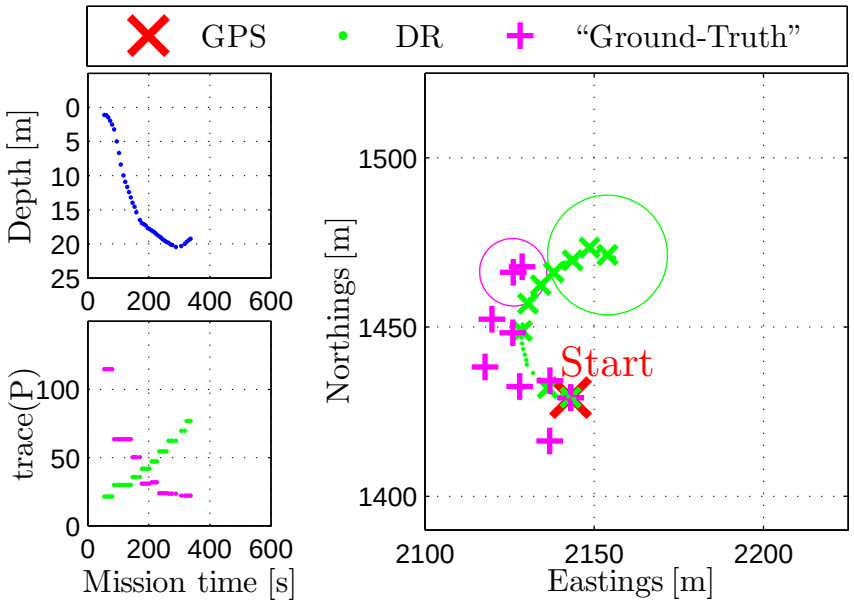


(a) Glider/ASC mission overview: The evolution of the computed and dead-reckoned position and the associated uncertainties are illustrated in the the figures 5-6b through 5-6d.

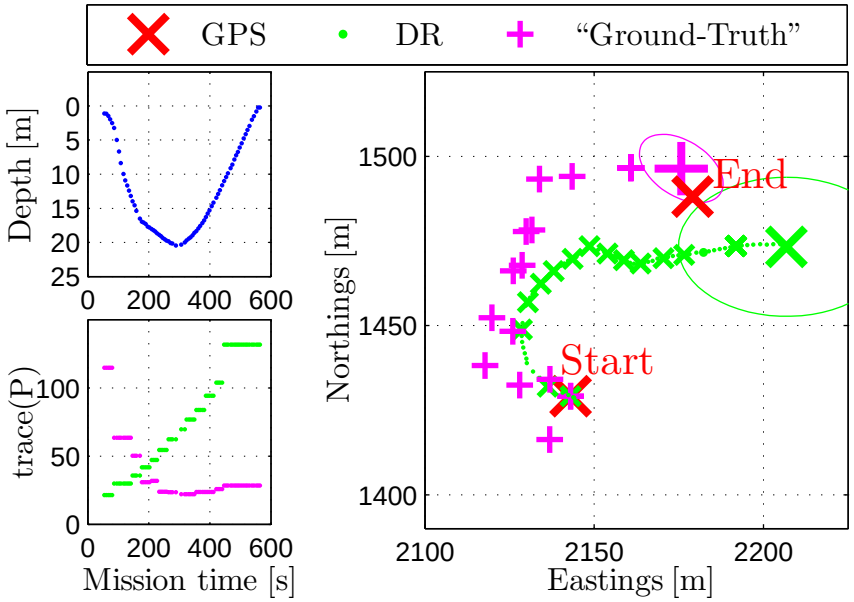


(b) Glider/ASC mission at t=123 s. The uncertainty of the initial estimate of the CN algorithm is large when compared to the dead-reckoning uncertainty.

data from the sophisticated and well calibrated sensor package and including the position obtained through the GPS after surfacing, accurate navigation information was



(c) Glider/ASC mission at t=337 s. The uncertainty of the computed position remains bounded while the uncertainty of the dead-reckoned estimate grows without bound.



(d) Glider/ASC mission at t=564 s (after surfacing). The ground-truth position is significantly closer to the computed position then the dead-reckoned position.

Figure 5-6: Glider/ASC mission.

available which was used to compare the results of the CN algorithm.

A total of 16 cooperative navigation missions were run during which the AUV received the CNA's position and measured the CNA-AUV range. During these runs the AUV acted as a master and requested a new position every 30 seconds switching

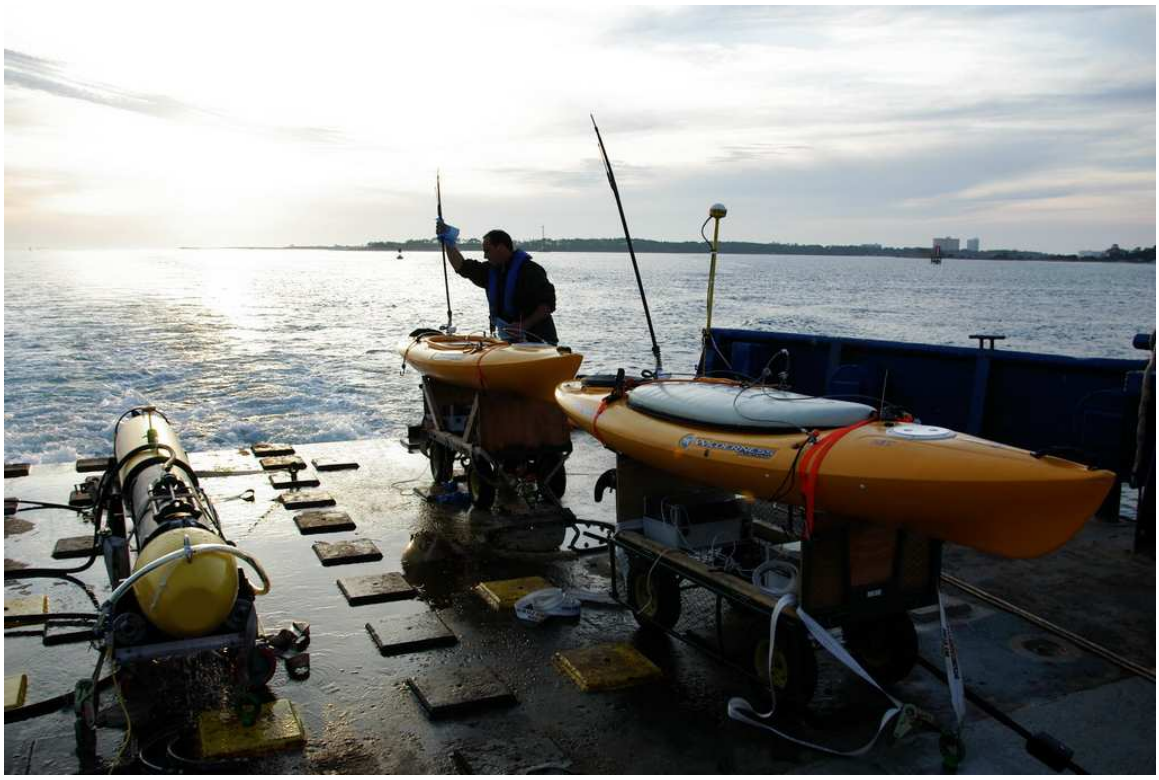


Figure 5-7: Two MIT ASCs and one Bluefin 12" AUV.

between the two CNAs. Of all positions requested the AUV would receive about 60 %. For the remaining 40 % of the queries the CNA did either not receive the request or the AUV did not receive the CNA's answer. Sometimes the AUV would also suspend requesting positions because it needed to transfer other mission specific information over the acoustic modem. As a result the update rate for position/range pairs was about one per minute.

After requesting a position/range from both CNAs, the AUV would send its own position estimate over the acoustic modem. Furthermore, the CNAs would continuously broadcast their GPS-derived position over the radio such that both CNAs were aware of where the other one is. Knowing where the AUV and the other CNA is, enabled the CNAs to follow the AUV in a way that was optimal for cooperative navigation:

- In order to maintain optimal acoustic communication, the AUV would try to stay 150 m behind the AUV.
- To minimize the covariance of the computed solution the CNAs would try to form a right-angled triangle with the AUV in the corner with the right angle and the CNAs in the other two.

As the AUV's position updates were received at a rate of only $O(1/\text{min})$, it was very difficult for the CNAs to maintain the triangular formation when the straight

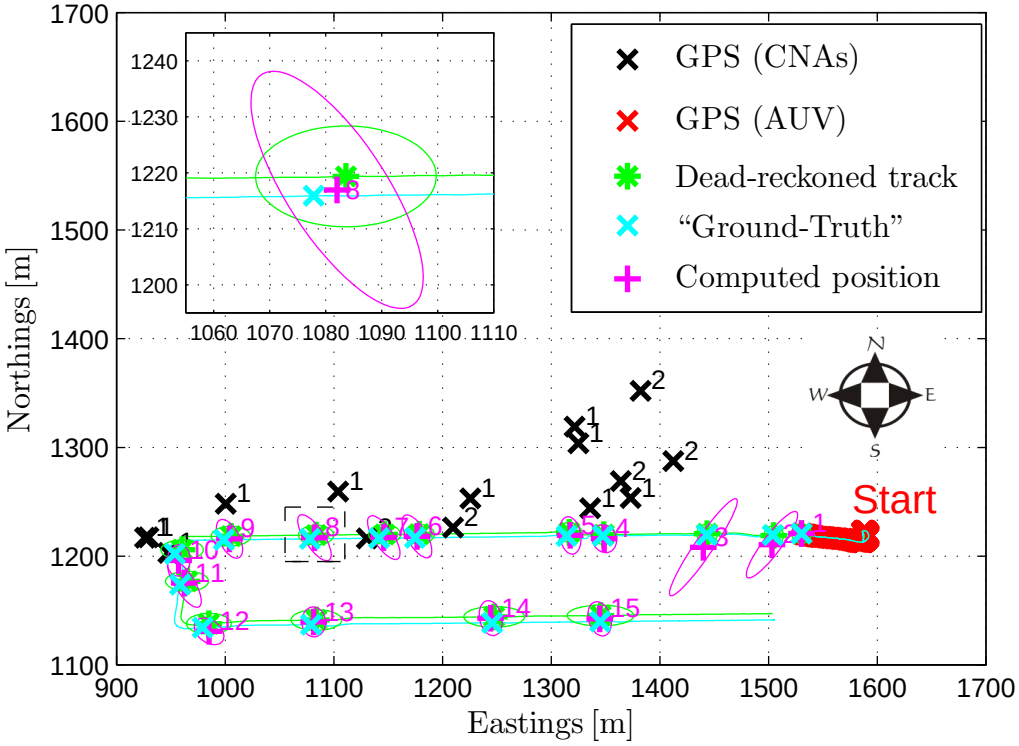


Figure 5-8: AUV/ASC mission 1: Dead-reckoned track and computed positions of AUV, GPS positions of CNAs; Inset: Detailed view of position #8.

transects were short (figure 5-8). During the second mission (figure 5-9) CNA1 was able to maintain an aft-starboard position with respect to the AUV while, CNA2 maintained an aft-port position. Even when the formation was not maintained the AUV's broadcast enabled the CNAs to stay close enough to maintain the acoustic communication channel. The navigation error was modeled using sensor noise as provided in [31]. While the results for only two runs are shown in figure 5-8 and figure 5-9 the quality of the results computed by the algorithm was the same for all 16 runs.

5.4.2 Results

Figure 5-8 and figure 5-9 show two of the missions carried out. The first mission consisted of a U-shaped trackline of about 1 km length. After initializing its position with GPS the AUV submerged to a depth of about 12 m and ran the mission at a constant speed of 1.5 m/s. The detail in figure 5-8 shows the computed position #8 and its covariance ellipse. Also shown is the "ground-truth" track as well as the "ground-truth" position estimate at the time of the computed solution. As the "ground-truth" position is based on post-processed dead-reckoning data the distance

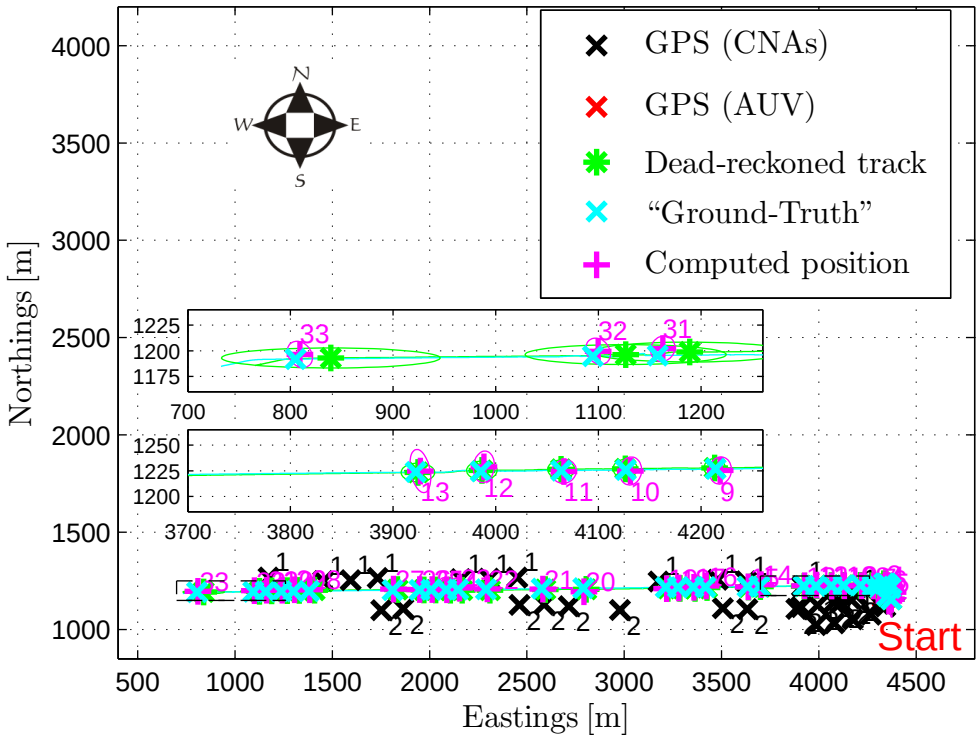


Figure 5-9: AUV/ASC mission 2: Dead-reckoned track and computed positions of AUV, GPS positions of CNAs; Inset (top): Detailed view of mission end (positions #31, #32 and #33) ; Inset (bottom): Detailed view of mission start (positions #9 through #13).

between it and the computed position can only provide a qualitative assessment of the algorithm's performance. As a result we did not compute the Euclidean distance between the two positions. Also, the post-processed track is the result of a non-linear optimization so no covariance estimate can be provided.

The second mission consisted of a 4 km east-to-west trackline. During this mission the kayaks were able to maintain the triangular formation for most of the time. On five occasions during this mission the AUV would spend four minutes transmitting mission specific data. During this time no positions were queried from the CNA which lead to the wide gaps between the computed solutions (e.g. between #19 and #20 as well as #27 and #28). The two insets in figure 5-9 show two magnified views of the track at the same scale. The bottom one near the beginning (eastern end) of the mission and the top one of the end (west). These illustrate how beneficial the information from the CNAs is for navigation accuracy. In the beginning the dead-reckoned position is very close to the "ground-truth" and the computed solution while at the end of trackline the "ground-truth" as well as the computed position have consistently moved away from the dead-reckoned position. The dead-reckoning error, represented by the growing error ellipse, depends on the distance traveled and will grow without bound if the AUV is submerged, while the error of the computed

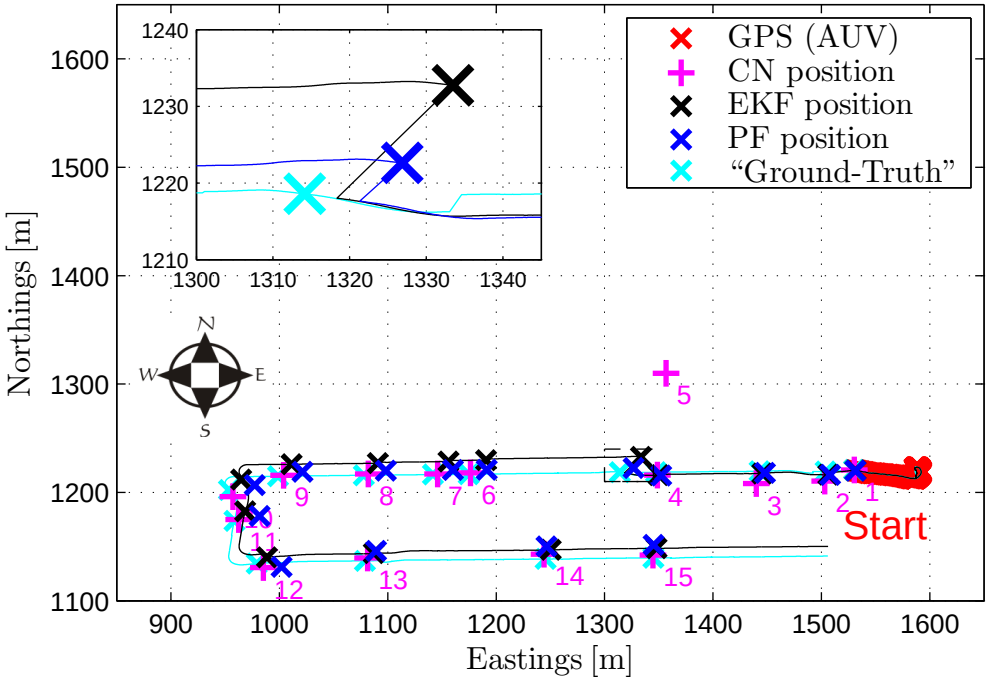


Figure 5-10: AUV/ASC mission 2 with falsified range measurement at $k = 5$: The erroneous range measurement causes a localization error for all 3 algorithms at $k = 5$, but while the CN algorithm has fully recovered at the next step, EKF and particle filter only converge slowly towards the correct solution.

solution only depends on the position error of the CNAs and the geometry. It is bounded if the position error of the CNAs is bounded and if positions which were computed from collinear or near collinear geometries are filtered out. Toward the end of the second mission, the CNAs were not able to keep up with the AUV which led to less favorable geometries resulting in slightly larger error covariances of the computed solution than in the beginning. As in the first mission, the algorithm's performance is hard to quantify. Qualitatively, the computed solutions are consistently very close to the "ground-truth" throughout the entire track while the dead-reckoned position drifts over time.

5.5 Comparison with Bayesian Estimators

In order to compare the performance of our CN algorithm with common classical approaches, an EKF and a PF with 300 particles, we computed the position using all three methods at each time instant k when a new range/position pair was available.

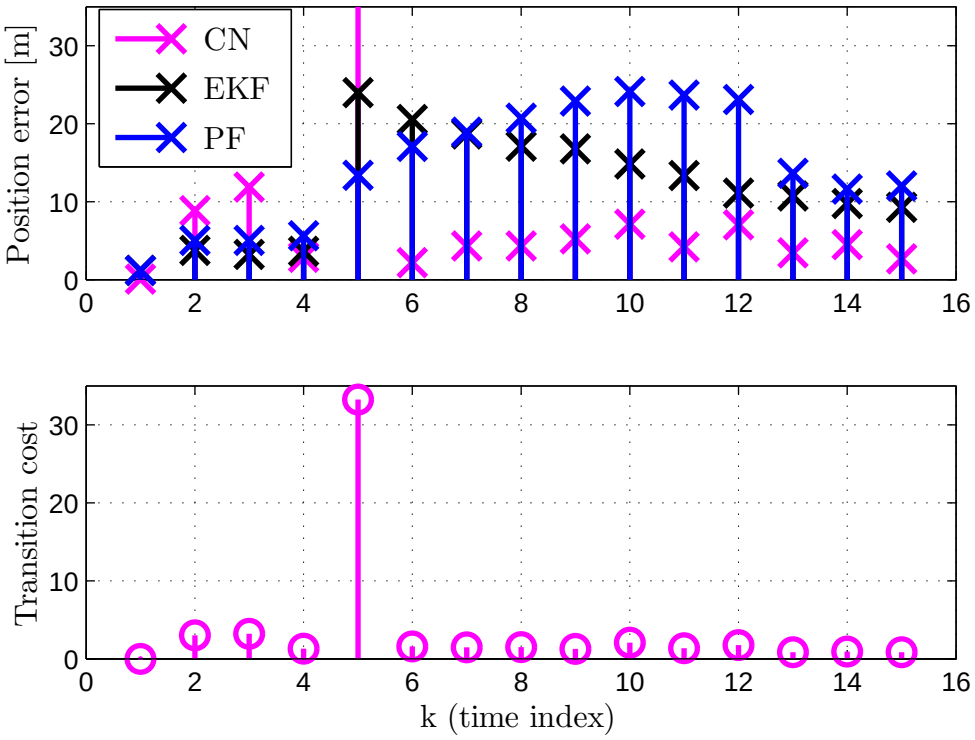


Figure 5-11: Position error for CN, EKF and PF for the tracks shown in figure 5-10: Top: position error for CN, EKF and Particle filter. Bottom: cost function for CN showing a very large peak at $k = 5$ which would enable one to filter out this inconsistent range measurement.

Because of the high quality dead-reckoning measurements and absence of range measurements outliers in the available kayak/AUV data sets, each of the three methods performed similarly and the results were within the accuracy of the ground truth.

Large underwater range measurement outliers can occur in more challenging experimental scenarios. In such a scenario the Gaussian noise assumption does not hold [66]. For this reason we simulated a typical outlier measurement by using the mission shown in figure 5-8 and setting the range measurement obtained by the AUV at $k = 5$ from $r(5)$ =116.86 m to $r(5)$ =60 m. All subsequent range measurements were unchanged. The computed tracks are shown in figure 5-10. Upon receipt of the fifth measurement the error of the position estimate "jumps" for all three methods, most significantly for the CN algorithm. However at $k = 6$ the CN algorithm instantly recovers to the correct position, while the EKF and particle filter only slowly converge towards the correct path. This is due to the very low measurement update frequency. The erroneous position produced by our CN algorithm at $k = 5$ is particularly large because our approach may only select from the solution set provided in $\boldsymbol{S}(5)$ (see appendix 3.5.3). This range measurement is however inconsistent with the previous range measurements and the dead-reckoned track and as a result has a much higher accumulated cost $\mathcal{C}(5)$, shown as a single peak in figure 5-11. Therefore

it would be possible to use $\mathcal{C}$ to detect and filter out false range measurements.

In summary an EKF is unsuitable for this application. However a more advanced particle filter with a sufficiently large number of particles could possibly provide similar performance to our proposed algorithm.

Chapter 6

Conclusion

In order to maintain a bounded position error, an individual underwater vehicle has to occasionally surface for a GPS-fix or operate within a small area surrounded by pre-deployed localization equipment. This is independent of the vehicle's size and the sophistication of its navigation sensors. Cooperative Navigation provides a framework for future deployments which cannot rely on external infrastructure and it minimizes the number of surfacings as long as two or more vehicles operate sufficiently close to each other.

6.1 Contributions

This thesis starts by examining the constraints which the underwater domain imposes on Cooperative Navigation. Its first major contribution is an algorithm which is designed to take these constraints into account. The algorithm provides a robust estimate of the vehicle's location when supplied with navigation information from cooperating platforms. It was extensively tested in a series of experiments using Autonomous Surface Crafts, propelled Autonomous Underwater Vehicles and a buoyancy driven glider. The results from these experiments show that it compares favorably against the two classical methods which are also presented in this thesis.

For a group of vehicles sharing information, two cooperation strategies can be devised. The first one does not assume any structure in the flow of navigation information. All vehicles occasionally broadcast their position estimate and incorporate messages which they receive from others. The second one assumes a hierarchy in which only a special subset of vehicles broadcasts information which all others use, thereby enforcing a uni-directional flow of information. Our algorithm, as well as the two classical approaches we present, can work in both scenarios without modification. If a vehicle however uses information from another one to update its own position, the estimates between the two vehicles become cross-correlated. This is not a problem as long as it remains a uni-directional cross-correlation. However in the case of the first scenario, in which the flow is omni-directional, information shared by a vehicle may at a later time be presented to that vehicle again as the information has been incorporated by others which now broadcast their position estimate. For all

navigation algorithms presented, this can lead to an over-confidence in the position estimate which can cause all of them to diverge.

An algorithm that ensures that the cross-correlations remain uni-directional, regardless of the direction of the information flow, and thereby prevents over-confidence, is the second major contribution of this thesis. It can be combined with any localization algorithm we present. While the algorithm requires additional information to be transmitted as well as additional computation we show how omitting this step can cause the navigation filters of all participating vehicles to diverge even in a very simple simulated scenario. We also propose an extension of the algorithm which trades off computation and bandwidth requirements against the improvement in the position estimate obtained through cooperation.

The quality of the position estimate obtained through measurements to beacons at known locations, in our case broadcasting vehicles, depends on the relative position between the beacons and the estimating vehicle. This is independent of the algorithm. The hierarchical scenario, in which the objective of the beacon vehicles is to minimize the position uncertainty of all others, we can control the geometry by actively positioning the beacon vehicles. Our third contribution is a distributed algorithm which runs on all beacon vehicles and positions them such that they jointly minimize the the uncertainty of all receiving vehicles. The algorithm does not assume a central controller and instead only relies on information which is locally available.

6.2 Future Work

Autonomous Underwater Vehicles have been successfully deployed for over a decade, but only the last few years have seen experiments involving several at a time. Even in the few experiments which involved multi-vehicle deployments, the individual members of the group were often not aware of each other and only communicated to a central control station or lacked underwater communication equipment altogether. As a result, only a very limited amount of data sets are available which can be used to test algorithms by post-processing data. Even fewer experiments have been carried out specifically with cooperation in mind, be it for navigation or any other purposes. More recently however an acoustic modem with access to a globally synchronized time signal, the key piece of equipment for the successful implementation of any CN algorithm, has become a standard feature on all new marine platforms, and many older ones have been retrofitted with this equipment. In addition, great progress was made in standardizing the communications protocol to ensure that a heterogeneous group of vehicles can successfully share information [82]. Experiments planned for the near future will offer many opportunities to verify and test the proposed algorithms in real-time. Results from these experiments will drive future research. As a navigation error can lead to the loss of a multi-million dollar vehicle, most research for cooperative navigation algorithms will focus on robustness. The emergence of new platforms and sensors not considered in this thesis will lead to new possibilities as well as constraints to which the navigation algorithms have to adapt. For the hierarchical approach which has a set of dedicated beacon vehicles there is a lot of

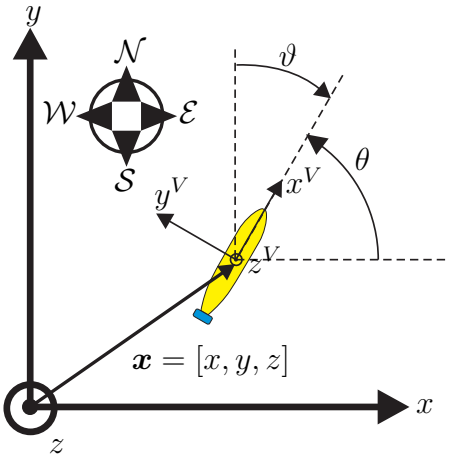
room for future research to optimize the path of these vehicles. Due to the strong variability of the acoustic communication channel, maximizing the likelihood that a transmission is successful, is a necessary additional objective. Recent advances in predicting the modem performance in real-time from in situ measurements [77] can provide additional information which is used in the path planning process.

With the necessary hardware infrastructure being available on all underwater platforms soon, a mature set of cooperative navigation algorithms will become a commodity in underwater navigation ensuring that every vehicle uses *all* available information to localize itself.

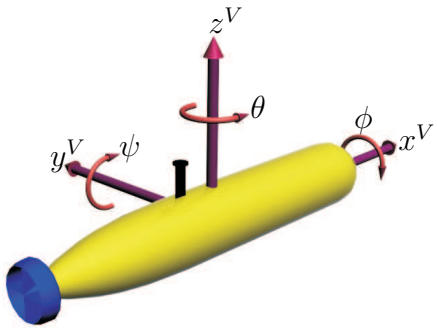
Appendix A

Coordinate Systems

A.1 Reference Frames



(a) World Coordinate System



(b) Vehicle Coordinate System

A.2 Vehicle Body Pose and Velocities

Table A.1: Vehicle Body Pose

Linear	Angular
$\boldsymbol{x} = \begin{bmatrix} x \\ y \\ z \end{bmatrix}$	ϕ (roll)
	ψ (pitch)
	θ (yaw)
	ϑ (heading) ; $\vartheta = \frac{\pi}{2} - \theta$

Table A.2: Vehicle Body Velocities

Linear		Angular
$\boldsymbol{u} = \begin{bmatrix} u \\ v \\ w \end{bmatrix}$	$u{=}\dot{x}^V{=}\frac{\partial x^V}{\partial t}$ (forward velocity)	$\dot{\phi}{=}\frac{\partial \phi}{\partial t}$ (roll rate)
	$v{=}\dot{y}^V{=}\frac{\partial y^V}{\partial t}$ (starboard velocity)	$\dot{\psi}{=}\frac{\partial \psi}{\partial t}$ (pitch rate)
	$w{=}\dot{z}^V{=}\frac{\partial z^V}{\partial t}$ (vertical velocity)	$\dot{\theta}{=}\frac{\partial \theta}{\partial t}$ (yaw rate)

Bibliography

[1] Liberdade XRay advanced underwater glider. Technical report, Office of Naval Research.

[2] P. O. Arambel, C. Rago, and R. K. Mehra. Covariance intersection algorithm for distributed spacecraft state estimation. In *Proc. American Control Conference the 2001*, volume 6, pages 4398–4403, 25–27 June 2001.

[3] R. C. Arkin and T. Balch. AuRA: Principles and practice in review. *Journal of Experimental and Theoretical Artificial Intelligence*, 9:175–189, 1997.

[4] R. A. Armstrong, H. Singh, J. Torres, R. S. Nemeth, A. Can, C. Roman, R. Eustice, L. Riggs, and G. Garcia-Moliner. Characterizing the deep insular shelf coral reef habitat of the hind bank marine conservation district (us virgin islands) using the seabed autonomous underwater vehicle. *Continental Shelf Research*, 26(2):194 – 205, 2006.

[5] P. Baccou and B. Jouvencel. Homing and navigation using one transponder for AUV, postprocessing comparisons results with long base-line navigation. In *Proc. IEEE International Conference on Robotics and Automation ICRA '02*, volume 4, pages 4004–4009, 11–15 May 2002.

[6] P. Baccou and B. Jouvencel. Simulation results, post-processing experimentations and comparisons results for navigation, homing and multiple vehicles operations with a new positioning method using on transponder. In *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003)*, volume 1, pages 811–817, 27–31 Oct. 2003.

[7] A. Bahr and J. Leonard. Minimizing trilateration errors in the presence of uncertain landmark positions. In *Proc. 3rd European Conference on Mobile Robots (ECMR)*, pages 48–53, Freiburg, Germany, September 2007.

[8] T. Balch and R. C. Arkin. Behavior-based formation control for multirobot teams. 14(6):926–939, Dec. 1998.

[9] R. Ballard, L. Stager, D. Master, D. Yoerger, D. Mindell, L. Whitcomb, H. Singh, and D. Piechota. Iron age shipwrecks in deep water off Ashkelon, Israel. *American Journal of Archaeology*, 106:151–168, 2002.

[10] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan. *Estimation with Applications to Tracking and Navigation*. Wiley Interscience, 2001.

[11] M. Benjamin, J. Curcio, J. Leonard, and P. Newman. Navigation of unmanned marine vehicles in accordance with the rules of the road. *Proc. IEEE Int. Conf. Robotics and Automation*, May 2006.

[12] M. Benjamin, M. Grund, and P. Newman. Multi-objective optimization of sensor quality with efficient marine vehicle task execution. In *Proc. IEEE International Conference on Robotics and Automation ICRA 2006*, pages 3226–3232, May 15–19, 2006.

[13] M. R. Benjamin. *Interval Programming: A Multi-Objective Optimization Model for Autonomous Vehicle Control*. PhD thesis, Brown University, Providence RI, USA, May 2002.

[14] M. R. Benjamin. The interval programming model for multi-objective decision making. Technical report, Massachusetts Institute of Technology, Cambridge, MA, USA, September 2004. 2004.

[15] M. R. Benjamin, D. Battle, D. Eickstedt, H. Schmidt, and A. Balasuriya. Autonomous control of an autonomous underwater vehicle towing a vector sensor array. In *Proc. IEEE International Conference on Robotics and Automation*, pages 4562–4569, 10–14 April 2007.

[16] D. Blidberg, S. Mupparapu, S. Chappell, R. Komerska, J. C. Jalbert, and N. R. The SAUV II (solar powered auv) test results 2004. In *Proc. Oceans 2005 - Europe*, volume 1, pages 545–550, 20–23 June 2005.

[17] Bluefin Robotics. Specifications for Bluefin21. `http://www.bluefinrobotics.com/bluefin21.htm`, December 2008.

[18] A. D. Bowen, D. R. Yoerger, C. Taylor, R. McCabe, J. Howland, D. Gomez-Ibanez, J. C. Kinsey, M. Heintz, G. McDonald, D. B. P. B. Fletcher, C. Young, J. Buescher, L. L. Whitcomb, S. C. Martin, S. E. Webster, and M. V. Jakuba. The Nereus hybrid underwater robotic vehicle for global ocean science operations to 11,000m depth. to appear, 2008.

[19] V. Caglioti, A. Citterio, and A. Fossati. Cooperative, distributed localization in multi-robot systems: a minimum-entropy approach. In *Proc. IEEE Workshop on Distributed Intelligent Systems: Collective Intelligence and Its Applications DIS 2006*, pages 25–30, 15–16 June 2006.

[20] J. Chaffee and J. Abel. GDOP and the cramer-rao bound. In *Proc. IEEE Symposium on Position Location and Navigation*, Las Vegas, NV, USA, April 1994.

[21] V. N. Christopoulos and S. Roumeliotis. Multi robot trajectory generation for single source explosion parameter estimation. In *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, pages 2803 – 2809, April 2005.

[22] D. M. Crimmins, C. T. Patty, M. A. Beliard, J. Baker, J. C. Jalbert, R. J. Komerska, S. G. Chappell, and D. R. Blidberg. Long-endurance test results of the solar-powered AUV system. In *Proc. OCEANS 2006*, pages 1–5, Sept. 2006.

[23] J. Curcio, J. Leonard, and A. Patrikalakis. SCOUT - a low cost autonomous surface platform for research in cooperative autonomy. In *Proc. MTS/IEEE OCEANS*, pages 725–729, 2005.

[24] J. A. Curcio, P. A. Mcgillivary, K. Fall, A. Maffei, K. Schwehr, B. Twiggs, C. Kitts, and P. Ballou. Self-positioning smart buoys, the "un-buoy" solution: Logistic considerations using autonomous surface craft technology and improved communications infrastructure. In *Proc. OCEANS 2006*, pages 1–5, Sept. 2006.

[25] J. Djugash, S. Singh, G. Kantor, and W. Zhang. Range-only SLAM for robots operating cooperatively with sensor networks. In *Proc. IEEE International Conference on Robotics and Automation ICRA 2006*, pages 2078–2084, May 15–19, 2006.

[26] A. Doucet, N. de Freitas, and N. Gordon, editors. *Sequential Monte Carlo methods in practice*. Springer-Verlag, 2000.

[27] A. Easton and S. Cameron. A gaussian error model for triangulation-based pose estimation using noisy landmarks. In *Proc. IEEE International Conference on Robotics, Automation and Mechatronics*, page submitted for publication, Bangkok, Thailand, June 2006.

[28] D. P. Eickstedt, M. R. Benjamin, H. Schmidt, and J. J. Leonard. Adaptive control of heterogeneous marine sensor platforms in an autonomous sensor network. In *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5514–5521, 9–15 Oct. 2006.

[29] D. P. Eickstedt and H. Schmidt. A low-frequency sonar for sensor-adaptive, multistatic, detection and classification of underwater targets with AUVs. In *Proc. OCEANS 2003*, volume 3, pages 1440–1447, 22–26 Sept. 2003.

[30] C. Eriksen, T. Osse, T. Light, R.D. Wen, et al. Seaglider: a long-range autonomous underwater vehicle for oceanographic research. *IEEE J. Ocean Engineering*, 26:424–436, October 2001.

[31] R. Eustice. *Large-Area Visually Augmented Navigation for Autonomous Underwater Vehicles*. PhD thesis, Massachusetts Institute of Technology, Cambridge, MA, USA, June 2005.

[32] M. Fallon. *Acoustic Source Tracking using Sequential Monte Carlo*. PhD thesis, University of Cambridge, June 2008.

[33] N. Farr, A. Chave, L. Freitag, J. Preisig, S. White, D. Yoerger, and F. Sonnichsen. Optical modem technology for seafloor observatories. *OCEANS 2006*, pages 1–6, Sept. 2006.

[34] P. Fearnhead. *Sequential Monte Carlo methods in filter theory*. PhD thesis, University Oxford, Oxford, UK, 1998.

[35] H. J. S. Feder, J. J. Leonard, and C. M. Smith. Adaptive mobile robot navigation and mapping. *The International Journal of Robotics Research*, 18(7):650–668, 1999.

[36] T. I. Fossen. *Guidance and Control of Ocean Vehicles*. John Wiley & Sons, New York, NY, USA, 1994.

[37] D. Fox, W. Burgard, and S. Thrun. A probabilistic approach to collaborative multi-robot localization. *Autonomous Robots*, 8(3):325–344, June 2000.

[38] U. Frese. *An O(logn) Algorithm for Simulateneous Localization and Mapping of Mobile Robots in Indoor Environments*. PhD thesis, University of Erlangen-Nürnberg, 2004.

[39] F.Schill, U.R.Zimmer, and J.Trumpf. Visible spectrum optical communication and distance sensing for underwater applications. In *Proc. Australasian Conf. Robotics and Automation*, 2004.

[40] M. E. Furlong, S. D. McPhail, and P. Stevenson. A concept design for an ultra-long-range survey class AUV. In *Proc. OCEANS 2007 - Europe*, pages 1–6, 18–21 June 2007.

[41] C. R. German, D. R. Yoerger, M. Jakuba, T. M. Shank, C. H. Langmuir, and K. Nakamura. Hydrothermal exploration with the autonomous benthic explorer. *Deep Sea Research Part I: Oceanographic Research Papers*, 55(2):203 – 219, 2008.

[42] J. Gillis. Computation of the circular error probability integral. *IEEE Trans. Aerospace and Electronic Systems*, 27(6):906–910, 1991.

[43] S. Grime and H. Durrant-Whyte. Data fusion in decentralized sensor networks. *Control engineering practice*, 2:849, 1994.

[44] O. Hegrenaes, E. Berglund, and O. Hallingstad. Model-aided inertial navigation for underwater vehicles. *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*, pages 1069–1076, May 2008.

[45] O. Hegrenaes, O. Hallingstad, and B. Jalving. A framework for obtaining steady-state maneuvering characteristics of underwater vehicles using sea-trial data. *Control & Automation, 2007. MED '07. Mediterranean Conference on*, pages 1–6, June 2007.

[46] IXSEA. Specifications of PHINS inertial navigation system. `www.ixsea.com/en/products/002.001.001.004/phins-6000.html`.

[47] A. H. Jazwinski. *Stochastic Processes and Filtering Theory*. Academic Press, New York, NY, USA, April 1970.

[48] S. J. Julier and J. K. Uhlmann. A new extension of the Kalman filter to nonlinear systems. In *In Int. Symp. Aerospace/Defense Sensing, Simul. and Controls*, pages 182–193, 1997.

[49] S. J. Julier and J. K. Uhlmann. A non-divergent estimation algorithm in the presence of unknown correlations. In *Proc. American Control Conference*, Albuquerque, New Mexico, USA, June 1997.

[50] S. J. Julier and J. K. Uhlmann. Simultaneous localisation and map building using split covariance intersection. In *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems*, volume 3, pages 1257–1262, 29 Oct.–3 Nov. 2001.

[51] R. E. Kalman. A new approach to linear filtering and prediction problems. *Transactions of the ASME - Journal of Basic Engineering*, 82(Series D):35–45, 1960.

[52] A. Kelly. Precision dilution in triangulation based mobile robot position estimation. In *Proc. of the 8th Int. Conf. on Intelligent Autonomous Systems*, page submitted for publication, Amsterdam, Netherlands, June 2003.

[53] J. Kinsey and L. Whitcomb. Model-based nonlinear observers for underwater vehicle navigation: Theory and preliminary experiments. *Robotics and Automation, 2007 IEEE International Conference on*, pages 4251–4256, April 2007.

[54] R. Kurazume, S. Hirose, and S. Nagata. Study on cooperative positioning system. In *Proc. IEEE Int. Conf. Robotics and Automation*, pages 1421–1426, Minneapolis, MN, USA, April 1996.

[55] M. Kyo, E. Hiyazaki, S. Tsukioka, H. Ochi, Y. Amitani, T. Tsuchiya, T. Aoki, and S. Takagawa. The sea trial of KAIKO the full ocean depth research ROV. In *Proc. MTS/IEEE. Challenges of Our Changing Global Environment OCEANS '95*, volume 3, pages 1991–1996, 9–12 Oct. 1995.

[56] M. B. Larsen. Synthetic long baseline navigation of underwater vehicles. In *Proc. OCEANS 2000 MTS/IEEE Conference and Exhibition*, volume 3, pages 2043–2050, 11–14 Sept. 2000.

[57] N. Leonard, D. Paley, F. Lekien, R. Sepulchre, D. Fratantoni, and R. Davis. Collective motion, sensor networks and ocean sampling. *Proceedings of the IEEE*, 95(1):48–74, January 2007. Special issue on the emerging technology of networked control systems.

[58] J. B. McKay and M. Pachter. Geometry optimization for GPS navigation. In *Proc. IEEE Conference on Decision and Control*, San Diego, CA, USA, December 1997.

[59] R. Merwe and E. Wan. Gaussian mixture sigma-point particle filters for sequential probabilistic inference in dynamic state-space models. In *In Proceedings of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP), Hong Kong*, pages 701–704. IEEE, 2003.

[60] D. Moore, J. Leonard, D. Rus, and S. Teller. Robust distributed network localization with noisy range measurements. In *SenSys '04: Proceedings of the 2nd international conference on Embedded networked sensor systems*, pages 50–61, New York, NY, USA, November 2004. ACM Press.

[61] U. S. Navy. Extremely low frequency transmitter site clam lake, wisconsin. `enterprise.spawar.navy.mil/UploadedFiles/fs_clam_lake_elf2003.pdf`.

[62] E. Nettleton and H. F. Durrant-Whyte. Delayed and asequent data in decentralized sensing networks. In G. T. McKee and P. S. Schenker, editors, *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, volume 4571 of *Presented at the Society of Photo-Optical Instrumentation Engineers (SPIE) Conference*, pages 1–9, Oct. 2001.

[63] E. Nevala. A glide across the gulf stream. *Oceanus*, 44:2, 2005.

[64] P. Newman. *MOOS - Mission Orientated Operating Suite*. Oxford University Robotics Research Group, Department of Engineering Science, Oxford University, Oxford, UK, 2002.

[65] E. Olson. *Robust and Efficient Robotic Mapping*. PhD thesis, Massachusetts Institute of Technology, Cambridge, MA, USA, June 2008.

[66] E. Olson, J. Leonard, and S. Teller. Robust range-only beacon localization. In *Proc. IEEE/OES Autonomous Underwater Vehicles*, pages 66–75, 17–18 June 2004.

[67] E. Olson, J. J. Leonard, and S. Teller. Robust range-only beacon localization. *IEEE Journal of Oceanic Engineering*, 31(4):949–958, Oct. 2006.

[68] D. A. Paley. *Cooperative Control of Collective Motion for Ocean Sampling with Autonomous Vehicles*. PhD thesis, Princeton, Princeton, NJ, USA, September 2007.

[69] D. A. Paley, F. Zhang, and N. E. Leonard. Cooperative control for ocean sampling: The glider coordinated control system. 16(4):735–744, July 2008.

[70] J. Refsnes, A. Sorensen, and K. Pettersen. Model-based output feedback control of slender-body underactuated AUVs: Theory and experiments. *Control Systems Technology, IEEE Transactions on*, 16(5):930–946, Sept. 2008.

[71] C. Roman and H. Singh. Micro-bathymetric mapping using acoustic range images. In *Proc. OCEANS '04. MTTS/IEEE TECHNO-OCEAN '04*, volume 3, pages 1574–1579, 2004.

[72] S. Roumeliotis and G. Bekey. Synergetic localization for groups of mobile robots. In *Proc. 39th IEEE Conference on Decision and Control*, pages 3477–3482, Sydney, Australia, December 2000.

[73] S. Roumeliotis and G. Bekey. Distributed multirobot localization. *Robotics and Automation, IEEE Transactions on Robotics and Automation*, 18(5):781–795, Oct 2002.

[74] S. I. Roumeliotis and I. Rekleitis. Analysis of multirobot localization uncertainty propagation. In *Proc. IEEE Int. Workshop on Intelligent Robots and Systems*, pages 1763–1770, Las Vegas, NV, USA, October 2003.

[75] S. I. Roumeliotis and I. M. Rekleitis. Propagation of uncertainty in cooperative multirobot localization: Analysis and experimental results. *Autonomous Robots*, 17(1):1573–7527, July 2004.

[76] A. P. Scherbatyuk. The AUV positioning using ranges from one transponder lbl. In *Proc. MTS/IEEE. Challenges of Our Changing Global Environment OCEANS '95*, volume 3, pages 1620–1623, 9–12 Oct. 1995.

[77] T. Schneider. Improving acoustic communications in the littoral zone through dynamic encoding and adaptive vehicle motion. to appear, 2008.

[78] D. Shnidman. Efficient computation of the circular error probability (cep) integral. *IEEE J. Robotics and Automation*, 40(8):1472–1474, 1995.

[79] H. Singh, A. Can, R. Eustice, S. Lerner, N. McPhee, O. Pizarro, and C. Roman. Seabed auv offers new platform for high-resolution imaging. *EOS, Transactions of the American Geophysical Union*, 85(31):289,294–295, August 2004.

[80] H. Singh, J. Howland, and O. Pizarro. Advances in large-area photomosaicking underwater. 29(3):872–886, July 2004.

[81] H. Sorenson, editor. *Kalman Filtering: theory and application.* IEEE Press, 1985.

[82] R. Stokey, L. Freitag, and M. Grund. A compact control language for AUV acoustic communication. *Oceans 2005 - Europe*, 2:1133–1137 Vol. 2, June 2005.

[83] H. Thomas. Commercial offer: GPS intelligent buoy system (A.C.S.A., `http://www.underwater-gps.com`), 2001.

[84] S. Thrun, W. Burgard, and D. Fox. *Probabilistic Robotics.* The MIT Press, Cambridge, MA, USA, 2005.

[85] D. Torrieri. Statistical theory of passive location systems. *IEEE Trans. Aerospace and Electronic Systems*, 20(2):183–197, 1984.

[86] N. Trawny and T. Barfoot. Optimized motion strategies for cooperative localization of mobile robots. In *Proc. IEEE International Conference on Robotics and Automation ICRA '04*, volume 1, pages 1027–1032, 2004.

[87] J. Vaganay, P. Baccou, and B. Jouvencel. Homing by acoustic ranging to a single beacon. In *Proc. OCEANS 2000 MTS/IEEE Conference and Exhibition*, volume 2, pages 1457–1462, 11–14 Sept. 2000.

[88] J. Vaganay, J. Leonard, J. Curcio, and J. Willcox. Experimental validation of the moving long base-line navigation concept. *Autonomous Underwater Vehicles, 2004 IEEE/OES*, pages 59–65, June 2004.

[89] I. Vasilescu, K. Kotay, D. Rus, M. Dunbabin, and P. Corke. Data collection, storage, and retrieval with an underwater sensor network. In *SenSys '05: Proceedings of the 3rd international conference on Embedded networked sensor systems*, pages 154–165, San Diego, CA, USA, 2005. ACM.

[90] M. Walter. *Sparse Bayesian Information Filters for Localization and Mapping*. PhD thesis, Massachusetts Institute of Technology, Cambridge, MA, USA, February 2008.

[91] M. Walter, F. Hover, and J. Leonard. SLAM for ship hull inspection using exactly sparse extended information filters. In *Proc. IEEE International Conference on Robotics and Automation ICRA 2008*, pages 1463–1470, 19–23 May 2008.

[92] WFS. Through-water and through-ground broadband data link - s5510. http://wirelessfibre.co.uk/datasheets.html.

[93] L. Whitcomb, D. Yoerger, and H. Singh. Combined Doppler/LBL based navigation of underwater vehicles. In *Proceedings of the International Symposium on Unmanned Untethered Submersible Technology (UUST)*, Durham, NH, USA, May 1999.

[94] O. Wijk and H. Christensen. Triangulation based fusion of sonar data with application in robot pose tracking. *IEEE Trans. Robotics and Automation*, 16(6):740–752, December 2000.

[95] S. Winitzki. A handy approximation for the error function and its inverse. lecture note, 2006.

[96] K. Zhou and S. Roumeliotis. Optimal motion strategies for range-only distributed target tracking. *American Control Conference, 2006*, pages 6 pp.–, June 2006.

50272-101

REPORT DOCUMENTATION PAGE	1. REPORT NO. MIT/WHOI 2009-05	2.	3. Recipient's Accession No.

4. Title and Subtitle

Cooperative Localization for Autonomous Underwater Vehicles

5. Report Date

February 2009

6.

7. Author(s) Alexander Bahr

8. Performing Organization Rept. No.

9. Performing Organization Name and Address

MIT/WHOI Joint Program in Oceanography/Applied Ocean Science & Engineering

10. Project/Task/Work Unit No.

MIT/WHOI 2009-05

11. Contract(C) or Grant(G) No.

(C) N00014-97-1-0202
N00014-05-1-0255

(G) N00014-02-C-0210
N00014-07-1-1102

12. Sponsoring Organization Name and Address

Office of Naval Research
WHOI Academic Programs Office

13. Type of Report & Period Covered

Ph.D. Thesis

14.

15. Supplementary Notes

This thesis should be cited as: Alexander Bahr, 2009. Cooperative Localization for Autonomous Underwater Vehicles. Ph.D. Thesis. MIT/WHOI, 2009-05.

16. Abstract (Limit: 200 words)

Self-localization of an underwater vehicle is particularly challenging due to the absence of GPS reception or features at known positions that could be used for position computation. This typically requires the pre-deployment of a set of beacons.

This thesis examines the scenario in which the members of a group of AUVs exchange navigation information with one another so as to improve their individual position estimates. We describe how the underwater environment poses unique challenges to vehicle navigation and how cooperation can improve the performance of self-localization. We also address the constraints of the communication channel and the effect that these constraints have on the design of cooperation strategies.

The classical approaches to underwater self-localization of a single vehicle are presented. We then examine how methods used for cooperating land-vehicles can be transferred to the underwater domain. An algorithm for distributed self-localization is proposed. We also address how correlated position estimates of cooperating vehicles can lead to overconfidence in individual position estimates.

Finally, the key to any successful cooperative navigation strategy is the incorporation of the relative positioning between vehicles. A distributed algorithm for the dynamic positioning of vehicles is proposed.

17. Document Analysis **a. Descriptors**

AUV
underwater navigation
multi-vehicle cooperation

b. Identifiers/Open-Ended Terms

c. COSATI Field/Group

18. Availability Statement

Approved for publication; distribution unlimited.

19. Security Class (This Report) UNCLASSIFIED

20. Security Class (This Page)

21. No. of Pages 140

22. Price

(See ANSI-Z39.18) *See Instructions on Reverse* **OPTIONAL FORM 272 (4-77)**
(Formerly NTIS-35)
Department of Commerce