

Thesis to get the degree of a doctor of philosophy

# **Localization of a Swarm of Underwater Robots Using Set-Membership Methods**

Mohamed Saad IBN SEDDIK

Date : November 24, 2015

Replace this box e. g. by  
the coat of arms of the  
university.

ENSTA Bretagne  
Lab-STICC  
Ocean Sensing and Mapping Laboratory

**Supervisors:**

Prof. Luc JAULIN

Mr. Jonathan GRIMSDALE

**Referees:**

Pr. Eric GOUBAULT

Dr. Micheal R. BENJAMIN

**Date of the graduation (optional)**

xx.yy.zzzz

*To My Parents and Sister*



# Contents

<b>Nomenclature</b>	<b>1</b>
<b>1 Introduction</b>	<b>3</b>
1.1 Unmanned Marine Vehicles . . . . .	4
1.1.1 Platforms . . . . .	4
1.1.2 Applications . . . . .	7
1.1.3 Levels of Autonomy . . . . .	10
1.2 Context: Oil & Gas Exploration . . . . .	11
1.2.1 Oil & Gas Exploration Industry . . . . .	11
1.2.2 Seismic Surveys . . . . .	12
1.3 Hypothesis, Constraints And Objectives . . . . .	15
1.3.1 Hypothesis and Constraints . . . . .	15
1.3.2 Objectives . . . . .	15
1.4 Thesis Outline . . . . .	16
<b>2 Localization For Underwater Vehicles</b>	<b>17</b>
2.1 Introduction . . . . .	17
2.2 Limitations to Underwater Vehicles . . . . .	17
2.2.1 Communications . . . . .	17
2.2.2 Energy . . . . .	19
2.2.3 Localization & Navigation . . . . .	20
2.3 Conclusion . . . . .	24
<b>3 Set-Membership Tools</b>	<b>25</b>
3.1 Introduction . . . . .	25
3.2 Set-Membership Approach . . . . .	27
3.3 Interval Analysis . . . . .	28
3.3.1 Real Intervals . . . . .	29
3.3.2 Boxes . . . . .	30
3.3.3 Interval Arithmetic . . . . .	30
3.3.4 Tubes . . . . .	32
3.4 Contractors . . . . .	33
3.4.1 Definitions and Properties . . . . .	33
3.4.2 Constraint Satisfaction Problem . . . . .	34
3.4.3 Forward-Backward Contractor . . . . .	35

3.5	Set Inversion Via Interval Analysis . . . . .	40
3.5.1	Sub-Pavings . . . . .	40
3.5.2	Inclusion Function . . . . .	41
3.5.3	SIVIA . . . . .	42
3.5.4	SIVIA With Contractors . . . . .	44
3.6	Conclusion . . . . .	48
<b>4</b>	<b>Phase Based Localization</b>	<b>49</b>
4.1	Introduction . . . . .	49
4.2	Motivation . . . . .	50
4.2.1	Scalability . . . . .	50
4.2.2	Minimum Communication . . . . .	50
4.2.3	Kidnapping . . . . .	50
4.2.4	Unknown Starting Conditions . . . . .	51
4.2.5	Path Reconstruction . . . . .	51
4.3	Problem Statement . . . . .	51
4.3.1	Motion Model . . . . .	51
4.3.2	Phase-Lock Algorithms . . . . .	52
4.4	Algorithm Description . . . . .	53
4.4.1	Casting the problem into a CSP . . . . .	53
4.4.2	Algorithm . . . . .	55
4.5	Simulation & Implementation . . . . .	56
4.5.1	Almost-Perfect Environment . . . . .	56
4.5.2	Real-World Conditioned Environment . . . . .	58
4.6	Conclusion . . . . .	71
<b>5</b>	<b>Collaborative Localization</b>	<b>73</b>
5.1	Introduction . . . . .	73
5.2	Problem Statement & Interval State Estimator . . . . .	74
5.2.1	Problem Statement . . . . .	74
5.2.2	Interval State Estimator . . . . .	74
5.3	Polar Contractor . . . . .	76
5.4	Cooperative Localization . . . . .	81
5.4.1	Range & Bearing Measurements . . . . .	81
5.4.2	Bearing Only . . . . .	83
5.4.3	Range Only . . . . .	83
5.5	Conclusion . . . . .	87
<b>6</b>	<b>Conclusion</b>	<b>89</b>
	<b>Bibliography</b>	<b>91</b>

# Nomenclature

AHRS	Attitude and Heading Reference System
AIS	Automatic Identification System
AUV	Autonomous Underwater Vehicle
COB	Center Of Buoyancy
COM	Center Of Mass
CSP	Constraint Satisfaction Problem
DVL	Doppler Velocity Log
EKF	Extended Kalman Filter
INS	Inertial Navigation System
LBL	Long BaseLine
PLL	Phase-Lock Loop
SIVIA	Set Inversion Via Interval Analysis
TCAS	Traffic Collision Avoidance System
TDMA	Time-Division Multiple-Access
ToF	Time-of-Flight
TRL	Technology Readiness Level
USBL	Ultra-Short BaseLine
UUV	Unmanned Underwater Vehicle



# 1 Introduction

Numerous robots made the headlines in the recent years as what the future could look like. From the affordability of robot vacuum cleaners to the democratization of aerial photography with user-friendly *drones*, both consumers and industries see a huge potential of robots to perform tasks that are either dangerous or just unattractive to human-beings.

Industrial robots are now standard in many manufacturing industries and considered the only option to keep up with the demand and the target cost. On the other hand, mobile robots have mainly been used in closed environments in laboratories and only started seeing sun-light in the last decade.

Arguably, the two fields that motivate the most mobile robotics are space and underwater exploration. Both share similar constraints for humans that difficulty can be overcome such as pressure, energy, and communications. Therefore, each can benefit from the advances of the other field to tackle its problems. Note that composite materials, more efficient battery technologies and autonomy are usually heavily tested and developed for one of these two domains.

Moreover, with the growing interest of the offshore industry to commercially use underwater robotics in two key activities: surveying and offshore engineering, the field is about to see a significant growth. Earlier development of Autonomous Underwater Vehicles (AUV) has mainly been funded for hydrographic surveying and charting, for navigation for military and defense, and for environmental assessments and monitoring and for national economic exploitation. In 2001, about 52% of all marine survey efforts were concentrated on these activities [1]. In the same year, the oil and gas industry accounted for more than 58% of all the offshore industry. And at that time, only 17% of its activities were deep-water and it has grown constantly since then.

Chance [2] illustrated the cost-effectiveness of AUV operations in contrast with towed systems. The main arguments to replace this technology with AUVs are:

- *Line running direction*: Towed systems cannot easily accommodate sharp changes in direction whereas an AUV can.
- *Line running position*: Keeping a deeply towed system on line is difficult without significant additional cost. AUVs on the other hand run in line.
- *Line running altitude/aspect ration*: Keeping a towed system at correct depth is not easy. An AUV can fly at the correct depth or height from the seabed precisely.

- *End of Line turns*: Towed systems require a long line turn to not get tangled. An AUV can turn on the spot.
- *Run-ins/run-outs*: Towed systems require a run-in to a line and a run-out, AUVs need neither a run-in or run-out.
- *Positioning*: In deep water, positioning of towed system is problematic and, if any degree of accuracy is needed, calls for a second positioning system. AUVs can run autonomously or require only one positioning system.
- *Survey speed*: Survey velocity of a towed system is a function of water depth and cable length, whereas AUVs can travel at same speed irrespective of water depth.

In this chapter, we first introduce the different types of unmanned marine vehicles commonly available. Then we describe the context of Oil and Gas exploration for which this thesis is being done. The hypothesis, constraints and objectives of this work are presented before giving the outline of the thesis.

## 1.1 Unmanned Marine Vehicles

After the unfortunate events of *Malaysian Airlines MH370* that disappeared from the radars while flying from Kuala Lumpur Airport to Beijing, the media closely followed the search for the missing aircraft shining light on the Autonomous Underwater Vehicles (AUVs) used as a key component of the mission, Figure 1.1. As explained earlier, the advantage of using AUVs over towed sonars for instance reflects mainly on the cost and the duration of the mission.

In this section, provided is a list of different Unmanned Marine Vehicles in general and their characteristics. We will then illustrate their applications in different fields and also detail the different levels of autonomy usually considered in this type of vehicles.

### 1.1.1 Platforms

#### 1.1.1.1 Passively Controlled Vehicles

For the sake of energy-efficiency, long-endurance, and range, many Unmanned Marine Vehicles rely essentially on the energy provided by the ocean currents and/or the lift on the “*wings*” due to the difference of buoyancy.

**Buoys** On one hand, buoys passively drift with the ocean current for a period of time while recording data and their position when possible. They either stay on the surface, a weather buoy for instance 1.2a, or maintain a constant depth during the



**Figure 1.1:** The U.S. Navy AUV *Artemis* (Bluefin 21') deployed from the *ADV Ocean Shield* vessel to search for the missing *MH370* at a 4000 meters depth using a side-scan sonar.

mission. Furthermore, the speed and the path of the vehicle totally depend on the currents making the prediction of the survey's output and duration a little random. Therefore, the buoys are only used in non-critical environmental surveys or to learn about currents in some remote areas.

**Gliders / Wave Gliders** On the other hand, gliders and wave gliders rely on wings, to produce lift and therefore generate forward movement. Gliders main propulsion is based on internal pumps that change the vehicles displaced volume by pumping liquid to an inside bladder. In some cases, a small motor is used to change the position of the Center Of Mass (COM) relatively to the Center Of Buoyancy (COB) to change the pitch of the vehicle. This dynamic change of the characteristics of the vehicle, buoyancy and positions of COB and COM, make it perform a sawtooth pattern from the surface to several thousands of meters. The *Seaglider* for instance [3], 1.2b, only consumes 0.5W in average over the entire period of a mission. This is due to the fact that the pumps are only activated on the surface and when near to the bottom. Therefore, this particular glider can travel for several thousands of kilometers before needing to recharge. Nevertheless, its speed is very limited (0.25-0.5 m/s) and may result on the vehicle drifting to an undesired location in the presence of strong currents.

Most of the missions based on these systems require a vessel to follow the vehicle to recover it and its data or a satellite data link when the vehicle is disposable. In the case where the vehicle has to be recovered, the vessel might have to travel a considerable distance to reach it as the path of these systems is not very predictable,



(a) A Weather Buoy.



(b) A Seaglider Glider Diving.



(c) A LiquidRobotics Wave Glider.

**Figure 1.2:** Passively Controlled Marine Vehicles.

therefore limiting their use.

### 1.1.1.2 Dynamically Controlled Vehicles

In contrast with the previous category, dynamically controlled vehicles can control their position or other variables actively to reach a desired state.

**Surface Vehicles** Repurposed boats or scaled-down versions have been modified to accomplish long duration missions or surveys where limited human interaction is needed. With recent advances in artificial intelligence and autonomy [4, 5], some even can interact with the environment and traffic. Because these vehicles are always on the surface, access to position information (through GPS or other positioning systems) and communication (AIS and TCAS for example) allow a much higher level of autonomy and control. Figure 1.3 shows different autonomous surface vehicles that have been developed in the recent years for different purposes.

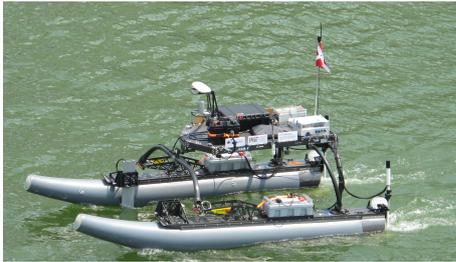
**Underwater Vehicles** Underwater, the most common types of vehicles are ROV (Remotely Operated Vehicles) and AUV (Autonomous Underwater Vehicles). ROVs,



(a) Ifremer's VAIMOS is an autonomous sailboat developed in collaboration with ENSTA Bretagne.



(b) NATO CMRE Autonomous Surface Vehicle.



(c) MIT-Olin's WAM-V.



(d) Remotely Operated Surveillance Jet-ski.

**Figure 1.3:** Autonomous Surface Vehicles

1.4d, in one hand, are very agile and can perform dexterous tasks with ease. Nevertheless, they require a pilot, or even two, to carry on the missions. Their tasks range from sea-bottom imaging to manipulation of underwater pipeline and valves. Therefore their shape is usually not very hydrodynamic and depends on an umbilical cable and a surface ship and usually move very slowly. AUVs, on the other hand, are mostly torpedo shaped to minimize the drag and consequently require a relatively low-power propulsion. Dimensions of AUV models vary strongly between less than a meter up to 10 meters in length, 1.4a, and diameter from a couple dozen centimeters, 1.4b, to meters. Dimensions mainly depend on the mission and its duration as batteries and sensors take most of the volume. Early AUVs were usually large, but with the recent advances in battery technologies and miniaturization the tendency is more towards smaller human-portable size that are customizable with modules depending on the nature of the mission. Nonetheless, large AUVs are still the main go-to solution for long duration missions at high depth.

### 1.1.2 Applications

Autonomous Marine Vehicles are used in a wide-spread range of applications. Most applications fall under one of the three main fields where marine robots are used:

- defense,



(a) Hydroid's REMUS AUV developed in collaboration.



(b) GraalTech's eFolaga AUV.



(c) ENSTA Bretagne hand-made AUV.



(d) ROV in deployment.

**Figure 1.4:** Autonomous Surface Vehicles

- oceanography,
- and marine industries.

These domains employ marine vehicles in different missions with different purposes. In this subsection is detailed the most known use cases of these robots.

### 1.1.2.1 Defense

The defense being one of the very first users of these technologies have been driving the development of these vehicles thanks to the important investments in research and development. Moreover, the hazardous nature of the environment for the human life and the equipment was a crucial driver for such an improvement.

The advantage of autonomous systems in the battle space range from the ability to collect tactical, operational, and strategic intelligence data for the commandment. Furthermore, their relatively small footprint and signatures, ease of clandestine deployment, and autonomous operation have noticeable advantage in contrast with other human driven solutions [6].

The US Navy has identified 40 distinct missions for Unmanned Underwater Vehicles (UUV) [7, 8] and prioritized them according to their usefulness in the field. The top five missions as listed below, all are feasible without UUVs. Nevertheless, the

use of the later will unquestionably improve the outcome of the battle of the navy equipped with them:

- Intelligence, surveillance and reconnaissance,
- Mine countermeasures,
- Antisubmarine Warfare,
- Battle space characterization,
- Inspection & identification.

Moreover, the equipment and embedded sensors needed for each mission are not necessarily required for another. Therefore, the modularity of these vehicles became a requirement to give the user a blank page so he can adapt the vehicle to his use as needed.

### 1.1.2.2 Oceanographic Research

Despite the fact that oceanographic missions are far less dangerous, they set the benchmark for several UUVs. Many UUVs have been first used for oceanographic purposes. In contrast with the defense type of missions that are relatively short and mostly near the littoral, oceanographic research requires the vehicles to stay longer and deeper.

For instance, when observing the oceans role in climate, marine robots can provide persistent and systematic measurements. Gliders for example are well suited to cover the required 200 km (and larger) spatial scales and seven days (and longer) timescales. UUVs can also observe deep and intermediate water masses in all weather and sea conditions. They can be equipped to determine the along-current extent of circulation patterns that are not readily apparent from the ship-collected data and are sometimes too remote for ship monitoring. Moreover, even coastal observations done using marine robots can present an advantage when observing multiple indicators at the same time. As robots yield more data per unit cost than ships and other moored systems, they had a notorious impact on understanding ocean processes.

### 1.1.2.3 Industrial Applications

Up to the recent years, the technology readiness level (TRL) of marine robots was not yet considered reliable for industrial use. Actually, because of the regulation around the use of military developed robots and research stage of those used in oceanography, such technology was often judged to be too young for large industrial usage.

Luckily, marine robots manufacturers did see the potential of the market for such vehicles. It then lead to the rise of a new category of robots that are cheaper than their military counterparts but also more reliable than those used in the research.

Oil and Gas industry, alongside the telecommunication industry, lead the way by using ROVs to survey areas and the underwater structures and infrastructures. However, only few projects are known to use on-board autonomy.

### 1.1.3 Levels of Autonomy

Autonomy for robotic systems does come in different levels, from very low-level teleoperation to fully intelligent autonomous aware robots.

The Standard Guide on UUV Autonomy and Control [9] considers a three dimensional space for assessing the autonomy requirements for a mission: (1) situation awareness; (2) decision-making, planning, and control; (3) external interaction. A summary of the definition of the levels of autonomy and capabilities of each level is provided in the tables below.

UUV (1) situation awareness levels		
0	Raw	Unprocessed data collected from sensors
1	Semi-Processed	
2	Feature	Data filtered, normalized, and characterized
3	Aggregate	Data aggregated over time, space, and/or feature characteristics (target detection)
4	Interpreted	Compared against some knowledge database
5	Inferred	Correlated to knowledge base of situational or behavioral characteristics
6	Intent	Correlated to a knowledge base of known patterns of operation including recognizing plans and objectives and recognizing and/or predicting intent

UUV (2) decision making, planning, and control levels of autonomy		
0	Direct control	Execution of externally defined commands; no decision-making
1	Sequenced execution	Execution of externally defined sequence of actions; limited decision-making
2	Adaptive execution	Execution of externally defined actions; closes loop on the actions; limited decision-making
3	Objective achievement	Combine actions to achieve a single objective at a time
4	Multiple objective	Simplistic combining of multiple objectives
5	Joint objective achievement	Balance multiple objectives

UUV (3) external levels of autonomy		
0	Teleoperation	Operator uses continuous video and sensor feedback to directly control the UUV
1	Remote control	Operator provides frequent control, but UUV can execute some actions
2	Semi-autonomous control	UUV chooses from a priori list of actions to execute to achieve the operator's goal; capable of continuing some operations during communications blackouts; operator can override and redirect actions
3	Fully autonomous	UUV functions without operator intervention from launch until recovery; if within range, underway communication and redirection is still possible from the operator

Depending on the use case of the UUV, it is sometimes preferred to have fully autonomous operations for a routine non-dangerous mission. And in the contrary, a human control over the vehicle is favored in defense situations not to cause any collateral damage.

For the purpose of the targeted application of this thesis, i.e. Oil & Gas exploration, the UUVs should be fully autonomous with multi-objective level of decision making [10]. However, due to cost related issue, the situation awareness level will be kept at aggregating data.

## 1.2 Context: Oil & Gas Exploration

### 1.2.1 Oil & Gas Exploration Industry

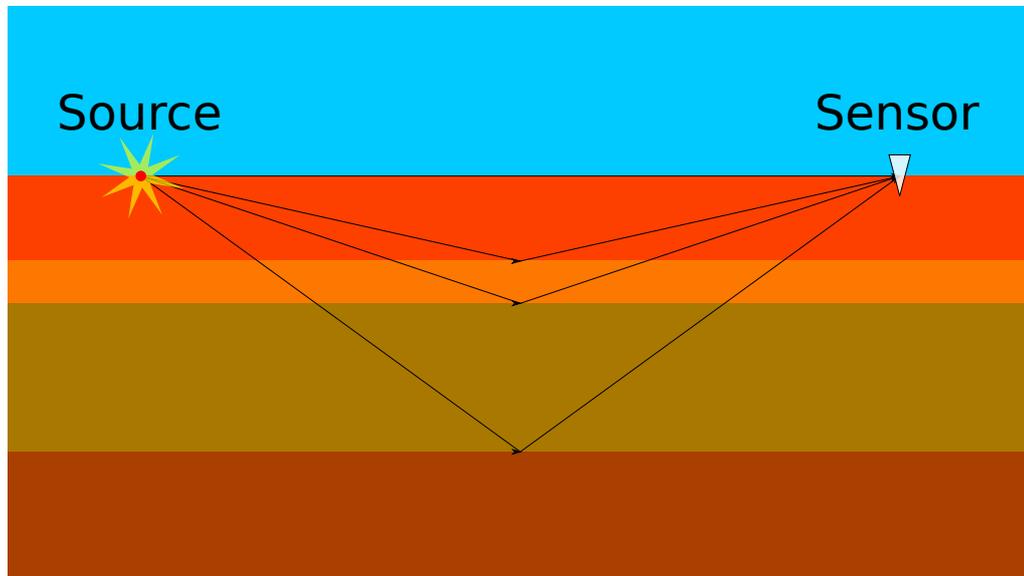
The Oil & Gas exploration industry is a multi-billion industry that is only a small part of the Oil & Gas industry. Its purpose is to search and monitor underground reservoirs to better extract the hydrocarbons.

Characterizing this subsurface deposits require highly sophisticated technology to detect and determine their extent. This process is called imaging reservoirs. To do so, surveys are performed on the area of interest. Thanks to the different composition of hydrocarbons in contrast with other rocks and sediments, it is possible to infer the depth and size of the different layers and then pinpointing the best position where to extract.

Techniques to image the underground vary from gravimetry, magnetometry, and seismic. Gravimetry and magnetometry are passive ways of detecting hydrocarbons as these parameters are affected by the existence of underground reservoirs. Seismic, in the other hand, can either be passive: by recording the seismic waves generated by an earthquake; or active by generating a relatively small seismic wave either using explosives or seismic wave generators, Figure 1.5. Because of the context of this

thesis, gravimetry and magnetometry will not be discussed, only seismic surveys are detailed next.

## 1.2.2 Seismic Surveys



**Figure 1.5:** Seismic Survey Is About Generating A Seismic Wave That Will Reflected By The Different Layers Forming The Underground.

Conducting a seismic survey of an area requires a tremendous amount of equipment and personnel. It is mainly due to the complexity of generating the seismic wave at low frequencies. Actually, the frequencies of interest for the geophysicists to study the geological composition of the area, range from few Hertz to several kilo-Hertz [11].

The principle of a seismic survey is that a source generates a seismic wave at a predefined position. By reflecting and diffracting along the different layers, the seismic wave propagates downwards then upwards at different speeds depending on the density and elasticity of the material. The reflected wave is then recorded using the adapted sensors, Figure 1.6. By computing the time travel between the generated wave and the recorded one, and by multiplying the number of sources and receivers, it is possible to conclude the velocity profile and the depth of each layer; this process is called inversion [12].

### 1.2.2.1 Land Acquisition

When the recorders are coupled to the earth surface, whether onshore or at the sea floor, it is considered to be a land seismic acquisition. As the sensors are in



**Figure 1.6:** Seismic Sensors.

direct contact with the earth, they can better record its tiniest movements. Three geophones (extremely sensitive accelerometers) per sensor are usually used in this situation to register the earth movement in the three dimensions.

Onshore, the seismic waves are often generated using a vibrating heavy mass in contact with the surface and carried by a truck, Figure 1.7. Offshore, very high pressure bubbles bursting inside the water generate the seismic wave. It is achieved by filling a closed underwater cylinder with pressurized air or water vapor, then opening the cylinder suddenly. The shock wave generated from the bursting bubbles is equivalent to dynamite that was used in early surveys. Nowadays, dynamite is only used in land missions where trucks cannot reach the desired position, mountains for instance.

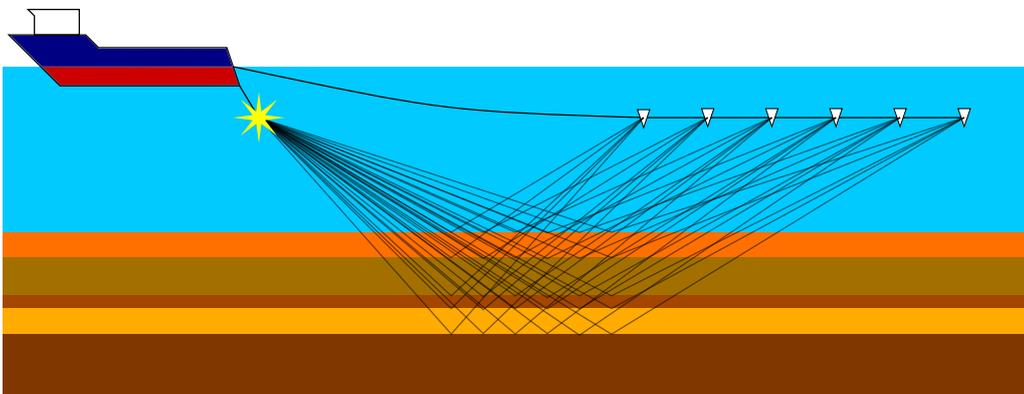
### **1.2.2.2 Marine Acquisition**

Marine acquisition is not different from the land acquisition in the way of generating the seismic wave through bursting bubbles. However, the sensors are not placed on the sea floor. They are organized in long cables dragged behind a moving vessel. These cables, often called streamers, can be several kilometers long. And a vessel can drag as much as a dozen cables, making the marine seismic surveys the largest man-made moving objects in the worlds.



**Figure 1.7:** A land seismic vibrator: Nomad 90.

The main difference between a marine survey and a land survey is that in marine surveys, the receivers are moving constantly, Figure 1.8. Therefore, to have a correct image of the underground, it is crucial to keep track of the streamers position at all time. The seismic vessel cannot simply stop risking its streamers to get tangled and sunk. Also because of the dimensions of the streamer, it generate several tons of drag, causing the vessel to be less efficient. Furthermore, an hour of marine operations is estimated to cost tens of thousands of dollars.



**Figure 1.8:** Marine Acquisition.

Motivated by the efficiency, the cost effectiveness, and the flexibility that UUVs could provide to marine seismic surveys in contrast with the traditional streamer based surveys, this thesis explores the possibility of such a system. This thesis does not look into the design of the UUV itself, but into localization algorithms and techniques adapted for the large number of required vehicles given the restrained

underwater environment.

### **1.3 Hypothesis, Constraints And Objectives**

#### **1.3.1 Hypothesis and Constraints**

The previous section presented the context in which this thesis is done. In a marine environment, where huge vessels are used to conduct seismic survey, we would like to optimize the efficiency and cost effectiveness of the mission by replacing the streamers as a carrier for the sensors with UUVs. The later are required to be positioned at all time in order to later extract the best image of the underground structures and therefore be able to localize the reservoirs.

As this thesis does not study the design of the vehicles themselves but the algorithms embedded, we first suppose that all UUVs have the computing power to perform the necessary computation for the algorithms and decision-making. The UUVs are not tethered to the surface vessels and due to the large number to be deployed. The vehicles have a propulsion system and are not only drifting. However, we do not suppose that the power source on the vehicles is unlimited. The vehicle should not constantly rely on its propulsion system and should drift with the current as possible.

Because of the nature of the environment, salty sea water, no radio-frequency waves can be used. Therefore GPS and radio-frequency communications cannot be used. Only acoustic signals are allowed. However, we suppose that most frequencies are accessible with very few noise. Acoustic sensors and sources are supposedly embedded in the UUV alongside the required electronics. It is supposed that the propagation of the acoustic signal in the medium is direct and not subject to attenuation within the predefined range. Passed the range, it is supposed to fade almost instantly, therefore solving medium access issues.

To reduce the cost per unit of the vehicles, they have been equipped with the noisy sensors. No inertial navigation is possible without a considerable drift making it useless. Inertial sensors can only be used to acquire the instantaneous orientation of the vehicle and cannot be integrated over time.

#### **1.3.2 Objectives**

The objective of this thesis is to provide a new localization system for underwater vehicles using set-membership techniques. This system should be able to scale to several thousands of vehicles operating simultaneously in the same area. It also should not require expensive nor very precise sensors that might bring the cost of the vehicles up.

## 1.4 Thesis Outline

The remainder of this thesis is organized as follows:

**The chapter 2: Localization for Underwater Vehicles** This chapter gives a state of the art of most known underwater localization systems that could be used for the UUV-based marine seismic survey. It underlines the known challenges to underwater vehicles: communication, energy and localization. It also provides a list of the three most commercially available positioning systems.

**The chapter 3: Set-Membership Tools** This chapter provides the set-membership tools that are used in this thesis. The set-membership approach to such a localization problem is a novelty as most underwater localization systems use Bayesian tools. As detailed in the chapter, this is not the first time set-membership tools are applied to a localization problem nor to underwater localization. However, they have never been used in such restricted environment with such a large number of collaborating vehicles.

**The chapter 4: Phase Based Localization** The first part of the contribution of this thesis is explained in this chapter. A system allowing all vehicles within range to acquire their relative position to the base of the system. The system is based on a single beacon emitting a continuous acoustic wave for the localization duration. The vehicles within range of the beacon can position themselves without any prior information but the synchronization of their clock and the clock of the beacon. Using the set-membership approach and correlating with the recorded behavior of the vehicle, the algorithm is capable of providing the set where the vehicle is likely to be. The precision and accuracy of this system depends on the frequency of the acoustic signal and the behavior of the vehicle.

**The chapter 5: Collaborative Localization** The second contribution of this system is an inter-vehicle collaborative localization system. Instead of the vehicles being localized according to a fixed beacon as the previous system, therefore constraining their navigation area ; this system allows vehicles to track their surrounding neighbors and then maintain their position accordingly in a moving frame. Because this system does not rely on any moored beacon, an additional tracking system is necessary to project the vehicles position on the global frame for the seismic data to be useful.

**The chapter 6: Conclusion And Perspectives** The last chapter summarizes the contributions of this thesis and the relaying issues that remain unsolved for the future research to explore.

# 2 Localization For Underwater Vehicles

## 2.1 Introduction

As previously stated, autonomous underwater vehicles are used in different fields including defense, oceanography, and oil and gas to name a few. These sectors usually preferred to invest in a complex, reliable, and fully capable system composed of only one vehicle that will carry on the mission. To supply this demand, most developed vehicle today go through very rough testing and specifications leading to a high price tag. Moreover, as most missions required one or two vehicles operating at the same time, positioning and navigation systems were requested to track very few number of vehicles simultaneously. For instance, a sonar could be used to track the AUV from the surface then sending the vehicle position back using an acoustic modem. The latter solution is unfortunately inappropriate for a large number of vehicles due to access time to the medium and the indistinguishably between the vehicles.

In this chapter, we explore the actual state of the art in underwater vehicle localization and their limitations, therefore laying the motivations of this thesis.

## 2.2 Limitations to Underwater Vehicles

### 2.2.1 Communications

Due to the environment in which the AUV will operate, *i.e.* water, communication is a big issue. The least communication is possible, the more the vehicle is supposed to be autonomous. Therefore, underwater vehicles usually embed more autonomy than their aerial and terrestrial counterparts as these can rely on higher bandwidths and easier access to the medium.

Data to be transmitted include commands, navigation information and status of the vehicles in addition to the sensors measurements. The effectiveness of the communication channel is affected by the range, the bandwidth, and the network infrastructure.

### 2.2.1.1 Electromagnetic Communication

Electromagnetic waves do not propagate far underwater because of their high absorption in water:  $a = \sqrt{\pi \cdot f \cdot \mu \cdot \sigma}$  ( $f$  being the frequency,  $\mu$  the magnetic permeability, and  $\sigma$  the electrical conductivity). Dissolved salts in sea water makes it more conductive, therefore making it more absorbent to these waves. Additionally, the higher speed of electromagnetic waves propagation in water compared to acoustic allow higher bandwidth keeping it interesting for close range at relatively low frequencies.

### 2.2.1.2 Acoustic Communication

Acoustic (pressure) waves are better for underwater communication due to their relatively low absorption in water. Nevertheless, acoustic propagation underwater faces other challenges because of the variability of the properties spatially and temporally. The most known issues are ambient noise, scattering, and multi-path.

**Ambient noise** Marine mammals, surface waves along with the vehicle propulsion system cause a considerable amount of background noise. As sources are multiple and usually unidentified [13], it is difficult to filter the noise that ranges from high frequencies from marine animals like shrimp and whales to low frequencies that might have been generated by waves and far traveling ships. At short-range, the ambient noise level might be too weak relatively to the desired signal. However, the ambient noise is usually a limiting factor.

**Scattering** Bouncing on the sea surface, acoustic waves scatter therefore producing delay spreads, fluctuation of intensity of the scattered signal, and reducing the spatial correlation. On a calm and flat surface, the scattered signal is fairly stable. However, the arrival of the signal fluctuates heavily in time and delay on a rough surface.

**Multi-path** In addition to surface scattering effect, the acoustic signal reflects on both the surface and the bottom of the ocean. It even refracts due to the variation of sound speed spatially. A traveling signal from the source to the receiver may take different paths resulting in a delay spread in the time varying-impulse response. These delays may vary from tens of milliseconds typically, to a hundred [14]. For instance, in Figure 3.1, a signal  $s_0$  emitted at  $t_0$  from the source arrives to the receiver at  $t_2$  along path 1. A second signal  $s_1$  emitted at  $t_1$  arrives along the path 1 to the receivers at  $t_3$  simultaneously as the first signal  $s_1$  along the path 2 therefore resulting in a collision of data. Even though, multi-path is common issue for most communication mediums and solutions and algorithms exist, implementing these sacrifices bandwidth which is already not high underwater.

### 2.2.2 Energy

In addition to communication, access to energy sources underwater poses a problem. On-board, power is required for the sensors, processing, and mainly for the propulsion. Energy usage inside an AUV can be described as follows:

$$E = \frac{(p_p + p_v + p_{ms} + p_{pc}) \times R}{3600 \times V} \text{ (KWh)} \quad (2.1)$$

such that :

- $E$  = on-board energy (KWh),
- $R$  = range of the survey (km),
- $V$  = velocity (m/s),
- $p_p$  = propulsion power (Watts),
- $p_v$  = vehicle equipment power (W),
- $p_{ms}$  = mission sensor power (W),
- $p_{pc}$  = payload computer (W).

While processing and sensors power can be relatively low, propulsion is a function of the diameter and the cube of the speed. Actually, the propulsion power can be expressed as the speed of the vehicle times the drag force  $p_p = F_D \times V$ , which is function of the square of the speed and the surface area of the vehicle  $F_D = \frac{1}{2} \rho \times V^2 \times S \times C_D$ , with  $C_D$  being the drag coefficient and  $S$  the surface area of the AUV.

With today's battery technologies, UUVs endurance range from hours to days mainly depending on the size of the vehicle. Because larger UUVs can carry more batteries, they can carry longer missions therefore requiring better navigation. The table below provides a comparison of different type of batteries and their energy density, [15].

Technology	Energy Density	Cost	Type
Lead Acid	10-20 Wh/dm <sup>3</sup>	Low	Rechargeable
NiCd/NiMH	10-30 Wh/dm <sup>3</sup>	Low	Rechargeable
Alkaline	10-30 Wh/dm <sup>3</sup>	Low/High	Primary
AgZn	30-50 Wh/dm <sup>3</sup>	High	Rechargeable
Lithium ion	40-70 Wh/dm <sup>3</sup>	Medium	Rechargeable
Lithium polymer	50-75 Wh/dm <sup>3</sup>	Medium	Rechargeable
AlO <sub>2</sub>	80-90 Wh/dm <sup>3</sup>	Medium	Semi-fuel cell
Hydrogen-oxygen	100+ Wh/dm <sup>3</sup>	Medium	Fuel cell
Lithium	100-150 Wh/dm <sup>3</sup>	Low	Primary

## 2.2.3 Localization & Navigation

Finally localization and navigation is one of the most important challenges for underwater vehicles. Aerial and ground robots can heavily rely on a global positioning system for navigation, like GPS or GLONASS. The lack of propagation of electromagnetic signal underwater makes commonly easy and available positioning system unavailable, [16]. In this subsection we provide a list of most used sensors in underwater positioning and navigation. It is important to note that our contribution does not aim to replace any of these sensors but to add a layer of redundancy to acquire a better position estimation. Then we provide a list of the most available acoustic localization systems commercially that the first contribution of this thesis compares to.

### 2.2.3.1 Sensors

Underwater, like any other environment, proprioceptive and exteroceptive sensors are critical for better localization and navigation. Data from the sensors can either be fused to provide better positioning or in a unique manner when no redundancy is available.

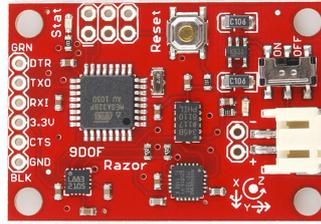
**Pressure Gauge** The pressure sensor allows underwater vehicles to measure their depth accurately. Therefore, other navigation systems only require to solve a 2D localization problem.



**Figure 2.1:** Pressure Gauge.

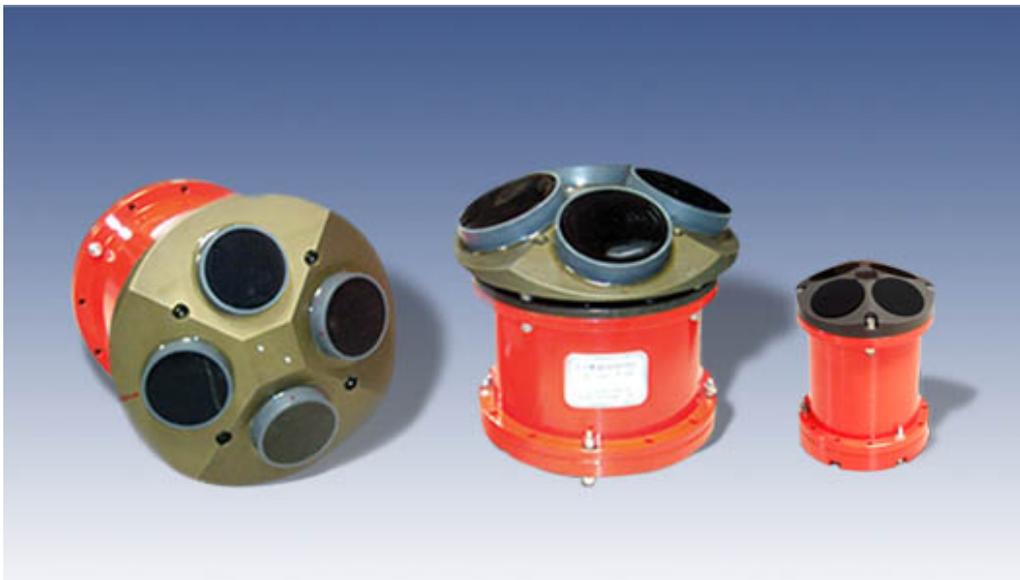
**Magnetic Compass** A magnetic compass is a basic sensor just like the pressure sensor that most UUVs are equipped with. It provides a 3D-vector of the magnetic field thus the orientation of the vehicle in the 3D space. However, the reliability of this sensor is questionable as some magnetic anomalies and electrical currents inside the vehicle affect the measurements heavily. The noise is usually too strong and cannot be discerned from Earth's magnetic field.

**Attitude and Heading Reference System** An Attitude and Heading Reference System (AHRS) combines 3 axis accelerometers, 3 axis gyroscopes and a magnetic compass to provide a more accurate heading of the vehicle in the 3D space.



**Figure 2.2:** 9-Degrees Of Freedom Attitude And Heading Reference System Made Of Inexpensive Phone Components.

**Velocity Meter** It is important to distinguish the two velocities that can be measured by a velocity meter underwater: velocity in the surrounding mass of water, and velocity over the ground or absolute speed. The first velocity can be measured using a flow sensor usually mounted in the bow of the vehicle, or approximated from the rotation rate of the propeller. Absolute velocity can be measured using a Doppler Velocity Log (DVL). A DVL measures the Doppler-shift of the reflected signals on the sea bottom that has emitted by its transceivers. A DVL typically has 4 sensors mounted with an angle with the respect of the sea floor plane. The measurement of the 4 sensors are combined to compute the vehicle's speed in the 3D space.



**Figure 2.3:** Different Sizes Of Doppler Velocity Logs Made By LinkQuest.

**Inertial Navigation System** An Inertial Navigation System (INS) is an AHRS that, in addition of providing the orientation of the vehicle in the 3D space, is capable of positioning the vehicle. This can be achieved usually by fusing data from the AHRS and other sensors in order to provide a position of the vehicle. For instance, when the vehicle is on the surface, it can acquire an absolute position from the GPS and then integrate the AHRS measurements underwater to estimate the vehicle's position. This technique is called Dead-Reckoning but is only used when accurate and precise sensors are available or for very short mission as integrated measurement errors lead to great drift rates, typically of few meters per minute [17].

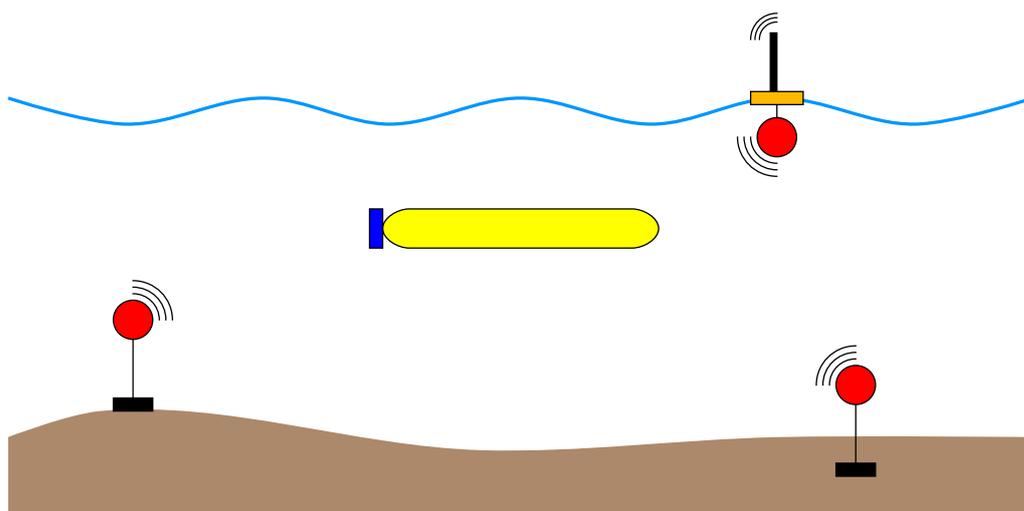
### 2.2.3.2 Commercially Available Acoustic Localization Systems

Along with the previously cited sensors, underwater navigation commonly uses acoustic beacon systems. Many solutions are widely commercially available from different manufacturers to provide either a global or a relative positioning system. These solutions measure the distance and/or directions of the vehicle from a set of transponders. Most of these systems can fall into one of the following category.

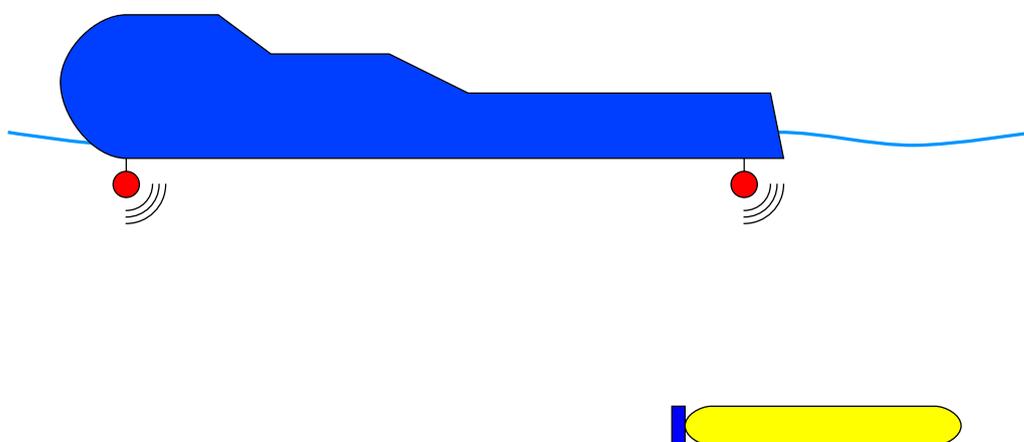
**LBL** A Long Baseline (LBL) is an acoustic system where the distance between the baseline stations (beacons) is long or similar to the distance between the vehicle and the transponders. For the target to acquire its position, it interrogates the baseline stations by emitting a pulse. Each station of the baseline responds back and the position is computed from the Time-of-Flight (ToF). The baseline can either be moored on the seafloor or the surface. Alternatively, the baseline can be synchronized and emit pings at the same time therefore not requiring the vehicle interrogation. This is used on the GPS Intelligent Buoys (GIB) which are basically a LBL on the sea surface emitting synchronized pings. Because the GIBs are aware of their global position, they can be drifting and embedding their position in the pings.

**SBL** A Short Baseline (SBL) is different from the LBL by the inter-distance between the beacons, 20 to 50 meters. These systems are usually mounted on the operation's ship itself. Like LBL, the SBL uses ToF of the acoustic signal between the emission and the reception of the signal and the greater the distance between the beacons, the better is the accuracy. Therefore it is preferred to use these systems on large ships rather than small boats.

**USBL** Being the easiest to use of the previous systems, the Ultra-Short Baseline (USBL) is the most popular positioning system underwater. The USBL uses a transceiver made of an array of transducers that are about 10cm apart, hence the name. For localization, a pulse is emitted by the array, the transducer on the underwater target then responds with another pulse that is detected by the array.



**Figure 2.4:** Long Baseline.



**Figure 2.5:** Short Baseline.

The round-trip propagation time provides the range between the array and the target. The direction is measured with the phase difference between the different components of the transceiver array, therefore providing a localization of the target in a 3D space. If necessary, the target gets its position via acoustic modem when available.

Surveys about these systems can be found in [18, 19, 20].



**Figure 2.6:** IXBlue Ultra-Short Baseline (GAPS) And A Modem Mounted On A Ship.

## 2.3 Conclusion

In this chapter we identified some the most noticeable issues that face autonomous marine vehicles. While this domain may still rely in advances made in other areas to improve its capabilities like battery technologies and sensors, untethered underwater communication and localization still need to be solved as separated issues.

As it will be described later in this thesis, localization systems developed in its scope require very little communication. Also these systems are required to compete with the commercially available localization system such as the USBL also solving the known issues when localizing a large number of vehicles. A brief comparison is provided to highlight the pros and cons of the developed systems against the available ones.

Next is an introduction to set-membership, the tool that will be used across the thesis to solve the localization issue.

# 3 Set-Membership Tools

## 3.1 Introduction

This chapter introduces set-membership approach to solve complex non-linear system of equations like the ones found in underwater localization as an alternative approach to probabilistic approaches [21, 22, 23, 24]. Previous work have been conducted in this field for SLAM [25, 26, 27, 28, 29, 30], path planning [31, 32, 33], design for robust controllers [34] or the characterization of the state evolution of dynamic systems[35, 36].

To illustrate the tools used, the problem of planar localization [37] where a robot is positioned at an unknown Cartesian position measures its distance to beacons at known positions. Suppose that each beacon has a unique signature that makes it distinguishable from the rest of the other beacons. This system can either be based on radio-communications [38] or acoustic communications LBL [39], Figure 3.1, as explained in the subsection 2.2.3.2. Distances between the robot and the beacons can be measured using the Time-of-Flight (ToF) techniques.

The observation model is:

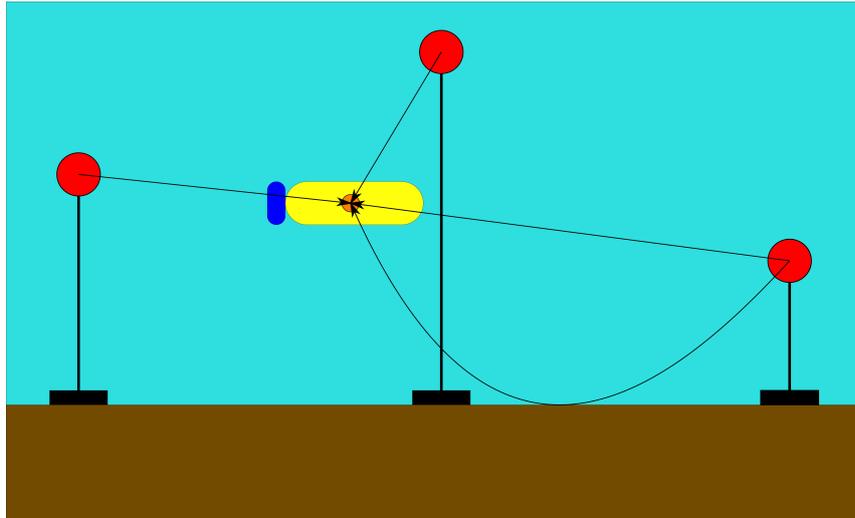
$$d_i = \sqrt{(x - x_i)^2 + (y - y_i)^2}, i = 1 \dots n_B, \quad (3.1)$$

where  $(x, y)$  are the coordinates of the robot, and  $(x_i, y_i)$  are the coordinates of the beacon  $i$ , and  $d_i$  is the measured distance between the robot and the beacon  $i$ . This is a classical non-linear problem often found in localization.

Plus, outliers are introduced either from measurement errors (skews on clocks, interference, etc.) or from the model itself as the hypothesis considers that all measurements came from direct-flight trajectories, where it is not always the case in the reality due to reflections and refraction that might cause multi-path and other unwanted effects of the measurements.

To solve the localization problem for this robot without any ambiguous solution, at least three non-aligned beacons must be used. This technique is called triangulation. Usually, the number of measurements is far more that the minimum required. The redundancy can then either help with the precision of the position or help detect and remove the outliers .

The section 3.2 describes how to solve this problem using the set-membership approach. The section 3.3 introduces the interval analysis techniques that will be used.



**Figure 3.1:** Localization of a robot using beacons (LBL) and having multi-path.

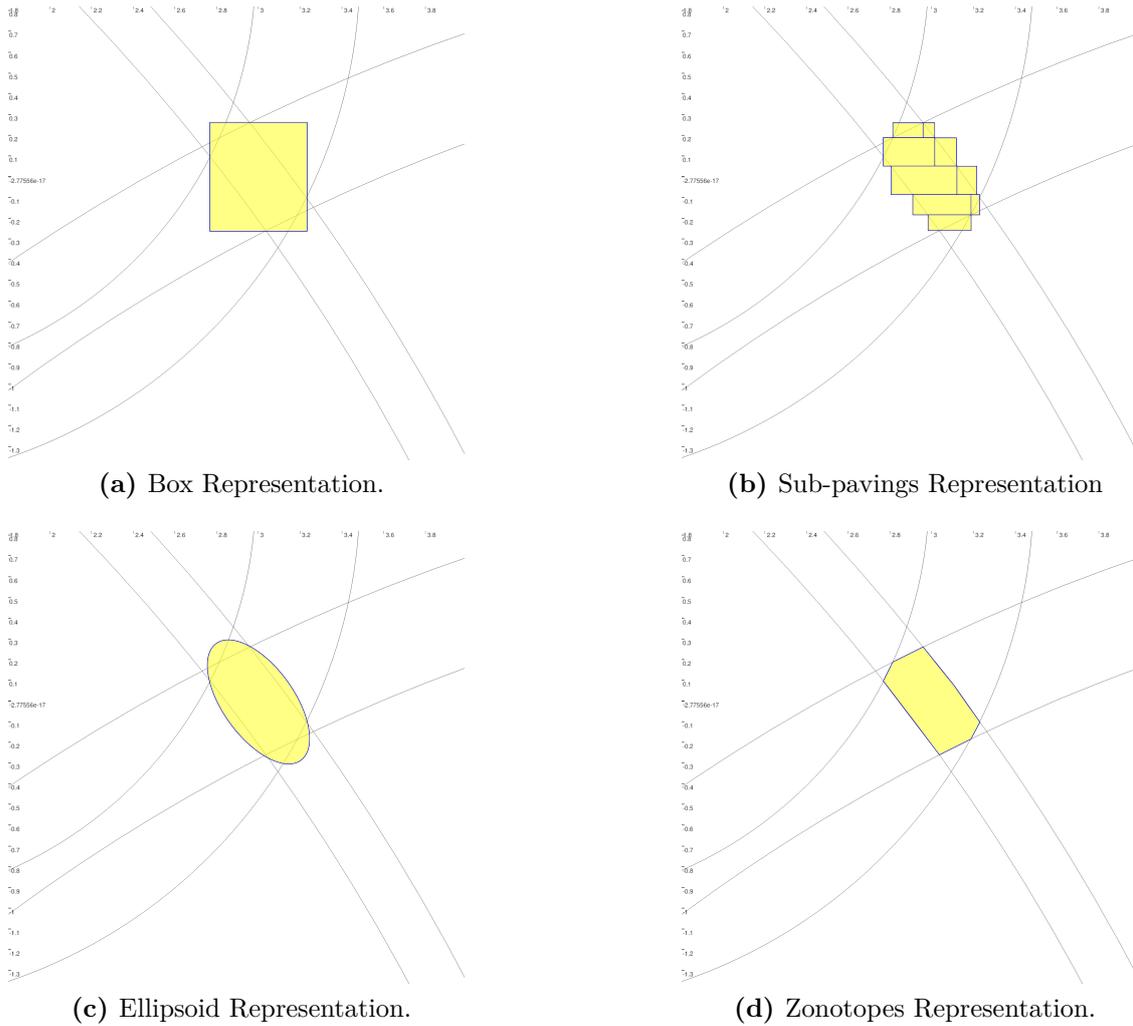
The subsection 3.4.2 explains how to represent a problem as a set of constraints. Finally, section 3.5 introduces the algorithm used to solve the problem.

## 3.2 Set-Membership Approach

The set-membership approach principle is to study a problem with bounded errors as a hypothesis. This approach consists on finding all the compatible parameters with the observations made and the error bounds that are supposed to be known. Therefore, the problem consists of computing all the parameter vectors compatible with the observations and not minimizing the error criteria.

Consider now the same system introduced earlier in this chapter where the robot measures its distance to the beacons with a bounded error. The true distance between the robot and a beacon  $i$  can be written  $d_i^{true} \in d_i^{measured} + [e_i]$  where  $d_i^{true}$  is the real distance between the robot and the beacon,  $d_i^{measured}$  is the measured distance and  $[e_i]$  is an interval enclosing the measurement error. Note that the interval  $[e_i]$  can vary from a measurement to another and from one beacon to another. Therefore, the position of the robot can only exist within a ring centered on the beacon  $i$  with  $d_i^{measured}$  as a diameter and a width  $[e_i]$ . The intersection of the rings must then enclose the robot position.

Notice that depending on the robot position and the position of the beacons and their number, the solution set can have random shape. For this reason, many set-membership representations have been developed to accommodate the solution sets and the observations. Figure 3.2 illustrates the most commonly used representation of a set of solution: *interval boxes* [40], *sub-pavings* [41], *ellipsoids* [42] and *polyhedral approximations (zonotopes)* [43, 44, 45]. *Ellipsoids* look the most familiar as most probabilistic methods use ellipsoids to represent the error around the estimated solution. In this thesis however, only *interval boxes* and *sub-pavings* will be considered as they have been proven to be the most efficient [46].



**Figure 3.2:** Different representation of a set of solution.

### 3.3 Interval Analysis

Interval analysis considers intervals instead of real numbers. It has been first developed to tackle the issue of errors in numerical computations [40]. As real numbers are represented by floating-points with a limited significant digits in computers, errors can accumulate drastically and lead to completely false results [47, 48]. Also before the standardization of floating-point arithmetic [49, 50] in 1985, chip manufacturers and libraries could use different ways to represent floating-points leading to various errors and different solutions from a system to another. However, in this thesis we are more interested in this method to manipulate the uncertainty of the observations.

### 3.3.1 Real Intervals

**Real Interval.** A real *interval* is a connected, closed set of  $\mathbb{R}$ . The set of all real intervals of  $\mathbb{R}$  is denoted by  $\mathbb{I}\mathbb{R}$ . Real interval values will be denoted between brackets to ease the reading and will be only called interval.

An interval  $[x]$  is defined as the set of the real values  $x$  that are bounded by the lower bound  $\underline{x}$  and the upper bound  $\bar{x}$ :

$$[x] = [\underline{x}, \bar{x}] = \{x \in \mathbb{R}, \underline{x} \leq x \leq \bar{x}\}. \quad (3.2)$$

For example  $[1, 3]$ ,  $\{1\}$ ,  $\mathbb{R}$ ,  $]-\infty, -3]$  and  $\emptyset$  are considered as real intervals. Whereas  $[3, 2]$ ,  $]1, 5]$  and  $[1, 2] \cup [3, 4]$  are not.

**Width.** The *width* of an interval  $[x]$  is defined by

$$w([x]) = \bar{x} - \underline{x}. \quad (3.3)$$

**Midpoint.** The *midpoint* of an interval  $[x]$  is defined by

$$mid([x]) = \frac{\underline{x} + \bar{x}}{2}. \quad (3.4)$$

**Intersection.** The *intersection* of two non-empty intervals  $[x]$  and  $[y]$  satisfies

$$[x] \cap [y] = \begin{cases} [\max(\underline{x}, \underline{y}), \min(\bar{x}, \bar{y})] & \text{if } \max(\underline{x}, \underline{y}) \leq \min(\bar{x}, \bar{y}) \\ \emptyset & \text{otherwise} \end{cases}. \quad (3.5)$$

**Enveloping Interval.** The *enveloping interval* of a subset  $\mathbb{X}$  of  $\mathbb{R}$  is the smallest interval containing  $\mathbb{X}$  and is denoted by  $[\mathbb{X}]$ .

For instance

$$[[-5, 1] \cup [4, 5]] = [-5, 5].$$

**Interval Union.** The *interval union*  $\sqcup$  of two non-empty intervals  $[x]$  and  $[y]$  is the enveloping interval of the simple *union*  $\cup$  of these intervals (that might not be an interval), *i.e.*

$$[x] \sqcup [y] = [[x] \cup [y]] = [\min(\underline{x}, \underline{y}), \max(\bar{x}, \bar{y})]. \quad (3.6)$$

**Deprivation.** The *deprivation*  $\setminus$  of two non-empty intervals  $[x]$  and  $[y]$  satisfies

$$[x] \setminus [y] = \{x | x \in [x] \text{ and } x \notin [y]\}. \quad (3.7)$$

**Punctual Interval.** A *punctual interval* is denoted by  $\{x\} = [x, x]$ .

### 3.3.2 Boxes

**Box.** A *box*  $[\mathbf{x}]$  (or *interval vector*) is an interval of  $\mathbb{R}^n$ . It corresponds to the Cartesian product of  $n$  intervals *i.e.* a vector with interval components:

$$[\mathbf{x}] = [x_1] \times \cdots \times [x_n]. \quad (3.8)$$

The set of all boxes of  $\mathbb{R}^n$  will be denoted by  $\mathbb{IR}^n$ .

**Bounds.** The *lower* and *upper bound* of a box  $[\mathbf{x}]$  are defined as follows

$$\underline{\mathbf{x}} = (\underline{x}_1, \dots, \underline{x}_n)^T, \quad (3.9)$$

$$\overline{\mathbf{x}} = (\overline{x}_1, \dots, \overline{x}_n)^T. \quad (3.10)$$

**Width.** The *width* of a box  $[\mathbf{x}]$  is the width of its largest side

$$w([\mathbf{x}]) = \max_{i \in \{1, \dots, n\}} w([x_i]). \quad (3.11)$$

By convention  $w(\emptyset) = -\infty$ . If  $w([\mathbf{x}]) = 0$ ,  $[\mathbf{x}]$  is said to be *degenerated*. In such a case,  $[\mathbf{x}]$  is a singleton of  $\mathbb{R}^n$  and will be denoted  $\{\mathbf{x}\}$ .

**Principal plane.** The *principal plane* of  $[\mathbf{x}]$  is the symmetric plane of  $[\mathbf{x}]$  perpendicular to its largest side.

**Bisection.** To *bisect* a box  $[\mathbf{x}]$  means to split it into two separate parts following a plane, usually the principal plane, Figure 3.3.

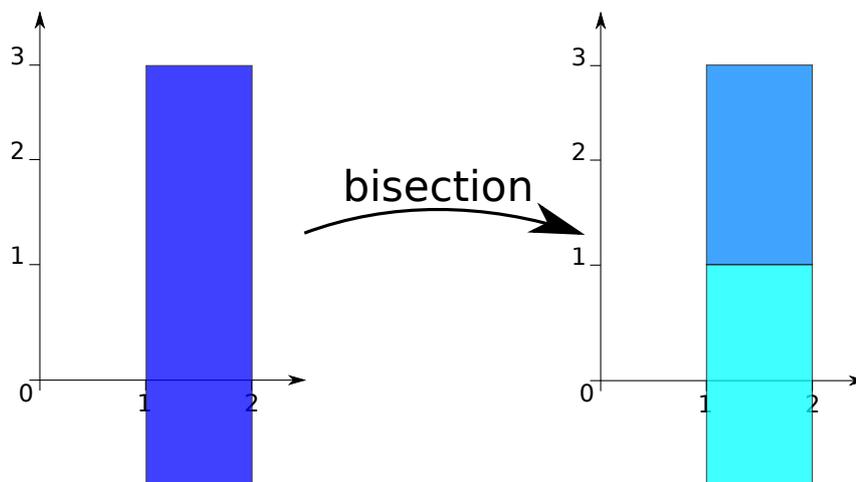
### 3.3.3 Interval Arithmetic

Given the  $n$  intervals  $[x_1], \dots, [x_n]$ . A  $n$ -ary operator  $\diamond$  can be extended to real intervals and boxes as follows

$$[x_1] \diamond \cdots \diamond [x_n] = \{x_1 \diamond \cdots \diamond x_n \mid x_1 \in [x_1], \dots, x_n \in [x_n]\}. \quad (3.12)$$

For instance, we shall consider the case where  $n = 1$ . Consider  $\diamond \in \{+, -, \times, /, \max, \min, \dots\}$  a binary operation in  $\mathbb{R}$ . The extended operator in  $\mathbb{IR}$  is defined as follows

$$[x] \diamond [y] = \{x \diamond y \mid x \in [x], y \in [y]\}. \quad (3.13)$$



**Figure 3.3:** The bisection of  $[x] = [1, 2] \times [-1, 3]$  according to its principal plane generates two boxes  $[x]_1 = [1, 2] \times [-1, 1]$  and  $[x]_2 = [1, 2] \times [1, 3]$ .

As a consequence,

$$[x] + [y] = [\underline{x} + \underline{y}, \bar{x} + \bar{y}], \quad (3.14)$$

$$[x] - [y] = [\underline{x} - \bar{y}, \bar{x} - \underline{y}], \quad (3.15)$$

$$[x] \times [y] = [\min(\underline{x}\underline{y}, \underline{x}\bar{y}, \bar{x}\bar{y}, \bar{x}\underline{y}), \max(\underline{x}\underline{y}, \underline{x}\bar{y}, \bar{x}\bar{y}, \bar{x}\underline{y})]. \quad (3.16)$$

The inversion is given by

$$1/[y] = \begin{cases} \emptyset & \text{if } [y] = \{0\} \\ [1/\bar{y}, 1/\underline{y}] & \text{if } 0 \notin [y] \\ [1/\bar{y}, \infty[ & \text{if } \underline{y} = 0 \text{ and } \bar{y} > 0 \\ ]-\infty, 1/\underline{y}] & \text{if } \underline{y} < 0 \text{ and } \bar{y} = 0 \\ ]-\infty, \infty[ & \text{if } \underline{y} < 0 \text{ and } \bar{y} > 0 \end{cases} \quad (3.17)$$

and the division by

$$[x] / [y] = [x] \times (1/[y]). \quad (3.18)$$

For example:

$$[1, 4] + [-3, 2] = [-2, 6]$$

$$[2, 6] - [3, 6] = [-4, 3]$$

$$[1, 3] \times [4, 7] = [4, 21]$$

$$[-1, 0] \times [3, 9] = [-9, 0]$$

$$[1, 2] / [3, 4] = \left[ \frac{1}{4}, \frac{2}{3} \right]$$

Similarly, a function  $f : \mathbb{R}^n \rightarrow \mathbb{R}^m$  can be extended to real intervals/boxes as follows

$$[f([x])] = [\{f(x) \mid x \in [x]\}]. \quad (3.19)$$

Notice that the enveloping interval was used and not the direct image of the interval/box as the later might not be an interval/box. Actually, the image of an interval using a discontinuous function can be a union of intervals.

Examples:

$$\sin([-3, 8]) = [-1, 1]$$

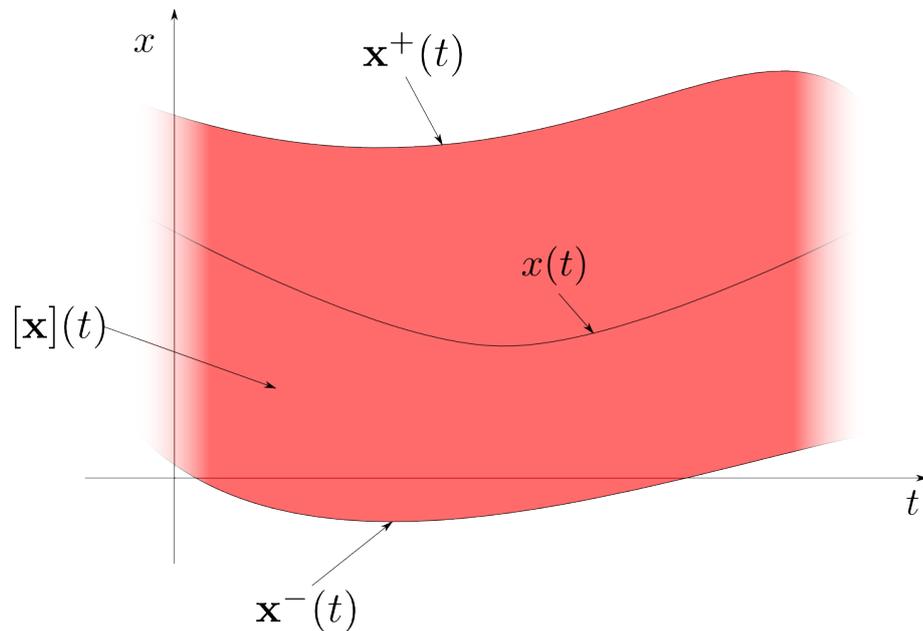
$$\sqrt{[-1, 4]} = [0, 2]$$

$$[-4, 3]^2 = [0, 16]$$

### 3.3.4 Tubes

**Tube.** A tube (or *interval of function*), [51, 52, 53],  $[\mathbf{x}](t)$  is a function from  $\mathbb{R} \rightarrow \mathbb{IR}^n$ .

**Bounds.** A tube  $[\mathbf{x}](t)$  is bounded by two trajectories, [54],  $\mathbf{x}^-(t)$  and  $\mathbf{x}^+(t)$  such as  $\forall t, \mathbf{x}^-(t) \leq \mathbf{x}^+(t)$ .



**Figure 3.4:** Tube Example.

Figure 3.4 shows an example of tube  $[\mathbf{x}](t) : \mathbb{R} \rightarrow \mathbb{IR}$ . The tube is bounded by the functions  $\mathbf{x}^-(t) \in [\mathbf{x}](t)$  and  $\mathbf{x}^+(t) \in [\mathbf{x}](t)$ .  $x(t) : \mathbb{R} \rightarrow \mathbb{R}$  is a function included in the tube  $[\mathbf{x}](t)$  that represents a real valued function.

## 3.4 Contractors

### 3.4.1 Definitions and Properties

**Contractor.** An operator  $\mathcal{C} : \mathbb{IR}^n \rightarrow \mathbb{IR}^n$  is a *contractor* if

$$\begin{aligned}
 \text{(i)} \quad & \forall [\mathbf{x}] \in \mathbb{IR}^n, \mathcal{C}([\mathbf{x}]) \subset [\mathbf{x}] && \text{(contractance)} \\
 \text{(ii)} \quad & (\mathbf{x} \in [\mathbf{x}], \mathcal{C}(\{\mathbf{x}\}) = \{\mathbf{x}\}) \Rightarrow \mathbf{x} \in \mathcal{C}([\mathbf{x}]) && \text{(consistence)} \\
 \text{(iii)} \quad & \mathcal{C}(\{\mathbf{x}\}) = \emptyset \Rightarrow (\exists \epsilon > 0, \forall [\mathbf{x}] \subset B(\mathbf{x}, \epsilon), \mathcal{C}([\mathbf{x}]) = \emptyset) && \text{(weak continuity)}
 \end{aligned}
 \tag{3.20}$$

where  $B(\mathbf{x}, \epsilon)$  is the ball with the center  $\mathbf{x}$  and the radius  $\epsilon$ .

A box  $[\mathbf{x}]$  is said *insensitive* to  $\mathcal{C}$  if  $\mathcal{C}([\mathbf{x}]) = [\mathbf{x}]$ .

From property (i), boxes can only be contracted. From property (ii), an insensitive point  $\mathbf{x}$  is never removed by  $\mathcal{C}$ . From property (iii), the set of all insensitive  $\mathbf{x}$  is closed.

A contractor represents a subset of  $\mathbb{R}^n$ . Set operations such as intersection, union, inversion are easy to perform with contractors.

We have the following definitions

$\mathcal{C}$ is <i>monotonic</i> if	$[\mathbf{x}] \subset [\mathbf{y}] \Rightarrow \mathcal{C}([\mathbf{x}]) \subset \mathcal{C}([\mathbf{y}])$
$\mathcal{C}$ is <i>minimal</i> if	$\forall [\mathbf{x}] \in \mathbb{IR}^n, \mathcal{C}([\mathbf{x}]) = [[\mathbf{x}] \cap \text{set}(\mathcal{C})]$
$\mathcal{C}$ is <i>idempotent</i> if	$\forall [\mathbf{x}] \in \mathbb{IR}^n, \mathcal{C}(\mathcal{C}([\mathbf{x}])) = \mathcal{C}([\mathbf{x}])$
$\mathcal{C}$ is <i>continuous</i> if	$\forall [\mathbf{x}] \in \mathbb{IR}^n, \mathcal{C}(\mathcal{C}^\infty([\mathbf{x}])) = \mathcal{C}^\infty([\mathbf{x}])$

We define also the following operations

intersection	$(\mathcal{C}_1 \cap \mathcal{C}_2)([\mathbf{x}]) \stackrel{def}{=} \mathcal{C}_1([\mathbf{x}]) \cap \mathcal{C}_2([\mathbf{x}])$
union	$(\mathcal{C}_1 \cup \mathcal{C}_2)([\mathbf{x}]) \stackrel{def}{=} [\mathcal{C}_1([\mathbf{x}]) \cup \mathcal{C}_2([\mathbf{x}])]$
composition	$(\mathcal{C}_1 \circ \mathcal{C}_2)([\mathbf{x}]) \stackrel{def}{=} \mathcal{C}_1(\mathcal{C}_2([\mathbf{x}]))$
repetition	$\mathcal{C}^\infty \stackrel{def}{=} \mathcal{C} \circ \mathcal{C} \circ \mathcal{C} \circ \dots$
repeat intersection	$\mathcal{C}_1 \sqcap \mathcal{C}_2 \stackrel{def}{=} (\mathcal{C}_1 \cap \mathcal{C}_2)^\infty$
repeat union	$\mathcal{C}_1 \sqcup \mathcal{C}_2 \stackrel{def}{=} (\mathcal{C}_1 \cup \mathcal{C}_2)^\infty$
central symmetry	$(\mathcal{S}_{\mathbf{a}} \circ \mathcal{C})([\mathbf{x}]) \stackrel{def}{=} \mathcal{S}_{\mathbf{a}} \circ \mathcal{C} \circ \mathcal{S}_{\mathbf{a}}([\mathbf{x}])$
axial symmetry	$(\mathcal{S}_{\mathbf{u}} \circ \mathcal{C})([\mathbf{x}]) \stackrel{def}{=} \mathcal{S}_{\mathbf{u}} \circ \mathcal{C} \circ \mathcal{S}_{\mathbf{u}}([\mathbf{x}])$
translation	$(\mathcal{T}_{\mathbf{u}} \circ \mathcal{C})([\mathbf{x}]) \stackrel{def}{=} \mathcal{T}_{\mathbf{u}} \circ \mathcal{C} \circ \mathcal{T}_{-\mathbf{u}}([\mathbf{x}])$
modulo	$(\mathcal{C} \text{ mod } \mathbf{u})([\mathbf{x}]) \stackrel{def}{=} \sqcup_i \mathcal{T}_{i, \mathbf{u}} \circ \mathcal{C}([\mathbf{x}])$

### 3.4.2 Constraint Satisfaction Problem

Consider  $n_{\mathbf{x}}$  variables  $x_i \in \mathbb{R}$ ,  $i \in \{1, \dots, n_{\mathbf{x}}\}$ , linked by  $n_{\mathbf{f}}$  relations (or *constraints*) of the form

$$f_j(x_1, \dots, x_{n_{\mathbf{x}}}) = 0, j \in \{1, \dots, n_{\mathbf{f}}\}. \quad (3.21)$$

Each variable  $x_i$  is known to belong to a domain  $\mathbb{X}_i$ . For simplicity, these domains will be intervals, denoted by  $[x_i]$ , but unions of intervals could be considered as well.

Let  $\mathbf{x}$  be the vector of all the variables  $x_i$  and  $\mathbf{f}$  be the function whose coordinates are  $f_j$ s. The Equation 3.21 can then be written  $\mathbf{f}(\mathbf{x}) = \mathbf{0}$ . This corresponds to a *constraint satisfaction problem (CSP)*  $\mathcal{H}$ , which can be formulated as

$$\mathcal{H} : (\mathbf{f}(\mathbf{x}) = \mathbf{0}, \mathbf{x} \in [\mathbf{x}]). \quad (3.22)$$

The *solution set* of  $\mathcal{H}$  is defined as

$$\mathbb{S} = \{\mathbf{x} \in [\mathbf{x}], \mathbf{f}(\mathbf{x}) = \mathbf{0}\}. \quad (3.23)$$

Such CSPs are not restricted to equality constraints and may also involve inequalities. For instance, the set of constraints

$$\begin{cases} x_1 + \sin(x_2^2) & \leq 0 \\ x_1 - x_2 & = 3 \end{cases}$$

can be cast into the CSP framework by introducing a slack variable  $x_3$  in order to get the set of constraints

$$\begin{cases} x_1 + \sin(x_2^2) + x_3 & = 0 \\ x_1 - x_2 - 3 & = 0 \end{cases},$$

where the domains for the variables are  $[x_3] = [0, \infty[$ ,  $[x_1] = \mathbb{R}$ ,  $[x_2] = \mathbb{R}$  and the coordinate functions  $f_j$  are  $f_1(\mathbf{x}) = x_1 + \sin(x_2^2) + x_3$  and  $f_2(\mathbf{x}) = x_1 - x_2 - 3$ . Characterizing the solution set  $\mathbb{S}$  is NP-hard in general, which means that no algorithm with a complexity polynomial in the number of variables is available to obtain an accurate approximation of  $\mathbb{S}$  in the worst case.

Originally, CSPs were defined for discrete domains, *i.e.* the values taken by the  $x_i$ s belonged to finite sets [55]. Later, CSPs were extended to continuous domains (subsets of  $\mathbb{R}$  or intervals) [56, 57, 58, 59, 60, 61, 62, 46]. Most of the algorithms described in these paper use consistency techniques to find an outer approximation of the set  $\mathbb{S}$  of all solutions of  $\mathcal{H}$ . The main advantage of these techniques is that they yield a guaranteed enclosure of  $\mathbb{S}$  with a complexity that can be kept polynomial in time and space.

*Contracting*  $\mathcal{H}$  means replacing  $[\mathbf{x}]$  by a smaller domain  $[\mathbf{x}']$  such that the solution set remains unchanged, *i.e.*  $\mathbb{S} \subset [\mathbf{x}'] \subset [\mathbf{x}]$ . There exists an optimal contraction of  $\mathcal{H}$  which corresponds to replacing  $[\mathbf{x}]$  by the smallest box that contains  $\mathbb{S}$ . Notice that the contractors defined in subsection 3.4.1 extend to CSPs. As many problems of estimation, control and robotics can be represented as CSP [63, 64] and many contractors can be designed depending on the class of the problem [65, 46], we propose to use the *forward-backward* contractor, also known as *HC4-Revise* [66], for our localization problem.

### 3.4.3 Forward-Backward Contractor

The *forward-backward* consist of contracting the domain of the CSP  $\mathcal{H}$  by isolating each constraint  $f_j$  separately and then breaking the constraint into a series of operations involving operators and elementary operations and functions such as  $+$ ,  $-$ ,  $\times$ ,  $/$ ,  $\exp$ ,  $\cos$ , etc. called *primary constraints* and their inverse.

Our localization problem can easily be put as a CSP as the associated constraint with each beacon is written

$$d_i - \sqrt{(x_i - x)^2 + (y_i - y)^2} = 0, i = 1 \dots n_B. \quad (3.24)$$

Therefore it can be broken into primary constraints by introducing intermediate variables:

$$\begin{aligned} t_1 &= -y \\ t_2 &= y_i + t_1 \\ t_3 &= -x \\ t_4 &= x_i + t_3 \\ t_5 &= t_2^2 \\ t_6 &= t_4^2 \\ t_7 &= t_5 + t_6 \\ t_8 &= \sqrt{t_7} \\ t_9 &= -t_8 \\ t_{10} &= d_i + t_9 \end{aligned} \quad (3.25)$$

The initial domains associated with the intermediate variables  $t_i$  are  $\mathbb{R}$ . Then each contractor related to a primitive constraint is applied until a fix point is reached. This is the principle of constraint propagation introduced by Waltz [67]. Constraints are then broken in two steps of contraction: one from the direct image of the function and one from the inverse.

For instance, the square root is rewritten in two forms:

$$t_8 = \sqrt{t_7} \quad (3.26)$$

$$t_7 = t_8^2 \quad (3.27)$$

and the contraction steps are:

$$[t_8] = [t_8] \cap \sqrt{[t_7]} \quad (3.28)$$

$$[t_7] = [t_7] \cap [t_8]^2 \quad (3.29)$$

For constraints linking more variables such as a binary operation, the constraint is rewritten in as many ways as possible using primary constraints. For instance, let's take the addition constraint  $t_7 = t_5 + t_6$ . This constraint can be rewritten in three different ways:

$$t_7 = t_5 + t_6 \quad (3.30)$$

$$t_5 = t_7 - t_6 \quad (3.31)$$

$$t_6 = t_7 - t_5 \quad (3.32)$$

To illustrate the contraction, suppose  $[t_7] = [4, \infty[$ ,  $[t_5] = ]-\infty, 2]$ , and  $[t_6] = ]-\infty, 3]$ . Then

$$\begin{aligned} [t_7] &= [4, \infty[ \cap (]-\infty, 2] + ]-\infty, 3]) \\ &= [4, \infty[ \cap ]-\infty, 5] \\ &= [4, 5] \end{aligned}$$

$$\begin{aligned} [t_5] &= ]-\infty, 2] \cap ([4, \infty[ - ]-\infty, 3]) \\ &= ]-\infty, 2] \cap [1, \infty[ \\ &= [1, 2] \end{aligned}$$

$$\begin{aligned} [t_6] &= ]-\infty, 3] \cap ([4, \infty[ - ]-\infty, 2]) \\ &= ]-\infty, 3] \cap [2, \infty[ \\ &= [2, 3] \end{aligned}$$

We obtain a smaller interval for each variable:  $[t_7] = [4, 5]$ ,  $[t_5] = [1, 2]$ , and  $[t_6] = [2, 3]$ .

The same principle is applied to Equation 3.25 in order to contract the intervals around the feasible values of Equation 3.1. The algorithm presented in Algorithm 3.1 and Figure 3.5 runs on every constraint, *i.e.* for each distance measurement. The contractor is minimal for each constraint, but might not be minimal for whole system due to the dependencies between the constraints. A solution might be to run the contractor multiple times until a fixed-point is reached. A similar approach has been used to localize a real car [68]. Figure 3.6 illustrates an example where the forward-backward contractor is applied for each constraint successively multiple times until a fixed-point is reached.

---

**Algorithm 3.1** FwdBwdDist(**in** :  $[x],[y],[x_i],[y_i],[d_i]$ ; **out** :  $[x],[y]$ )

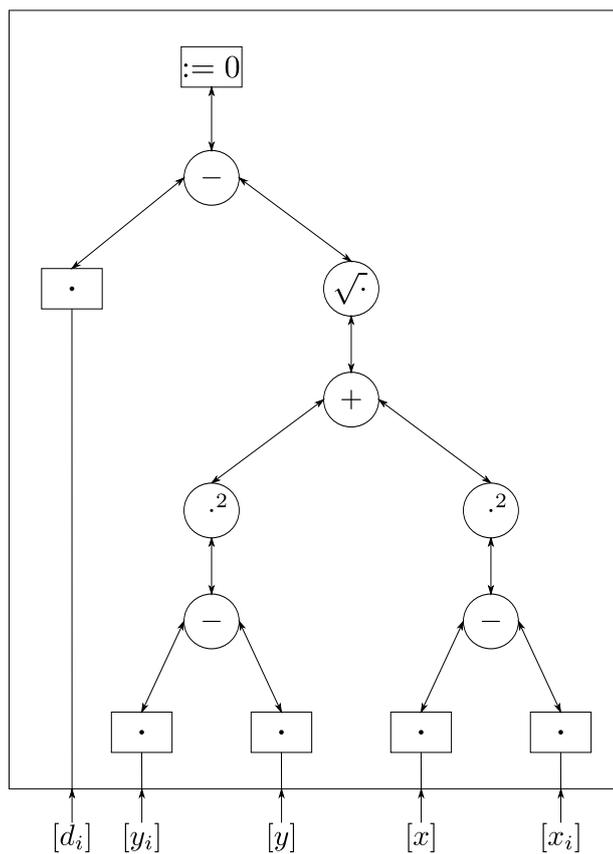
---

```

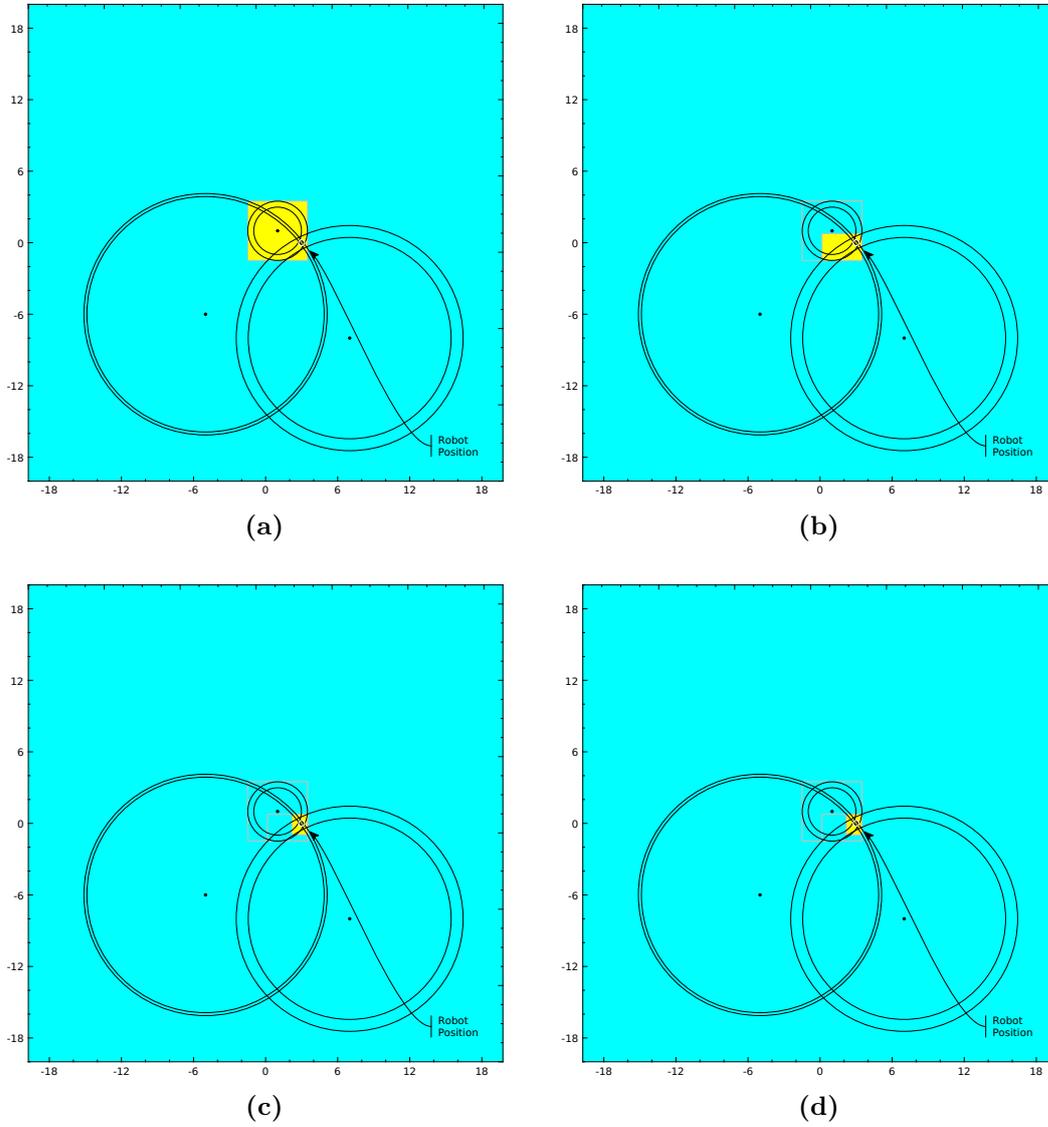
1:  $[t_1] := [t_1] \cap (-[y])$ 
2:  $[t_2] := [t_2] \cap ([y_i] + [t_1])$ 
3:  $[t_3] := [t_3] \cap (-[x])$ 
4:  $[t_4] := [t_4] \cap ([x_i] + [t_3])$ 
5:  $[t_5] := [t_5] \cap [t_2]^2$ 
6:  $[t_6] := [t_6] \cap [t_4]^2$ 
7:  $[t_7] := [t_7] \cap ([t_5] + [t_6])$ 
8:  $[t_8] := [t_8] \cap \sqrt{[t_7]}$ 
9:  $[t_9] := [t_9] \cap (-[t_8])$ 
10:  $[t_{10}] := [t_{10}] \cap ([d_i] + [t_9])$  // with  $[t_{10}] = \{0\}$  for this CSP.
    // Backward Contraction
11:  $[t_9] := [t_9] \cap ([t_{10}] - [d_i])$ 
12:  $[t_8] := [t_8] \cap (-[t_9])$ 
13:  $[t_7] := [t_7] \cap ([t_8])^2$ 
14:  $[t_6] := [t_6] \cap ([t_7] - [t_5])$ 
15:  $[t_5] := [t_5] \cap ([t_7] - [t_6])$ 
16:  $[t_4] := [t_4] \cap \sqrt{[t_6]}$ 
17:  $[t_3] := [t_3] \cap ([t_4] - [x_i])$ 
18:  $[t_2] := [t_2] \cap \sqrt{[t_5]}$ 
19:  $[t_1] := [t_1] \cap ([t_2] - [y_i])$ 
20:  $[x] := [x] \cap (-[t_3])$ 
21:  $[y] := [y] \cap (-[t_1])$ 

```

---



**Figure 3.5:** A Representation of the Forward Backward Contractor of Algorithm 3.1.



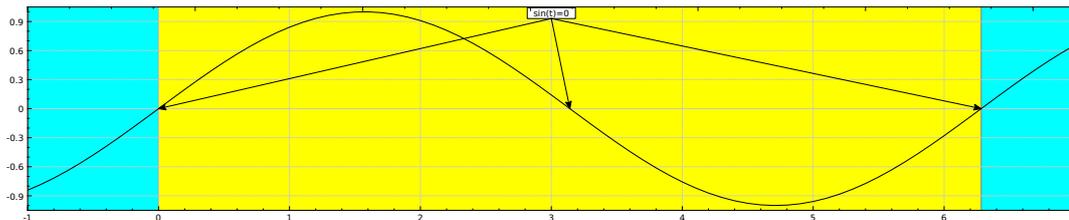
**Figure 3.6:** Forward-Backward Algorithm Applied To Distance Constraints Until A Fixed Point Is Reached

### 3.5 Set Inversion Via Interval Analysis

Previously in section 3.4, the solution set  $\mathbb{S}$  of the CSP  $\mathcal{H}$  is contracted until a fixed-point is reached. However, it is rarely the case where the resulting box provides enough accuracy around  $\mathbb{S}$ .

For instance, consider the following CSP  $\mathcal{H}_{\sin}$ , figure Figure 3.7,

$$\mathcal{H}_{\sin} : \sin [\mathbf{x}] = 0, [\mathbf{x}] = [-1, 7].$$



**Figure 3.7:** Applying A Contractor To The CSP  $\mathcal{H}_{\sin}$  Is Not Effective.

Simply applying a contractor to  $\mathcal{H}_{\sin}$  would not provide better accuracy around the solutions  $\{0\}$ ,  $\{\pi\}$ , and  $\{2\pi\}$  as two solutions bound a larger domain.

Therefore, the Set Inversion Via Interval Analysis (SIVIA) [41] was developed to solve similar issues using sub-pavings.

#### 3.5.1 Sub-Pavings

A *sub-paving* of  $\mathbb{I}\mathbb{R}^n$  is a set of non-overlapping boxes of  $\mathbb{I}\mathbb{R}^n$ . Compact sets  $\mathbb{X}$  can be bracketed between inner and outer sub-pavings:

$$\mathbb{X}^- \subset \mathbb{X} \subset \mathbb{X}^+. \quad (3.33)$$

The Figure 3.8 illustrates bracketing of the set

$$\mathbb{X} = \{(x_1, x_2) \mid x_1^2 + x_2^2 \in [3, 4]\}$$

between sub-pavings with an increasing accuracy from the left to the right. The frame corresponds to the box  $[-6, 6] \times [-6, 6]$ , the sub-paving  $\Delta\mathbb{X}$  in yellow contains the boundary of  $\mathbb{X}$ , whereas  $\mathbb{X}^-$  is represented in red, and the rest is in cyan. Also,  $\mathbb{X}^+ = \Delta\mathbb{X} \cup \mathbb{X}^-$  and since  $\mathbb{X}^- \subset \mathbb{X}$ , if  $\mathbb{X}^-$  is non-empty then  $\mathbb{X}$  is non-empty, and similarly if  $\mathbb{X}^+$  is empty then  $\mathbb{X}$  is empty.

Set operations such as addition, intersection, inverse image, etc. can be approximated by sub-paving operations.

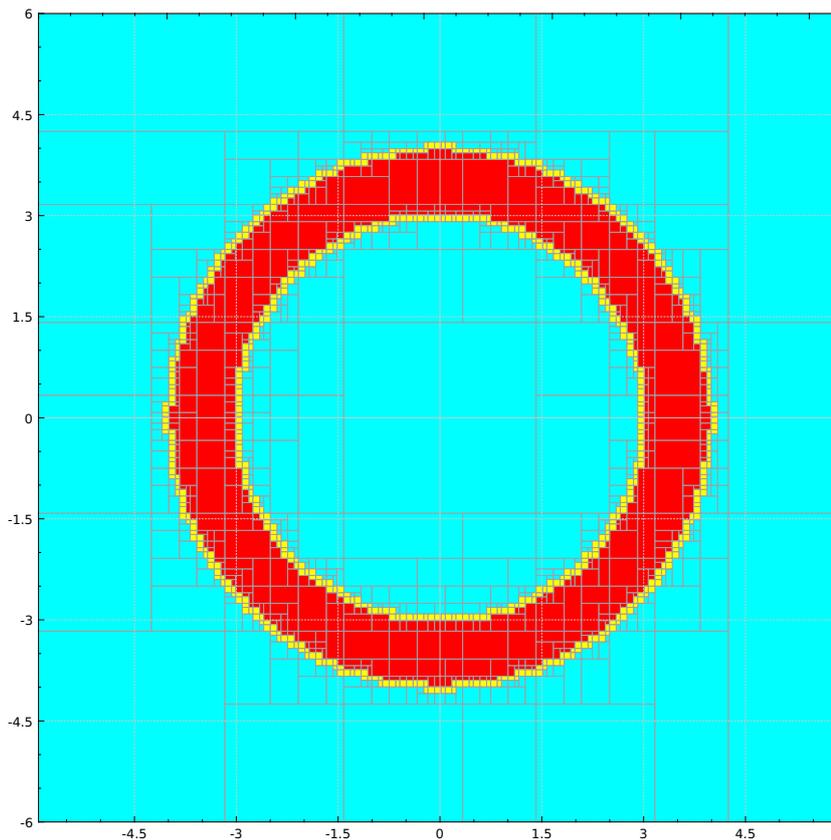


Figure 3.8: Sub-pavings characterizing a ring.

### 3.5.2 Inclusion Function

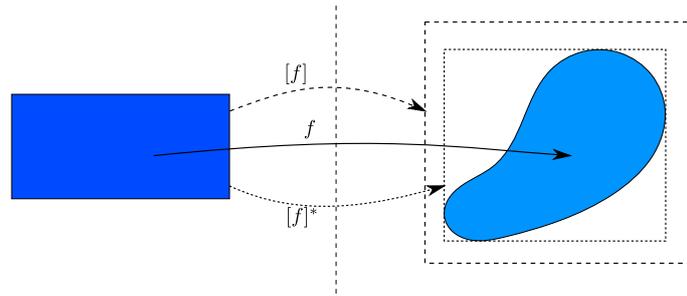
Let  $f$  be a function from  $\mathbb{R}^n \rightarrow \mathbb{R}^m$ . The interval function  $[f]$  from  $\mathbb{IR}^n \rightarrow \mathbb{IR}^m$  is an *inclusion function* of  $f$  if

$$\forall [\mathbf{x}] \in \mathbb{IR}^n, f([\mathbf{x}]) \subset [f]([\mathbf{x}]). \quad (3.34)$$

We define the following properties for inclusion functions

$[f]$ is <i>monotonic</i> if	$([\mathbf{x}] \subset [\mathbf{y}]) \Rightarrow ([f]([\mathbf{x}]) \subset [f]([\mathbf{y}]))$
$[f]$ is <i>minimal</i> if	$\forall [\mathbf{x}] \in \mathbb{IR}^n, [f]([\mathbf{x}]) = [f]([\mathbf{x}])$
$[f]$ is <i>thin</i> if	$\forall \mathbf{x} \in \mathbb{R}^n, [f](\{\mathbf{x}\}) = f(\{\mathbf{x}\})$
$[f]$ is <i>convergent</i> if	$\lim_{k \rightarrow \infty} w([\mathbf{x}](k)) = 0 \Rightarrow \lim_{k \rightarrow \infty} w([f]([\mathbf{x}](k))) = 0$

The Figure 3.9 illustrates these notions in the case where  $n = m = 2$ .



**Figure 3.9:** Inclusion functions  $[f]$  and *minimal* inclusion function  $[f]^*$ .

### 3.5.3 SIVIA

Let  $\mathbf{f}$  be a function from  $\mathbb{R}^n \rightarrow \mathbb{R}^m$  and let  $\mathbb{Y}$  be a subset of  $\mathbb{R}^m$ . *Set inversion* is the characterization of

$$\mathbb{X} = \{\mathbf{x} \in \mathbb{R}^n | \mathbf{f}(\mathbf{x}) \in \mathbb{Y}\} = \mathbf{f}^{-1}(\mathbb{Y}). \quad (3.35)$$

For any  $\mathbb{Y} \subset \mathbb{R}^m$  and for any function  $\mathbf{f}$  admitting a convergent inclusion function  $[f](\cdot)$ , two sub-pavings  $\mathbb{X}^-$  and  $\mathbb{X}^+$  such that

$$\mathbb{X}^- \subset \mathbb{X} \subset \mathbb{X}^+ \quad (3.36)$$

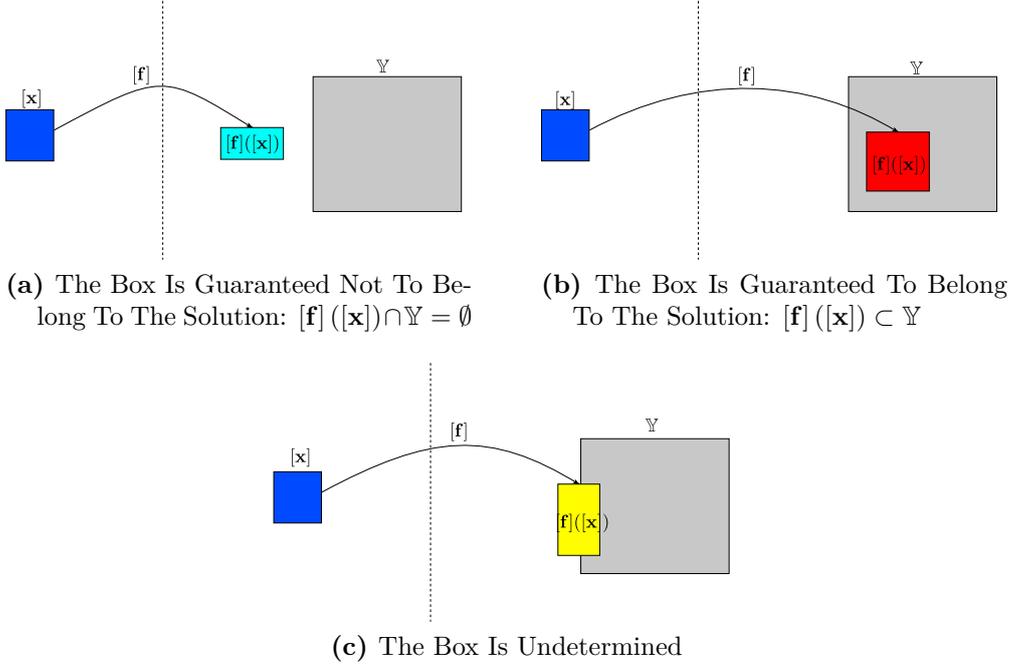
can be obtained with the SIVIA algorithm, to be described, by choosing the following inclusion test  $[t_f]$  as defined below.

$$[t_f]([\mathbf{x}]) = \begin{cases} 1 & \text{if } [f]([\mathbf{x}]) \subset \mathbb{Y} \\ 0 & \text{if } [f]([\mathbf{x}]) \cap \mathbb{Y} = \emptyset. \\ [0, 1] & \text{otherwise} \end{cases} \quad (3.37)$$

The SIVIA algorithm works as follow:

- An initial box  $[\mathbf{x}]$  is provided and is guaranteed to contain the solution. If no prior information about the set is available, the box is taken very large.
- The inclusion test  $[t_f]$  is performed on  $[\mathbf{x}]$ :
  - if  $[t_f]([\mathbf{x}]) = 0$ , *i.e.*  $[f]([\mathbf{x}]) \cap \mathbb{Y} = \emptyset$ , then the box  $[\mathbf{x}]$  is discarded as the image of  $[\mathbf{x}]$  by  $[f]$  is disjoint from  $\mathbb{Y}$ . It is then guaranteed that the box  $[\mathbf{x}]$  not to belong to the solution set, 3.10a.
  - else if  $[t_f]([\mathbf{x}]) = 1$ , *i.e.*  $[f]([\mathbf{x}]) \subset \mathbb{Y}$ , then the box  $[\mathbf{x}]$  is added to the sub-pavings of  $\mathbb{X}^-$  and  $\mathbb{X}^+$ . The box  $[\mathbf{x}]$  is then guaranteed to belong to the solution set, 3.10b.
  - else, the box  $[\mathbf{x}]$  is considered undetermined, 3.10c:

- \* if the width of the box  $[x]$  is smaller than a desired precision  $\epsilon$ , it is then added to the sub-pavings of  $\mathbb{X}^+$ .
  - \* else, the box  $[x]$  is bisected into two sub-boxes  $[x_1]$  and  $[x_2]$  on which SIVIA will run again.
- The algorithm is run recursively until all the boxes are either discarded, added to the solution set or considered too small to be evaluated.



**Figure 3.10:** Inclusion Test

The SIVIA pseudo-code is summarized in Algorithm 3.2. Initially, the sub-pavings  $\mathbb{X}^+$  and  $\mathbb{X}^-$  are empty and  $[x]$  is guaranteed to contain the solution.

In figure Figure 3.11, an example of the application of SIVIA is presented. Initially in the space  $[-6, 6] \times [-6, 6]$ , a robot is present at an unknown position. The robot measures a guaranteed distance  $[3, 4]$  from a beacon at a known position  $[1, 1]$ . The SIVIA algorithm is applied to the initial box  $[x_0] = [-6, 6] \times [-6, 6]$ . The algorithm outputs the following: the boxes filled with red represent the inner approximation  $\mathbb{X}^-$  of the solution. The boxes filled with yellow represent the undetermined boundary of the solution  $\Delta\mathbb{X}$ . The union of the red and yellow boxes represent the outer approximation of the solution  $\mathbb{X}^+$ . The sub-paving  $\mathbb{X}^+$  is guaranteed to contain all the solutions compatible with the observations. Finally the blue filled boxes represent the discarded boxes that are guaranteed not to contain any solution.

Applying now this algorithm to the previously mentioned CSP  $\mathcal{H}_{\text{sin}}$  of Figure 3.7, it is able to go beyond the wide interval  $[0, 2\pi]$  and output narrower sub-pavings around the three solutions without never finding the singletons  $\{0\}$ ,  $\{\pi\}$  and  $\{2\pi\}$ .

---

**Algorithm 3.2** SIVIA(in :  $[\mathbf{x}], \mathbf{f}, \mathbb{Y}, \epsilon$ ; out :  $\mathbb{X}^-, \mathbb{X}^+$ )

---

```

1: if ( $([\mathbf{f}]([\mathbf{x}]) \cap \mathbb{Y} = \emptyset)$ ) then
2:   return
3: else if ( $([\mathbf{f}]([\mathbf{x}]) \subset \mathbb{Y})$ ) then
4:    $\mathbb{X}^- = \mathbb{X}^- \cup [\mathbf{x}]$ 
5:    $\mathbb{X}^+ = \mathbb{X}^+ \cup [\mathbf{x}]$ 
6:   return
7: else
8:   if ( $w([\mathbf{x}]) < \epsilon$ ) then
9:      $\mathbb{X}^+ = \mathbb{X}^+ \cup [\mathbf{x}]$ 
10:    return
11:  else
12:    bisect  $[\mathbf{x}]$  into  $[\mathbf{x}_1]$  and  $[\mathbf{x}_2]$ 
13:    SIVIA(in :  $[\mathbf{x}_1], \mathbf{f}, \mathbb{Y}, \epsilon$ ; out :  $\mathbb{X}^-, \mathbb{X}^+$ )
14:    SIVIA(in :  $[\mathbf{x}_2], \mathbf{f}, \mathbb{Y}, \epsilon$ ; out :  $\mathbb{X}^-, \mathbb{X}^+$ )

```

---

Therefore a better algorithm using both the idea of SIVIA and the power of contractors has been developed.

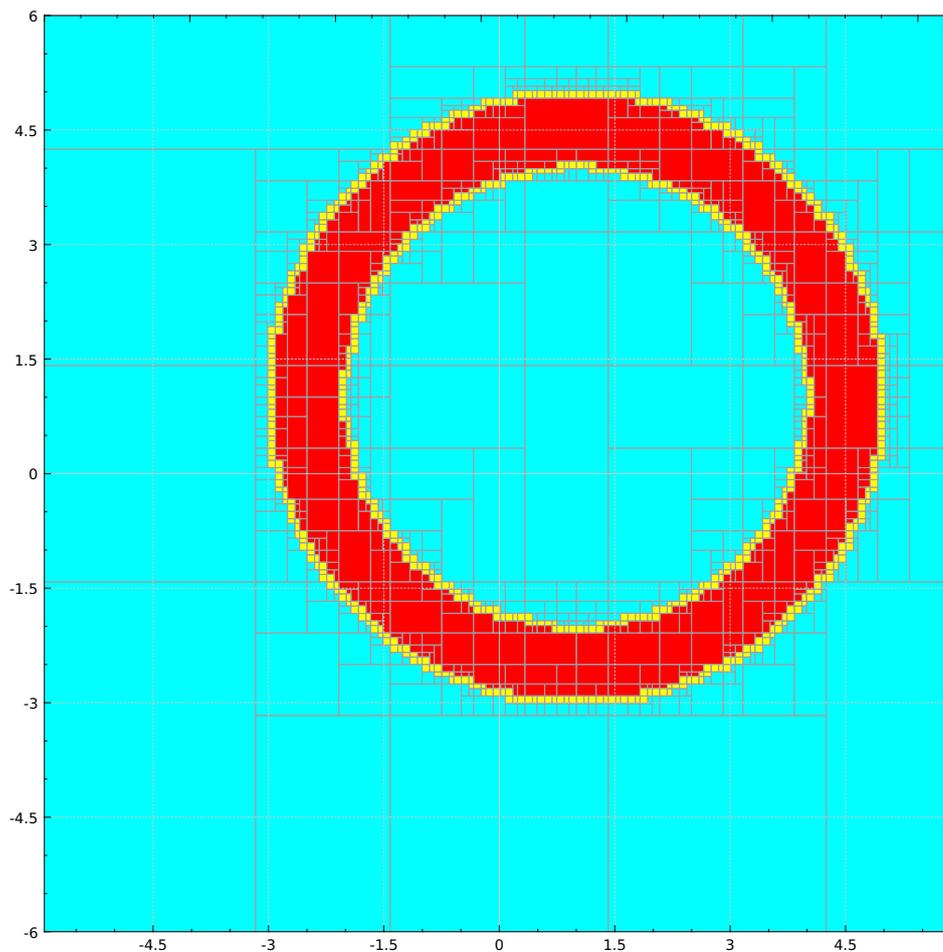
### 3.5.4 SIVIA With Contractors

To characterize a set  $\mathbb{X} \subset \mathbb{R}^n$ , bisection-based algorithms, like SIVIA, need to bisect all the boxes in all directions. For instance, for  $n = 20$ , bisecting a box in all directions generates  $2^{20} = 1048576$  boxes. Therefore, for high-dimensions, bisections should be avoided as much as possible.

An alternative SIVIA algorithm applies a contractor  $\mathcal{C}$  on the box  $[\mathbf{x}]$  before bisecting it. For a given number of bisections, the resulting sub-pavings are more precise than those obtained with a simple SIVIA.

Any contractor that characterizes the CSP  $\mathcal{H}$  can be used. For instance, we propose to use two contractors  $\mathcal{C}_{in}$  and  $\mathcal{C}_{out}$  to characterize the sets  $\mathbb{X}^-$  and  $\mathbb{X}^+$  respectively. A pseudo-code describing this SIVIA algorithm using contractor is described in Algorithm 3.3. Notice that the contractor  $\mathcal{C}_{in}$  characterizes the set  $\mathbb{X}^-$  by removing it from the given box, therefore the boxes  $[\mathbf{x}_{in}] \in \mathbb{X}^-$  are the result of the deprivation between the original box and the output box by the contraction, *i.e.*  $[\mathbf{x}_{origin}] \setminus [\mathbf{x}_{contracted}]$  as shown in the algorithm.

Figure Figure 3.12 shows the output of a SIVIA algorithm using contractors on the same example as below, where a robot measures a distance of  $[3, 4]$  from a beacon at a position  $[1, 1]$ . Notice that the sub-pavings obtained are not regular anymore. In this particular case, every box touches the border  $\Delta\mathbb{X}$  showing that the contractors that has been used are minimal (also called optimal).



**Figure 3.11:** Classic SIVIA. In The Space  $[-6, 6] \times [-6, 6]$ , A Robot Measures A Distance  $[3, 4]$  From One Beacon At A Known Position  $[1, 1]$ .

In Figure 3.13, a comparison between the three previous techniques to solve the CSP  $\mathcal{H}_{\text{sin}}$  is given. In all three cases, the algorithm is given the initial box  $[\mathbf{x}] = [-1, 7]$ . For the first case, 3.13a, only an optimal contractor has been used, there for the out  $\mathbb{X}^+ = \{[0, 2\pi]\}$ . The classical SIVIA, 3.13b, took 45 iterations to find  $\mathbb{X}^+ = \{[6.25, 6.3125], [3.125, 3.1875], [0, 0.0625], [-0.0625, 0]\}$ . Finally, SIVIA with an optimal contractor, figure 3.13c, took only 5 iterations to find  $\mathbb{X}^+ = \{\{2\pi\}, \{\pi\}, \{0\}\}$ , hence the importance of using SIVIA with contractors.

---

**Algorithm 3.3** CSIVIA(in :  $[\mathbf{x}], \mathcal{C}_{in}, \mathcal{C}_{out}, \epsilon$ ; out :  $\mathbb{X}^-, \mathbb{X}^+$ )
 

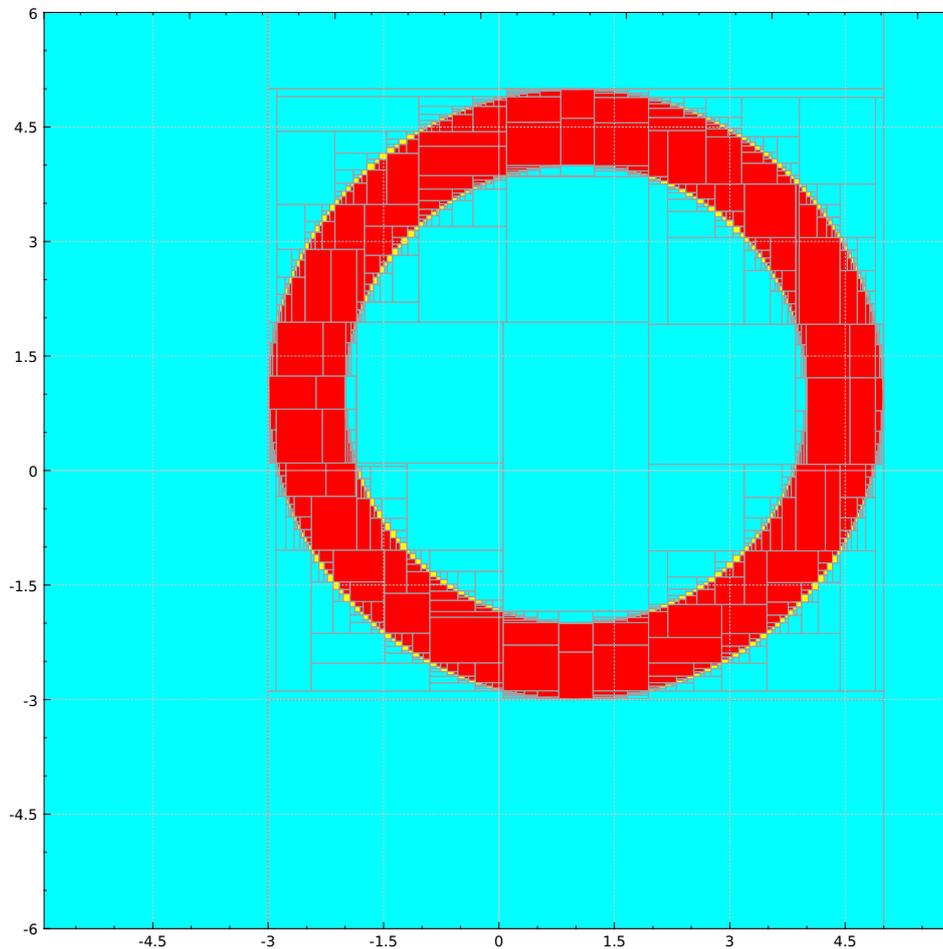
---

```

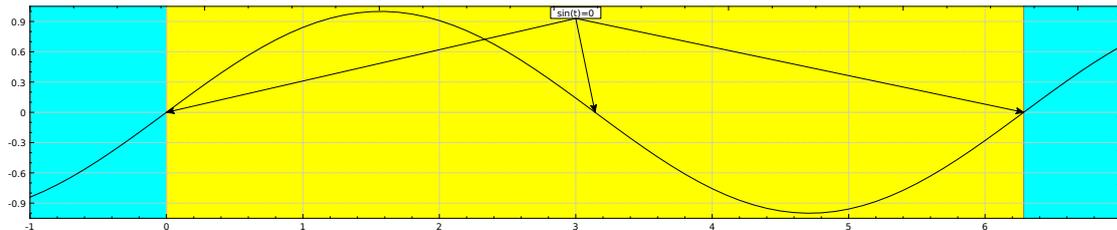
1:  $[\mathbf{tmp}] := [\mathbf{x}]$ 
2:  $[\mathbf{x}] := \mathcal{C}_{in}([\mathbf{x}])$ 
3:  $\mathbb{X}^- = \mathbb{X}^- \cup ([\mathbf{tmp}] \setminus [\mathbf{x}])$ 
4:  $[\mathbf{x}] := \mathcal{C}_{out}([\mathbf{x}])$ 
5: if ( $w([\mathbf{x}]) < \epsilon$ ) then
6:    $\mathbb{X}^+ = \mathbb{X}^+ \cup [\mathbf{x}]$ 
7:   return
8: else
9:   bisect  $[\mathbf{x}]$  into  $[\mathbf{x}_1]$  and  $[\mathbf{x}_2]$ 
10:  CSIVIA(in :  $[\mathbf{x}_1], \mathcal{C}_{in}, \mathcal{C}_{out}, \epsilon$ ; out :  $\mathbb{X}^-, \mathbb{X}^+$ )
11:  CSIVIA(in :  $[\mathbf{x}_2], \mathcal{C}_{in}, \mathcal{C}_{out}, \epsilon$ ; out :  $\mathbb{X}^-, \mathbb{X}^+$ )

```

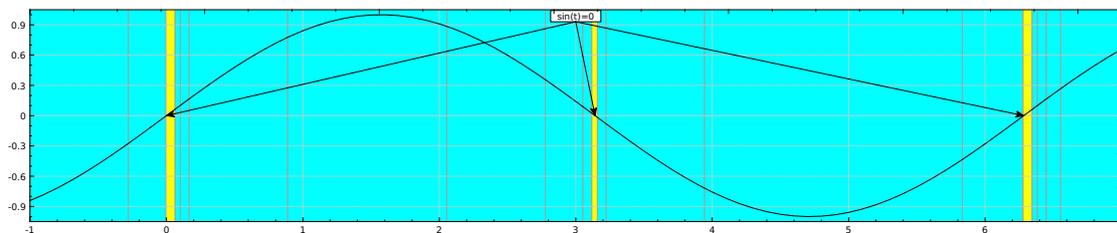
---



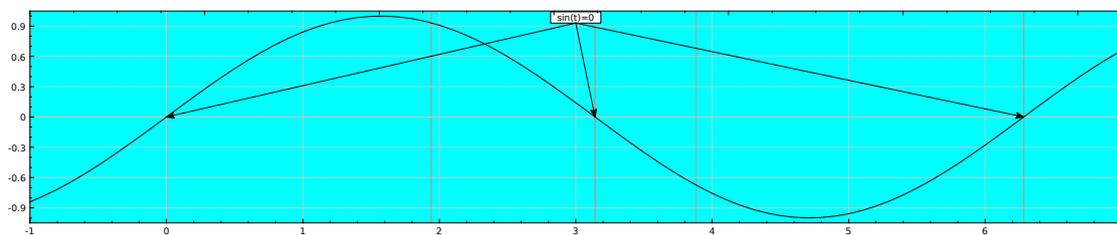
**Figure 3.12:** SIVIA using contractors. In The Space  $[-6, 6] \times [-6, 6]$ , A Robot Measures A Distance  $[3, 4]$  From One Beacon At A Known Position  $[1, 1]$ .



(a) Using Only A Contractor.



(b) Using A Classic SIVIA.

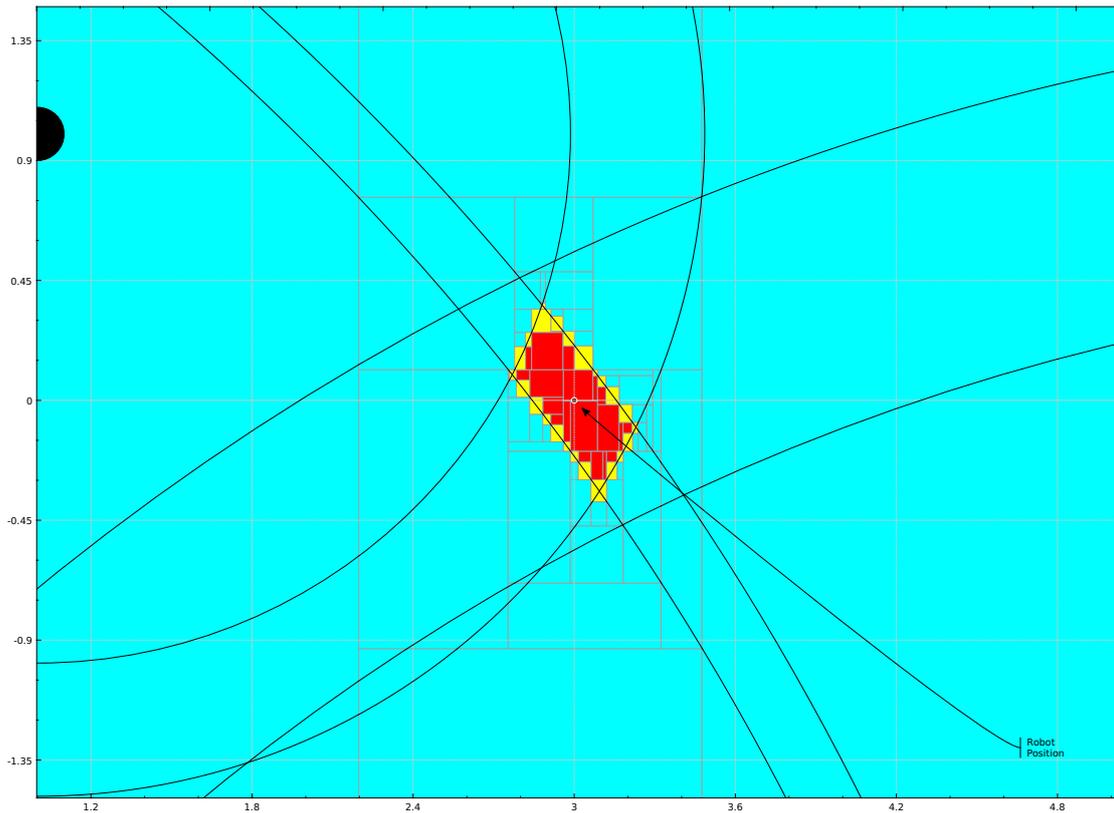


(c) Using SIVIA With Contractors.

**Figure 3.13:** Solving The CSP  $\mathcal{H}_{\sin}$ .

### 3.6 Conclusion

In this chapter, we presented the set-membership tools that will be used throughout this thesis. Foundations of interval arithmetic and parameter estimation given measurements with bounded errors have been laid down. Figure 3.14 shows the solution given by a SIVIA using contractors on the triangulation problem presented in the introduction. In a space  $[-20, 20] \times [-20, 20]$ , a robot measures three distances  $d_1$ ,  $d_2$ , and  $d_3$  from three different beacons at the position  $(1, 1)$ ,  $(7, -8)$ , and  $(-5, -6)$ .



**Figure 3.14:** Triangulation Using Interval Analysis.

In the next chapter, we apply these techniques to solve a localization problem where an underwater vehicle records an acoustic signal coming from a single beacon and tries to locate itself using this information and fusing it with its on-board sensors data.

# 4 Phase Based Localization

## 4.1 Introduction

In the previous chapters, we have established different localization systems used for an underwater environment. Moreover, we have introduced the foundations to solve a localization problem using the set-membership approach. In this chapter, we present a new system to localize robots underwater and the enabling algorithm based on interval analysis.

This underwater localization system is based on acoustic communications due to the constraints on electromagnetic waves penetration in water. The novelty of the system is that it fuses the on-board sensors with minimal information coming from the stationary beacon. At a fixed position, the beacons emits a continuous acoustic sine wave to be recorded by the moving vehicles. The vehicles within the range of receiving the acoustic signal can then determine their position from the beacon based on their proprioceptive measurements and the phase-shift between the recorded signal and the original signal using synchronized clocks. The data is fused using interval analysis techniques described in the previous chapter.

Proprioceptive sensors are widely used in underwater robotics as the main sensors because of their accuracy. Still, to get a position of the vehicles, proprioceptive sensors are useless without either an external help or a prior knowledge of the initial position. In the latter case, the errors on the measurements can lead to very large position estimation that can be optimized notably using interval analysis [29].

In this chapter, we first explain the motivations behind the development of such a system, in section 4.2. The problem statement is explained in section 4.3. The algorithm enabling the localization system is described in section 4.4. The section 4.5 shows the results of simulation and a possible implementation on real targets. Finally, the section 4.6 concludes this chapter.

The drawing conventions used for boxes in the previous chapter are used in this chapter too, *i.e.* red, yellow, and cyan or white boxes represent respectively the inner sub-paving  $\mathbb{X}^-$ , the boundary  $\Delta\mathbb{X}$  and complementary of  $\mathbb{X}$ .

## 4.2 Motivation

Many underwater localization systems have been developed over the years as we previously exposed in a previous chapter. However, no system is widely used yet as the GPS for instance. Therefore, there is still a possibility to develop a relatively simple system to use for the increasing amount of underwater vehicles that are being used for multiple purposes. Nevertheless, most of the existing systems have some flaws that does not make them good candidates for a globally used system. Hence the motivations behind this particular system: scalability, minimum communication, robustness against kidnapping, no prior assumptions, ability to reconstruct the path.

### 4.2.1 Scalability

One of the main issues of existing underwater localization systems is the scalability issue. Most can only track and localize a finite number of vehicles, which makes it difficult for multi-vehicle operations. For instance, the ultra-short baseline systems (USBL) and the long baseline systems (LBL) are limited to number of frequencies and IDs that the system can identify. Most of these limitations come from the fact that these systems rely on a two-way communication between the target and the base. The base or the target, depending on the system, replies to a previously emitted signal in order to estimate the distance from the ToF. This factor therefore limits the number of either the targets or the refresh rate as a time division multiple access (TDMA) method is usually required.

### 4.2.2 Minimum Communication

For a system to be scalable, it is then necessary for it to require as less communication as possible. One of the success factor of the GPS, for example, is the capacity of any device, capable of receiving and decoding the data streams from the GPS satellites, to then find its position with a decent accuracy without having to send data back to the satellites or any other receiver. Therefore, the receivers are more affordable and consume very little power. A key to a reliable system can then use as minimum communication as possible and preferably passive.

### 4.2.3 Kidnapping

The advancements made in proprioceptive sensors for underwater systems enable some underwater vehicles to only rely on these sensors. Moreover these advancements are actually fueled by the previously stated problems with communication and scalability of the exteroceptive sensors. However, the kidnapping effect [69], in case of strong currents or power loss, can have huge consequences on the estimated position.

### 4.2.4 Unknown Starting Conditions

Also, most of the algorithms require a prior estimation of the position. In case of a continuous mission using estimator based on proprioceptive sensors, like Extended Kalman Filter (EKF) [70] or particle filters [71], the vehicle gets a position fix before starting the mission and/or at the end of the mission. Without a fix, the vehicle will never be able to find its position and will at most be able to track its position according to the starting point with a forever growing error. This is similar to a kidnapping effect, where the vehicle is totally lost in the space and can no longer estimate its position.

### 4.2.5 Path Reconstruction

Some missions may further require the capacity to replay the mission at some point to further review the data. In missions like mine-hunting or geophysical surveys, recalling the exact position of the vehicle is crucial for the mission to position the mine or the geophysical characteristics desired. Therefore, the system should be able to not only provide the position of the vehicle at an instant but also to reconstruct the full trajectory of the vehicle.

## 4.3 Problem Statement

Consider an environment where a swarm underwater vehicle is on a mission. Each vehicle is equipped with proprioceptive sensors, speed and orientation, and can record an acoustic signal coming from a beacon at a fixed position.

Because the system proposed should be scalable and require no communication between the vehicles and the beacon as it will be detailed later, the following example take into account only one vehicle. However the same process can be applied to as many vehicles that are within the range of the beacon's signal.

### 4.3.1 Motion Model

Let's use a simplified model of an underwater vehicle. Let  $\mathbf{x}=[x, y, z, \theta]^T$  be the state vector describing the pose of the vehicle, where  $\theta$  is the heading of the vehicle measured by the on-board sensor and  $x, y$  are the coordinates of the vehicle according to a reference system centered on the beacon, and  $z$  be the depth of the vehicle, Figure 4.1.

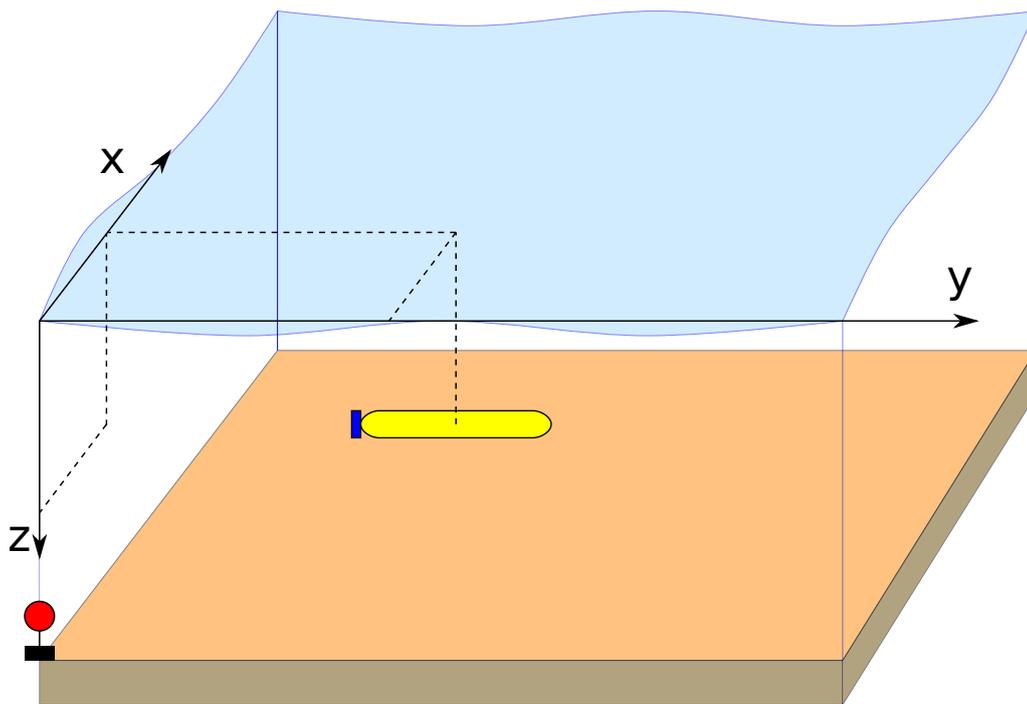


Figure 4.1: Vehicle pose in the environment.

The motion model is

$$\dot{\mathbf{x}} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} v \cdot \cos \theta \\ v \cdot \sin \theta \\ v_z \\ \omega_\theta \end{bmatrix}, \quad (4.1)$$

where  $v$  is the speed of the vehicle over the horizontal plan measured by the on-board sensor,  $v_z$  is the diving rate, and  $\omega_\theta$  is the rotation rate.

A more complicated model could be used, however compared to the observation errors, this model errors can be neglected.

### 4.3.2 Phase-Lock Algorithms

For this system to be operational, each vehicles clock must be synchronized to the beacon clock modulo  $\omega_s = 2\pi f$ , where  $f$  is the frequency of the emitted acoustic signal. This can be achieved easily by using a classical phase-lock loop (PLL) algorithm at the moment of deploying the vehicles. Example of PLL algorithms are given in the annex ??.

The beacon emits a source signal  $s_{\text{source}}$  which is a pure tone at a frequency  $f$ :

$$s_{\text{source}} = \sin(\omega_s t), \quad (4.2)$$

where  $t$  is the time of provided by the synchronized clock and  $\omega_s = 2\pi f$  is the angular velocity.

A vehicle, at a distance  $d$  from the beacon, receives a  $\delta_\psi$  phase-shifted signal  $s$  described by

$$s = \sin(\omega_s t + \delta_\psi), \quad (4.3)$$

where  $\delta_\psi = -\omega_s \frac{d}{c}$ , and  $c$  is the celerity of the acoustic wave in the environment. Therefore the PLL algorithm will provide the phase-shift  $\delta_\psi$  that embeds information about the position of the vehicle in the space. We will consider also that the vehicle movements only introduces a phase shift but no Doppler effect.

Consequently, the recorded signal  $s$  is rewritten

$$s = \sin\left(\omega_s \left(t - \frac{\sqrt{x^2 + y^2 + z^2}}{c}\right)\right). \quad (4.4)$$

This signal combined with proprioceptive data provides enough information to locate the vehicle as we will demonstrate in the next section. Both equations Equation 4.1 and Equation 4.4 will be cast into a CSP and used to inverse the system to retrieve the vehicle position.

## 4.4 Algorithm Description

### 4.4.1 Casting the problem into a CSP

First, as we would like to solve the system

$$\mathcal{H}_{\text{PBL}} : \begin{cases} \dot{\mathbf{x}} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} v \cdot \cos \theta \\ v \cdot \sin \theta \\ v_z \\ \omega_\theta \end{bmatrix} \\ s = \sin\left(\omega_s \left(t - \frac{\sqrt{x^2 + y^2 + z^2}}{c}\right)\right) \end{cases} \quad (4.5)$$

in a discrete time space, Euler discretization will be used to integrate from the continuous time space to a discrete time space.

The system Equation 4.5 is rewritten in a discrete time with a sampling rate  $h$ , small enough to assume that all variables are constant within this interval, at an

instant  $k$  as  $\mathcal{H}_{\text{PBL}}(k)$ :

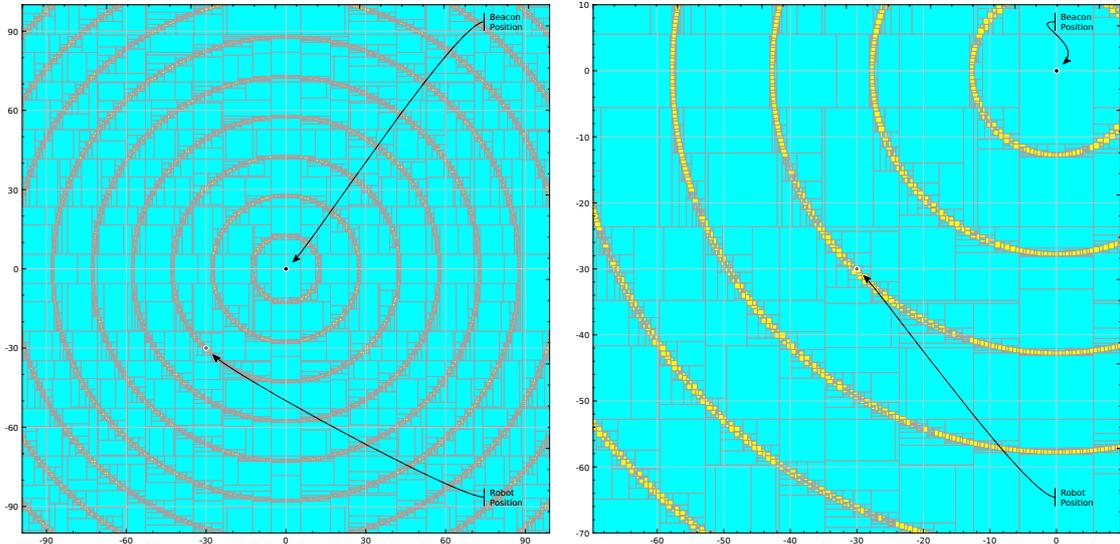
$$\mathbf{x}_k = \mathbf{x}_{k-1} + h \cdot \begin{bmatrix} v_{k-1} \cdot \cos \theta_{k-1} \\ v_{k-1} \cdot \sin \theta_{k-1} \\ v_{z_{k-1}} \\ \omega_{\theta_{k-1}} \end{bmatrix} \quad (4.6a)$$

$$s_k = \sin \left( \omega_s \left( t_k - \frac{\sqrt{x_k^2 + y_k^2 + z_k^2}}{c} \right) \right) \quad (4.6b)$$

The vehicle measures at each time sample its horizontal speed  $v_k$ , its rotation rate  $\omega_{\theta_k}$ , and its vertical speed  $v_{z_k}$ . An additional sensor allows the recording of the signal  $s_k$ .

The problem can then be cast into a CSP  $\mathcal{H}_{\text{PBL}}(k)$  with the following inputs:  $[v_k]$ ,  $[\omega_{\theta_k}]$ ,  $[v_{z_k}]$ , and  $[s_k]$ . Intervals were taken around the measurements to take the errors into account.

Applying a SIVIA to the CSP will unfortunately not provide enough precision for a reliable solution as shown in the Figure 4.2. Therefore a better approach should be developed.



(a) Set Of Solutions Of  $\mathcal{H}_{\text{PBL}}(k)$  Where The Robot Is At  $(-30, -30)$  And The Space Is  $[-100, 100]^2$  (b) The Same Set Of Solutions Of  $\mathcal{H}_{\text{PBL}}(k)$ , Zoomed Around The Vehicle's Position.

**Figure 4.2:** SIVIA on  $\mathcal{H}_{\text{PBL}}(k)$ .

### 4.4.2 Algorithm

For a better precision of the vehicle position, another method is proposed with this system. Instead of only considering the position of the vehicle at the sample  $k - 1$  and the measurements made at the sample  $k$ , all measurements from the sample  $k_i$  to  $k_f$  are considered. To do so, we propose to mix the SIVIA algorithm of solving CSPs with the concept of tubes and contractors over tubes. This algorithm would enable contraction over time samples, using an approach similar to non causal estimator [72].

The algorithm described in Algorithm 4.1 uses the initial tube  $[\mathbf{x}_0](t)$  as an input along with :

- the contractor  $\mathcal{C}_{\text{motion}}$  that contracts according to the motion of vehicle from a time sample to the next, from the equation Equation 4.6a,
- the contractor  $\mathcal{C}_{\text{observation}}$  that contracts according to the measurements of the vehicle, *i.e.* the recorded signal, from the equation Equation 4.6b. It takes as a second parameter a measurement.
- The tube of measurements  $[s](t)$ .

As the time domain is now considered discrete instead of continuous, tubes can be set as vectors of interval with each component of the vector being the value of the tube at defined moment  $t_k$ . Therefore, tubes will be noted with sample indexes:  $[\mathbf{x}]_{i,f}$  represents a vector of intervals, *i.e.* a discrete tube,  $[\mathbf{x}](t)$  from the sample  $t_i$  to  $t_f$  with  $i < f$ . Later, we will explain why it is useful to take a limited number of samples instead of the whole tube. Moreover,  $[\mathbf{x}]_j$  will represent the  $j^{\text{th}}$  component of  $[\mathbf{x}]_{i,f}$ , *i.e.* the value of the tube at the instant  $t_j$ :  $[\mathbf{x}](t_j)$ .

The algorithm first puts the tube to be contracted  $[\mathbf{x}_0]_{i,f}$  in a stack where the not yet processed tubes are stored. Then, while the stack is not empty, a tube  $[\mathbf{x}]_{i,f}$  is pulled from the stack and contracted. First, the tube  $[\mathbf{x}]_{i,f}$  is contracted using the contractor  $\mathcal{C}_{\text{motion}}$  that will contract each component  $[\mathbf{x}]_j$  based on the previous tube element  $[\mathbf{x}]_{j-1}$ . This first part is where the constraints propagate forward in time, *i.e.* from the initial moment  $i$  to the final moment  $f$ , therefore it is called *Forward-Time Propagation*. The contractor  $\mathcal{C}_{\text{observation}}$  is applied afterwards using the observations  $[s]_{i,f}$ . Conversely, the second part where the constraints are propagated from the final moment  $f$  to the initial moment  $i$  through the contraction using  $\mathcal{C}_{\text{motion}}^{-1}$  is called *Backward-Time Propagation*. Finally, just like a SIVIA algorithm, if the tube meets the precision  $\epsilon$  then it is added to the solution set  $\mathbb{X}$ , otherwise, the tube is bisected and the resulting tubes are pushed back to the stack to be processed.

---

**Algorithm 4.1** TubeCSIVIA(**in** :  $[\mathbf{x}_0]_{i,f}, \mathcal{C}_{\text{motion}}, \mathcal{C}_{\text{observation}}, [s]_{i,f}, \epsilon$ ; **out** :  $\mathbb{X}$ )

---

```

1: push  $[\mathbf{x}_0]_{i,f}$  to stack
2: while(stack not empty):
3:   pull  $[\mathbf{x}]_{i,f}$  from stack
   // Forward-Time Propagation
4:    $[\mathbf{x}]_{i,f} := \mathcal{C}_{\text{motion}}([\mathbf{x}]_{i,f})$ 
5:    $[\mathbf{x}]_{i,f} := \mathcal{C}_{\text{observation}}([\mathbf{x}]_{i,f}, [s]_{i,s})$ 
   // Backward-Time Propagation
6:    $[\mathbf{x}]_{i,f} := \mathcal{C}_{\text{motion}}^{-1}([\mathbf{x}]_{i,f})$ 
7:   if( $w([\mathbf{x}]_{i,j}) < \epsilon_{\text{precision}}$ ):
8:      $\mathbb{X} = \mathbb{X} \cup [\mathbf{x}]_{i,j}$ 
9:     continue
10:  else:
11:    bisect  $[\mathbf{x}]_{i,j}$  into  $[\mathbf{x}_1]_{i,j}$  and  $[\mathbf{x}_2]_{i,j}$ 
12:    push  $[\mathbf{x}_1]_{i,j}$  and  $[\mathbf{x}_2]_{i,j}$  into stack

```

---

## 4.5 Simulation & Implementation

We propose to illustrate this localization system first with scenarios on a simulated environment with very few disturbances and analyze each result. The second part will focus on the different issues that the system could face and to solve them when possible.

For this subsection, we will suppose that the source of the signal  $s$  is always positioned at the same spot at the origin of the frame. Also, we will only be considering the planar problem to better illustrate the examples in two-dimensional drawings.

### 4.5.1 Almost-Perfect Environment

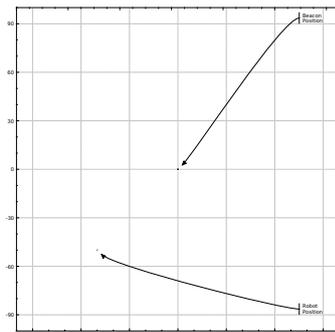
#### 4.5.1.1 Static Robot

Suppose a vehicle  $A$  at a stationary position from the source, Figure 4.3.

The Figure 4.4 shows that no matter how long the measurements tube is, the solution set is never satisfying. Actually, the fact that the vehicle is being stationary cancels the effect of the contractors  $\mathcal{C}_{\text{motion}}$ , therefore the resulting solution set is equivalent to the solution set of a SIVIA algorithm applied used the contractor  $\mathcal{C}_{\text{observation}}$ .

#### 4.5.1.2 Straight Path

Suppose now that the vehicle is moving following a straight line, Figure 4.5.



**Figure 4.3:** Stationary Vehicle.

The Figure 4.6 shows different solutions provided by the algorithm depending on the size of the tube, *i.e.* the number of recorded samples.

Notice that the system is incapable of deciding which of the two possible solution sets is the true location of the vehicle, 4.6c and 4.6d. This is due to the symmetry introduced by the square and square root functions inside the phase shift expression. Therefore, for this system to be effective, the vehicle should preferably perform a symmetry breaking maneuver at some point during his path.

#### 4.5.1.3 Asymmetrical Path

In order to solve the ambiguity resulting of the distance expression in the phase-shift expression, we propose that the vehicle performs a symmetry breaking maneuver during its mission. Similar technique is used by the military in anti-submarine warfare to solve what is commonly called left-right ambiguity [73].

Suppose now that the vehicle follows the path in Figure 4.7, the beginning of a lawnmower pattern.

The Figure 4.8 shows the results of the algorithm on a vehicle following the path in Figure 4.7 depending on the number of measurements taken into account. In figure 4.8a and 4.8b, the algorithm is not capable of converging towards the position of the vehicle. This is due to the fact that the vehicle is considered to only follow the straight line from its buffered data. Notice that in 4.8c, even though the vehicle performed a turn it is not enough to solve all the ambiguities. But finally, with enough data, the algorithm converges towards the unique set of solutions.

Note that the vehicle could perform a zigzag on its path, like the left-right maneuver dictates, to faster converge towards the solution. However, this case has been selected to show that just breaking the symmetry does not lead to the convergence of the solution right away.

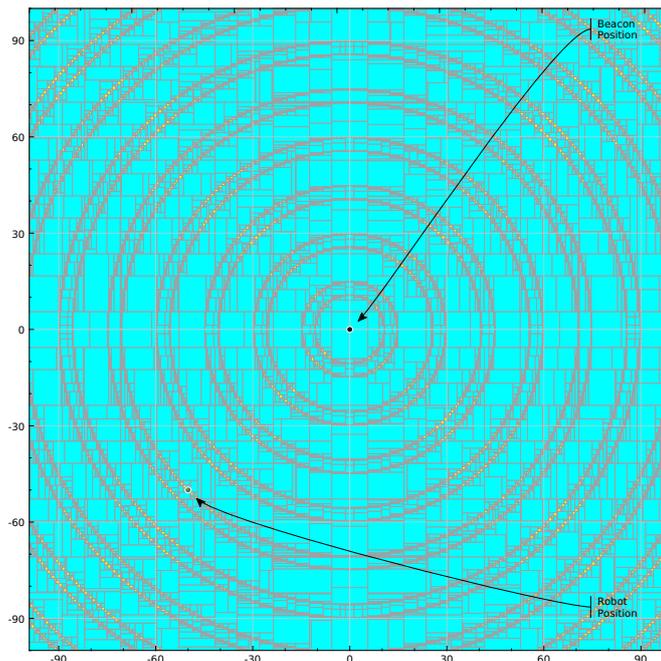


Figure 4.4: Stationary Vehicle Solution.

## 4.5.2 Real-World Conditioned Environment

In the previous subsection, we showed the performance of the algorithm in different situations and we also shown that it performs best when the vehicle is in movement and following an asymmetrical path. Nevertheless, both the observations and the state estimation were noise-free, which is never the case in real-life. In this subsection, noise will be introduced in the measurements and state vector to show the behavior of the algorithm. Finally, the case where the vehicle is kidnapped is tested and an improvement for the algorithm is proposed based on sliding horizon techniques.

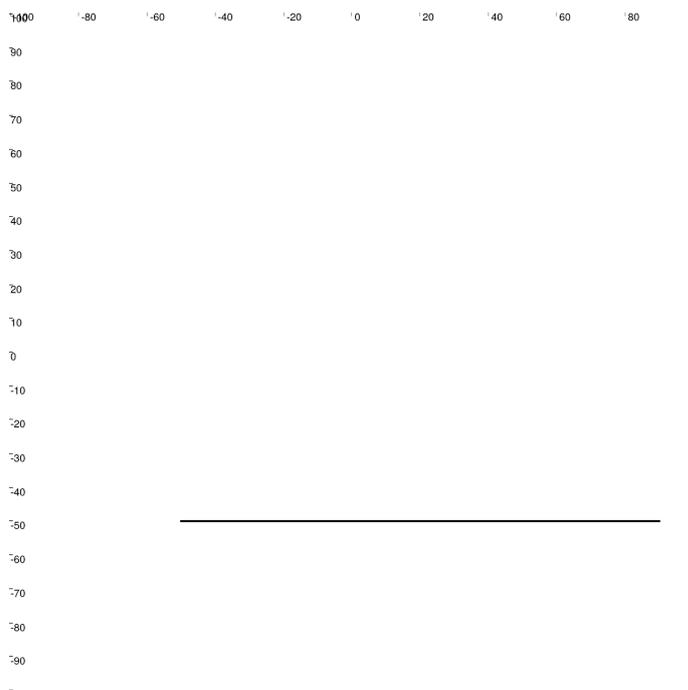
### 4.5.2.1 Noisy Exteroceptive Sensors

In this first scenario, noise is introduced in the measurement of the received signal  $s$ . The new input  $s_{\text{noisy}}$  to the algorithm is described by

$$s_{\text{noisy}} = \sin(\omega_s t + \delta_\psi + [\epsilon_\psi]) + [\epsilon_s], \quad (4.7)$$

where  $[\epsilon_\psi]$  is the noise interval on the phase-shift, and  $[\epsilon_s]$  is the noise interval on the signal itself. The Figure 4.9 shows the difference between the noiseless signal  $s$  and the different type of noise on the noisy recorded signal  $s_{\text{noisy}}$ .

For one of the three cases, a simulation is made using the same path as in Figure 4.7. The results of the algorithm are shown in Figure 4.10, Figure 4.11, and Figure 4.12,



**Figure 4.5:** Path Of The Vehicle (7000 steps).

for a signal with noise only on the phase-shift  $[\epsilon_\psi]$ , a signal with noise on the recording  $[\epsilon_s]$ , and a signal with both the previous noises respectively.

Notice that the algorithm hardly finds any inner sub-pavings  $\mathbb{X}^-$ . It also takes more time than previously to converge towards the right solution.

### 4.5.2.2 Noisy Proprioceptive Sensors

Let's now introduce a noise  $[\epsilon_x]$  on the estimation as follows:

$$\dot{\mathbf{x}} \in \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \dot{\theta} \end{bmatrix} + [\epsilon_x] = \begin{bmatrix} v \cdot \cos \theta \\ v \cdot \sin \theta \\ v_z \\ \omega_\theta \end{bmatrix} + [\epsilon_x]. \quad (4.8)$$

Such noise can be due to bad proprioceptive sensors reading speed and gyroscopic data of the vehicle, which is usually common in low-cost vehicles.

In Figure 4.13, the result of the algorithm is shown at different moments. Note that the algorithm no longer finds the inner sub-pavings  $\mathbb{X}^-$  and the outer approximation is even wider.

In such cases, the vehicle autonomy should be aware of the performance of the algorithm, for instance by evaluating the envelop of the solution set, and decide to take necessary maneuvers accordingly.

### 4.5.2.3 Kidnapping

In this last test case, we simulate a kidnapping effect on the vehicle by deliberately breaking the non-holonomic path that the vehicle should theoretically follow. This can eventually happen when the vehicle is subject to strong current gusts that pushes the vehicle away from its predefined path.

The Figure 4.14 shows the path of a vehicle that has been subject of strong gusts that modified its path almost instantly.

Applying the algorithm to the recordings first produces similar results to the previous scenarios as shown in figure 4.15a. However, once the kidnapping occurs, figure 4.15b, the algorithm misses the real solution and tracks a set that solves the equations. As long as the measurements are coherent with proprioceptive measurements, it will not be able to detect the anomaly introduced with the kidnapping, figure 4.15c. However, as soon as an exteroceptive measurement is outside of the awaited interval, figure 4.15d, the algorithm supposes that no set can be considered a solution of the CSP  $\mathcal{H}_{\text{PBL}}$ .

In order to solve this issue, we propose to change the depth of the history until the anomaly is outside of the processed window. For instance, if the kidnapping happens at an instant  $k_{\text{kidnapping}}$  where  $i_0 < k_{\text{kidnapping}} < f_0$ , then the algorithm will loop and reduces the size of the tube until it finds possible solution again, *i.e.* consider a new moment  $i_1$  to start the tubes at, beyond the anomaly:  $k_{\text{kidnapping}} < i_1$ . This algorithm is described in Algorithm 4.2.

---

**Algorithm 4.2** RTubeCSIVIA(**in** :  $[\mathbf{x}]_{i,f}, \mathcal{C}_{\text{motion}}, \mathcal{C}_{\text{obs}}, [s]_{i,f}, \epsilon$ ; **out** :  $\mathbb{X}$ )

---

- 1: TubeCSIVIA(**in** :  $[\mathbf{x}]_{i,f}, \mathcal{C}_{\text{motion}}, \mathcal{C}_{\text{obs}}, [s]_{i,f}, \epsilon$ ; **out** :  $\mathbb{X}$ )
  - 2: **while**( $\mathbb{X}$  is empty):
    - // Reduce history size*
  - 3:     increase  $i$
  - 4:     TubeCSIVIA(**in** :  $[\mathbf{x}]_{i,f}, \mathcal{C}_{\text{motion}}, \mathcal{C}_{\text{obs}}, [s]_{i,f}, \epsilon$ ; **out** :  $\mathbb{X}$ )
- 

In this particular case, figure 4.15e, because the kidnapping almost instantaneously leads to an empty  $\mathbb{X}$ , reducing the size of the history is equivalent to resetting it. However, this is not usually the case as sometimes the outliers might be compatible with the following readings but not with others in the future. Therefore, it might be more interesting to only reduce the history instead of re-initializing it to gain time on convergence towards the solution. The algorithm user might as well prefer the reinitialize the history each time it does not find a set of solutions.



(a) Buffer Size: 500 Samples.



(b) Buffer Size: 2000 Steps.

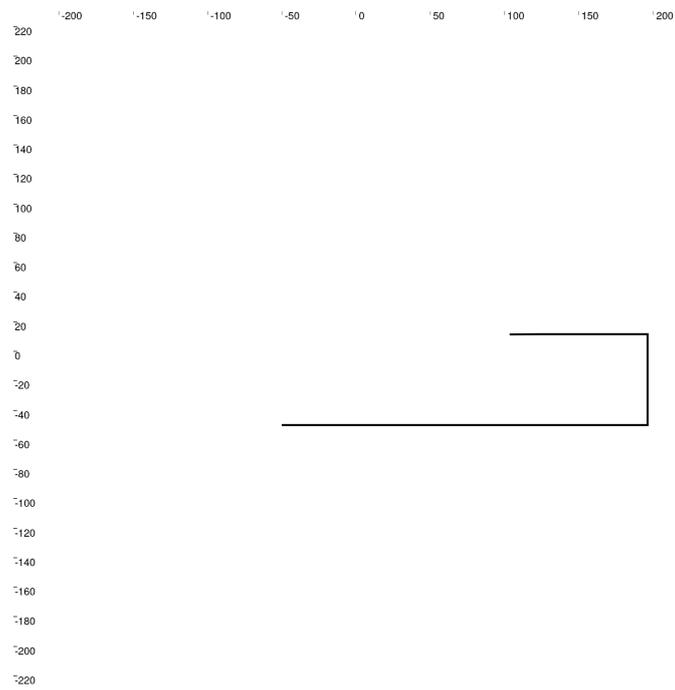


(c) Buffer Size: 5000 Samples.



(d) Buffer Size: 7000 samples.

**Figure 4.6:** Solution Sets Of The Vehicle Position According To The TubeCSIVIA Algorithm. The Vehicle Is Following A Straight Line.



**Figure 4.7:** Path Of The Vehicle (20000 steps).



(a) Buffer Size: 1000 samples.



(b) Buffer Size: 7000 Samples.

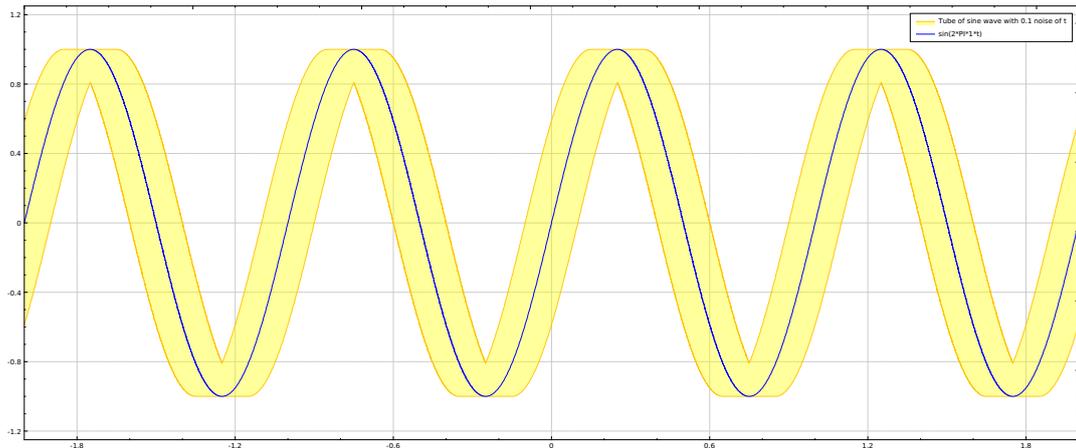


(c) Buffer Size: 10000 samples.

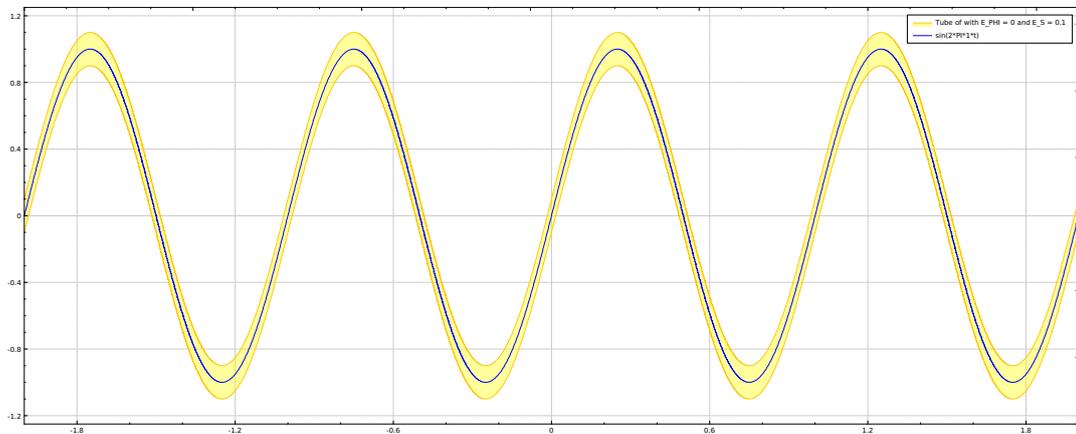


(d) Buffer Size: 20000 samples.

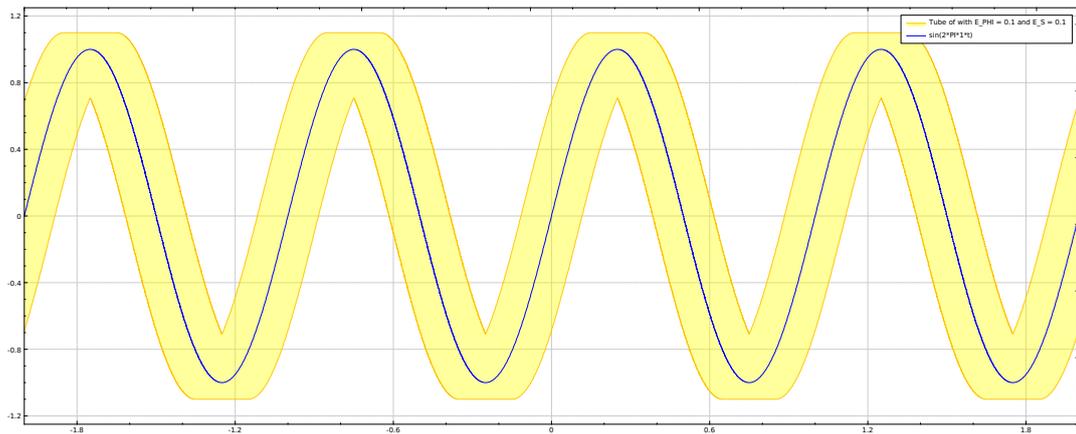
**Figure 4.8:** Solutions Of The Vehicle Position According To The TubeCSIVIA Algorithm. The Vehicle Is Following A Asymmetrical Path.



(a) The received signal  $s_{\text{noisy}}$  with noise  $\epsilon_{\psi} = 0.1$ .

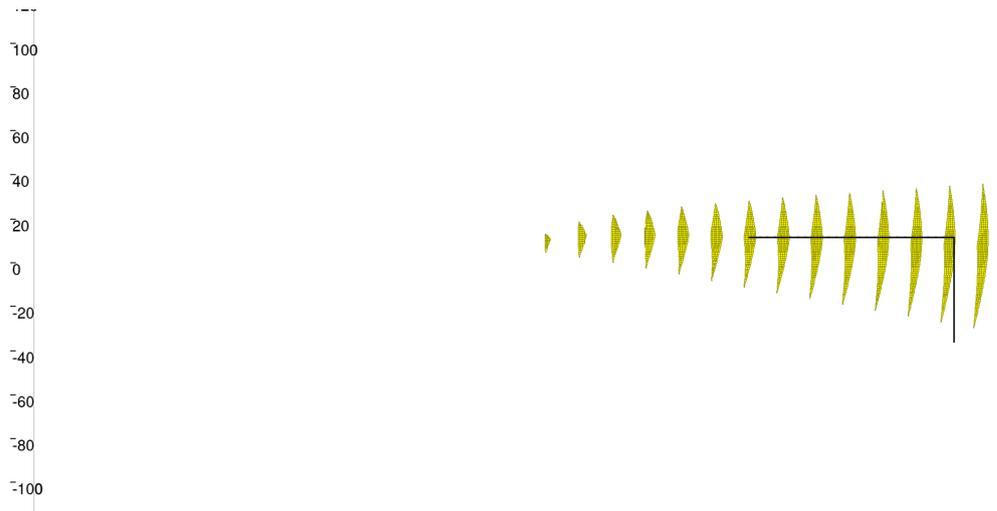


(b) The received signal  $s_{\text{noisy}}$  with noise  $\epsilon_s = 0.1$ .



(c) The received signal  $s_{\text{noisy}}$  with both noises  $\epsilon_s = 0.1$  and  $\epsilon_{\psi} = 0.1$ .

**Figure 4.9:** Representation of the received signal  $s_{\text{noisy}}$  with different noises.



(a) Buffer Size: 7000 samples.



(b) Buffer Size: 10000 samples.



(c) Buffer Size: 20000 samples.

**Figure 4.10:** Solutions Of The Vehicle Position According To The TubeCSIVIA Algorithm with the noise  $[\epsilon_\psi]$ .

100  
80  
60  
40  
20  
0  
-20  
-40  
-60  
-80  
-100

100  
80  
60  
40  
20  
0  
-20  
-40  
-60  
-80  
-100

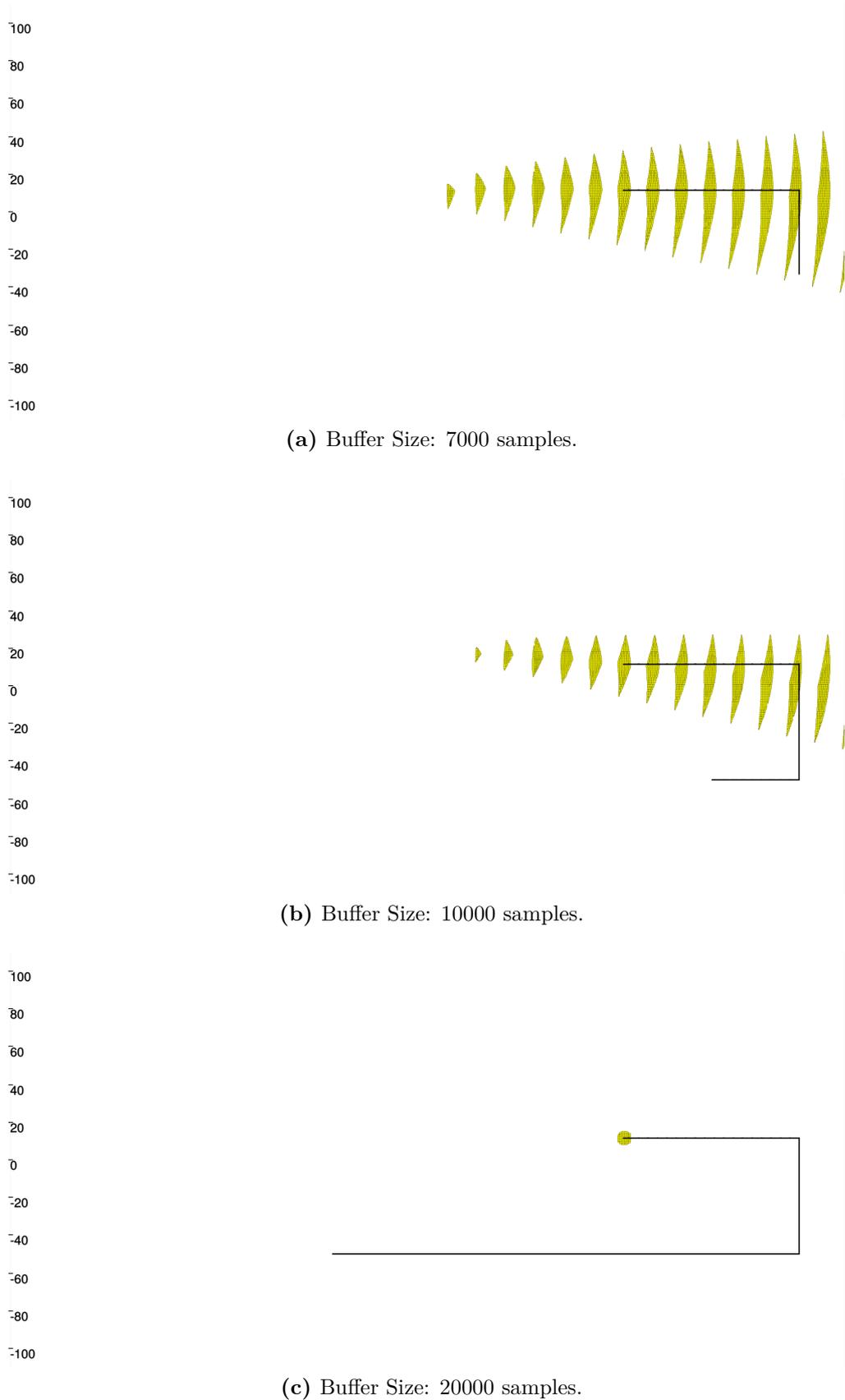
100  
80  
60  
40  
20  
0  
-20  
-40  
-60  
-80  
-100

(a) Buffer Size: 7000 samples.

(b) Buffer Size: 10000 samples.

(c) Buffer Size: 20000 samples.

**Figure 4.11:** Solutions Of The Vehicle Position According To The TubeCSIVIA 66Algorithm with the noise  $[\epsilon_s]$ .



**Figure 4.12:** Solutions Of The Vehicle Position According To The TubeCSIVIA Algorithm with the noises  $[\epsilon_\psi]$  and  $[\epsilon_s]$ .

100  
80  
60  
40  
20  
0  
-20  
-40  
-60  
-80  
-100

100  
80  
60  
40  
20  
0  
-20  
-40  
-60  
-80  
-100

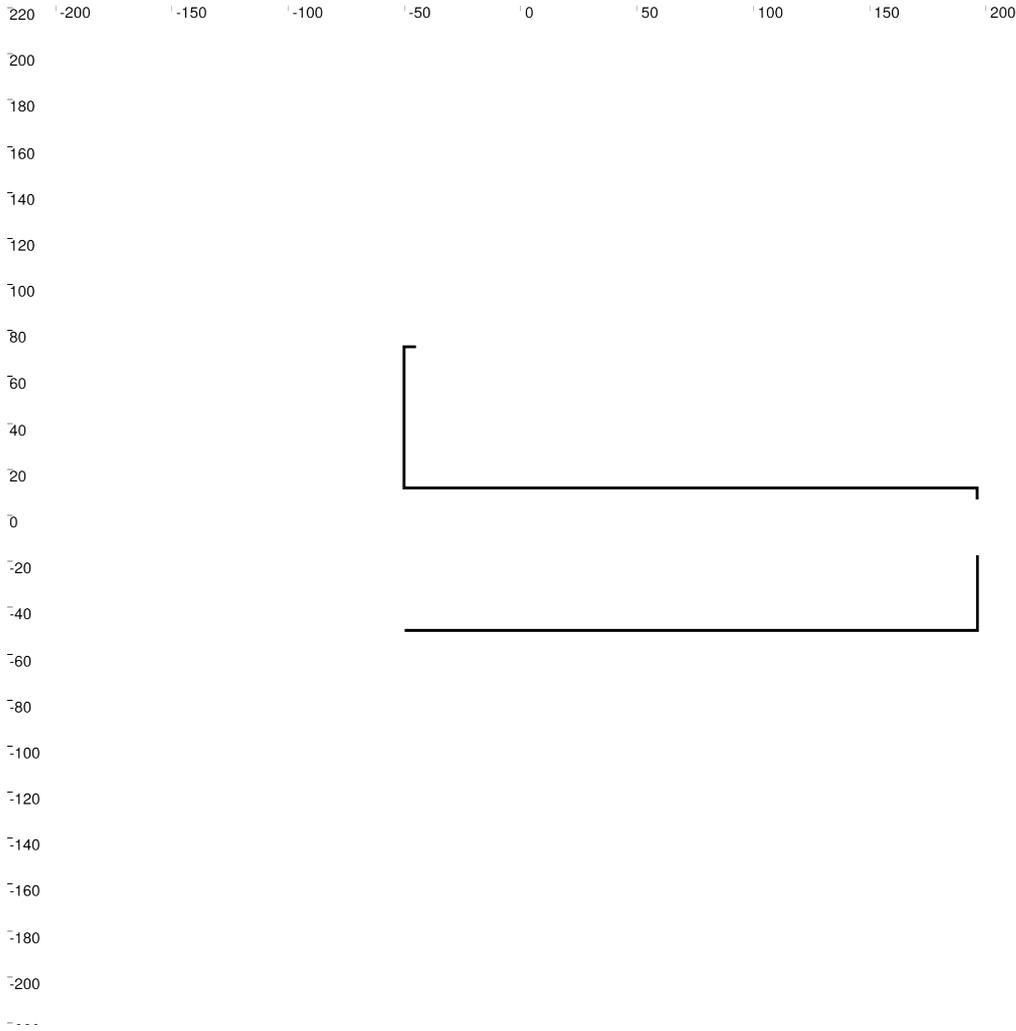
100  
80  
60  
40  
20  
0  
-20  
-40  
-60  
-80  
-100

(a) Buffer Size: 7000 samples.

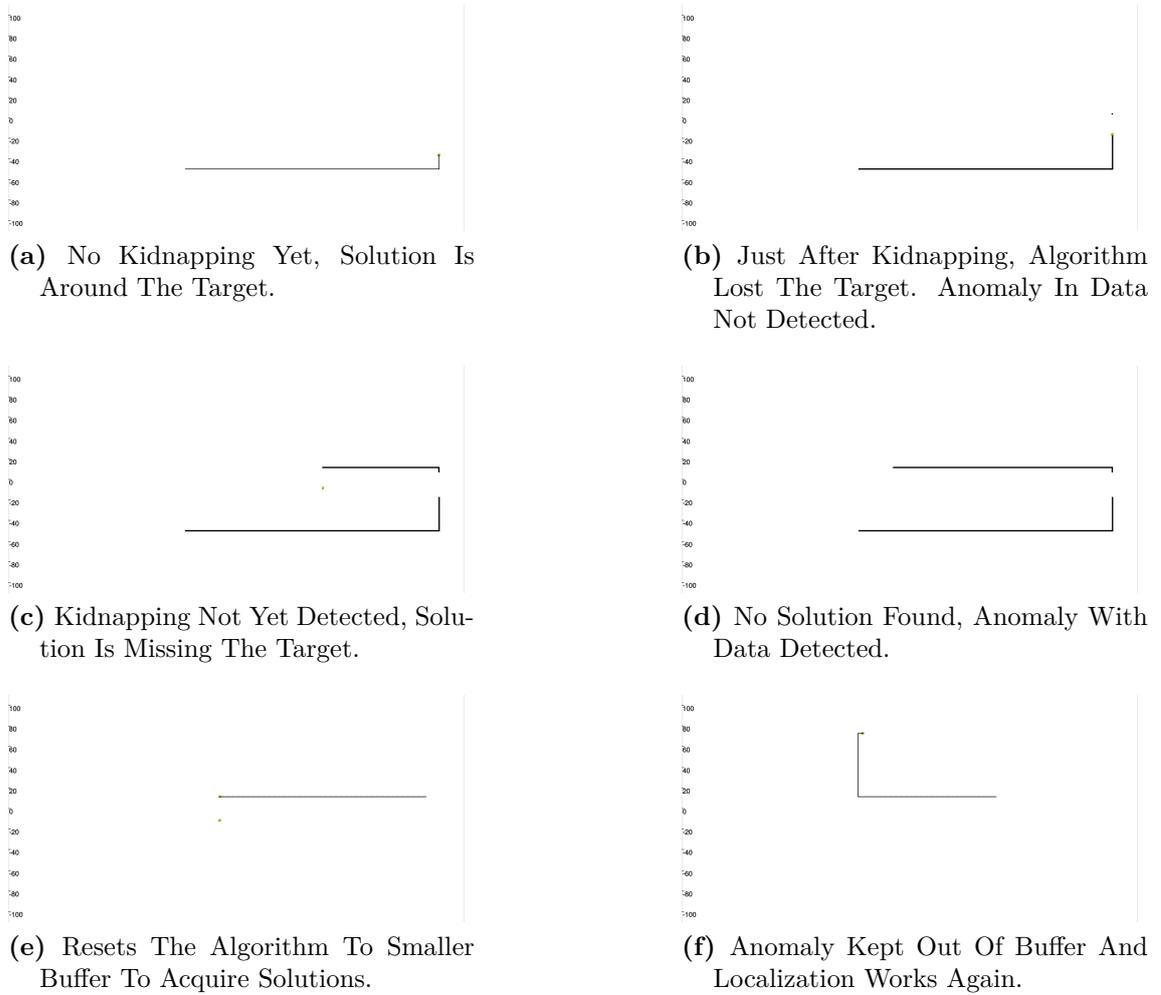
(b) Buffer Size: 10000 samples.

(c) Buffer Size: 20000 samples.

**Figure 4.13:** Solutions Of The Vehicle Position According To The TubeCSIVIA 68Algorithm with the noise  $[\epsilon_x]$ .



**Figure 4.14:** Path Of The Vehicle During Kidnapping.



**Figure 4.15:** Solutions Of The Vehicle Position According To The TubeCSIVIA Algorithm With Kidnapping.

## 4.6 Conclusion

In this chapter, we presented a new on-board localization technique capable of localizing an infinite number of vehicle within range of the source. This method based on interval analysis and set-inversion only requires the vehicles to be synchronized to the beacons clock, modulo  $\omega_s$ .

The algorithm behind the method relies on the previously explained basis of interval analysis and combine the power of contractors and tubes to inverse a non-linear system. The use of tubes does also allow the reconstruction of the path and does not require the starting point to be known. Moreover, modifying the length of the tubes robustifies the algorithm against outliers without requiring the computationally expensive task Q-Intersection or Robust-SIVIA, see annex ??.

This method requires only one way communication from the beacon, therefore it is scalable and the vehicles do not need to ping back therefore being more power efficient. Nevertheless, producing a pure tone signal continuously over a long period of time with today's technologies is not yet reliable in such conditions. Luckily, many acoustic source systems are in development [74] that will enable this method to be used in the future.



# 5 Collaborative Localization

## 5.1 Introduction

When missions require a multi-robot operation to collaborate for instance, usually the knowledge of the of the collaborator pose in the environment and the task it is performing is crucial to the success of the mission. In the previous chapter, a system enabling each individual robot to know its position according to a fixed beacon. However this system does not enable a swarm of vehicles to acquire the position of each other vehicle.

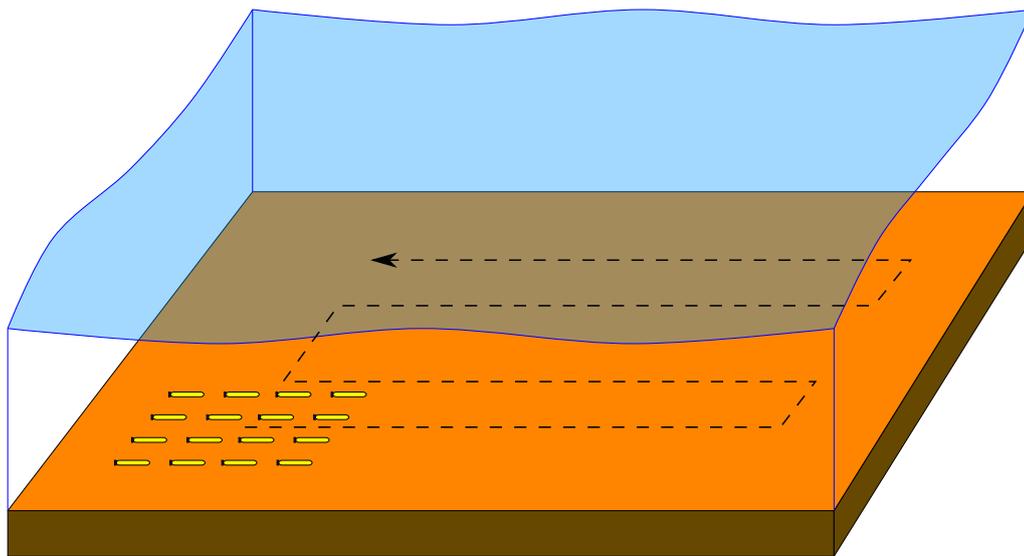
In this chapter, we explore a method for the cooperative localization using range and bearing information between the vehicles. Similar work has explored cooperative localization in aerial and ground robots [75, 76], but also for underwater robots [77]. Bethencourt thesis [78] even uses interval analysis to propagate position information between underwater vehicles that have access to GPS and those who are performing tasks deeper underwater. However in order to use the system as presented by Bethencourt, vehicles need to exchange the necessary data when they are within range of communication. For this condition to be satisfied, the vehicles need to be close to exchange the necessary information. Therefore either the vehicles on the surface need to dive or conversely the deeper vehicle to almost surface to exchange data. In both cases, one of the vehicles must interrupt its mission for the exchange to happen. To tackle these issues, we propose a method based on interval analysis but that will not require data exchange between the vehicles.

In section 5.2 the problem to be studied is stated and an interval state estimator is described. An interval state estimator has already been used in the previous chapter, but not explained in details. Then the *Polar Contractor* is introduced in section 5.3. This contractor contracts Cartesian coordinates from polar coordinates and inversely. The section 5.4 exposes the different case-studies where such a system can be used. These case-studies are simulated and commented. A summary and perspectives are provided in section 5.5.

## 5.2 Problem Statement & Interval State Estimator

### 5.2.1 Problem Statement

Suppose a swarm of vehicles on a mission where a large area must be swept using the on-board sensors, Figure 5.1. For the mission to be successful, the whole area must be surveyed and preferably in the minimum required time. Therefore each vehicle should as much as possible know its position within the formation.



**Figure 5.1:** Area To Be Surveyed By The Swarm

Long-baseline can offer a good positioning accuracy, provided the array is correctly calibrated. However, the deployment, the recovery, and the calibration process is time-consuming and therefore very expensive, especially for large areas. Therefore to overcome the limitation of the beacon array, we propose a multi-vehicle navigation approach similar to the one tested by Curtin and al. [79] where each vehicle is equipped with an ultra-short baseline in the nose of a vehicle and a beacon in the other. Nevertheless, in our approach, each vehicle would be equipped with an acoustic range-and-bearing device that can be assimilated to an USBL. Yet, for economical purposes and because the high number of vehicles to used, we not only study the case range-and-bearing is available but also when the vehicles are equipped with range-only and bearing-only sensors.

### 5.2.2 Interval State Estimator

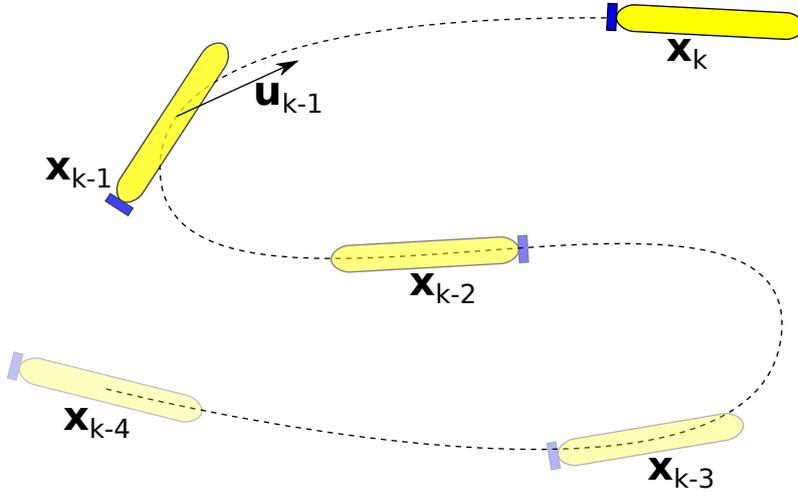
For a vehicle operating in a 3D-space, the state vector is usually comprised of the vehicle's position in a reference frame and its orientation (Euler angles):  $\mathbf{x} = [x, y, z, \phi, \psi, \theta]^T$ . Suppose that all the vehicles in swarm are almost at the same

depth  $z$ , and do not control their roll  $\phi$ , and pitch  $\psi$ . Then, it is only significant to use the remaining variables:  $\mathbf{x} = [x, y, \theta]^T$  to determine their pose.

Because even the most complex models cannot represent fully a real robot operating in a real environment due to uncertainties, we chose the following 2D kinematic model as the motion model to be used in this chapter:

$$\begin{aligned} \dot{\mathbf{x}} &= f(\mathbf{x}, \mathbf{u}) \\ \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} &= \begin{bmatrix} u_v \cos \theta \\ u_v \sin \theta \\ u_\omega \end{bmatrix}, \end{aligned} \quad (5.1)$$

where  $\mathbf{u} = [u_v, u_\omega]^T$  is the control input vector controlling the speed of the vehicle  $u_v$  and its angular velocity  $u_\omega$ . Figure 5.2 shows this simple kinematic model.



**Figure 5.2:** Simple Motion Model

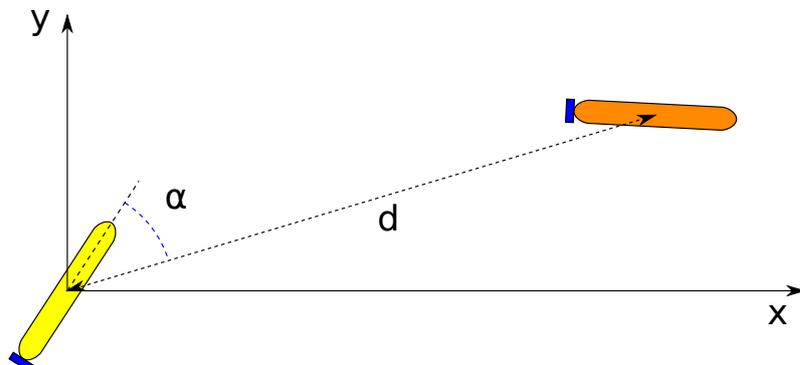
In a discrete time space, we choose a sampling rate  $h$  small enough to assume the variable are constant in this time interval. For a sample  $k$ , the discrete version of Equation 5.1 is then:

$$\mathbf{x}_k = f(\mathbf{x}_{k-1}, \mathbf{u}_{k-1}) = \mathbf{x}_{k-1} + h \begin{bmatrix} u_{v_{k-1}} \cos \theta_{k-1} \\ u_{v_{k-1}} \sin \theta_{k-1} \\ u_{\omega_{k-1}} \end{bmatrix} + [\epsilon], \quad (5.2)$$

where  $[\epsilon]$  is the noise term introduced by both the discretization and the model.

Consider now the variables in  $\mathbb{IR}$ ,  $[\mathbf{x}_k]$  can be estimated from the previous state  $[\mathbf{x}_{k-1}]$  and knowledge of the input  $[\mathbf{u}_{k-1}]$ . Nevertheless, in the case of cooperative relative localization, the vehicles are not estimating their own state but the state of the surrounding vehicles. Therefore, because of the underwater communication issue, the vehicles would not be able to share their data with ease.

In order to solve the problem of cooperative localization, additional exteroceptive sensors measure range (distance)  $d_i$  and bearing angle  $\alpha_i$  from other vehicles, Figure 5.3.



**Figure 5.3:** Collaborative Localization - Range & Bearing

Given that each vehicle is equipped with a compass, conversion between the position of a remote vehicle in the local vehicle's frame and the range and bearing is equivalent to the conversion between Cartesian and polar coordinates.

### 5.3 Polar Contractor

As the only information available about the remote vehicle is its range and bearing, therefore a polar to Cartesian contractor should be considered.

This contractor would contract the Cartesian coordinates  $x$  and  $y$  and the polar coordinates  $d$  and  $\alpha$  according the following constraints

$$x = d \cos \alpha \quad (5.3)$$

$$y = d \sin \alpha \quad (5.4)$$

$$d = \sqrt{x^2 + y^2} \quad (5.5)$$

$$\alpha = \text{atan2}(y, x). \quad (5.6)$$

The section 3.4 details how to contract for instance Equation 5.3, Equation 5.4, and Equation 5.5. Nevertheless, the contractor of  $\text{atan2}(.,.)$  for Equation 5.6 is more complex because the function is non-monotonic and with discontinuities. Bethencourt proposes a contractor for  $\text{atan2}(.,.)$  in his thesis [78] based on Herrero approach [80]. However, we found that using a smaller seed for the contractor and using contractors properties subsection 3.4.1 eliminates the issue of the undefined  $\tan(.)$  around  $\frac{\pi}{2}$ , *i.e.* we propose to start with a the seed set  $\mathbb{S}_{\text{seed}} = \mathbb{S} \cap (\mathbb{I}\mathbb{R}^+ \times \mathbb{I}\mathbb{R}^+ \times [0, \frac{\pi}{4}])$ .

Since in  $\mathbb{S}_{\text{seed}}$

$$(x, y, \alpha) \in \mathbb{S}_{\text{seed}} \Leftrightarrow \begin{cases} x &= y \cot(\alpha) \\ y &= x \tan(\alpha) \\ \alpha &= \arctan\left(\frac{y}{x}\right) \end{cases},$$

the minimal contractor is

$$\mathcal{C}_{\text{atan2}_{\text{seed}}}([x], [y], [\alpha]) \rightarrow \left( \begin{array}{c} \left( [x] \cap \mathbb{I}\mathbb{R}^+ \right) \cap \left( \left( [y] \cap \mathbb{I}\mathbb{R}^+ \right) \cdot \cot \left( [\alpha] \cap \left[ 0, \frac{\pi}{4} \right] \right) \right) \\ \left( [y] \cap \mathbb{I}\mathbb{R}^+ \right) \cap \left( \left( [x] \cap \mathbb{I}\mathbb{R}^+ \right) \cdot \tan \left( [\alpha] \cap \left[ 0, \frac{\pi}{4} \right] \right) \right) \\ \left( [\alpha] \cap \left[ 0, \frac{\pi}{4} \right] \right) \cap \arctan \left( \frac{([y] \cap \mathbb{I}\mathbb{R}^+)}{([x] \cap \mathbb{I}\mathbb{R}^+)} \right) \end{array} \right).$$

From this seed, it is now possible to build a contractor for  $\mathbb{S}_{\left[\frac{\pi}{4}, \frac{\pi}{2}\right]} = \mathbb{S} \cap (\mathbb{I}\mathbb{R}^+ \times \mathbb{I}\mathbb{R}^+ \times \left[\frac{\pi}{4}, \frac{\pi}{2}\right])$  using axial symmetry:

$$\mathcal{C}_{\text{atan2}_{\left[\frac{\pi}{4}, \frac{\pi}{2}\right]}}([x], [y], [\alpha]) = \mathcal{C}_{\text{atan2}_{\text{seed}}}([y], [x], \frac{\pi}{2} - [\alpha]),$$

and then a contractor for  $\mathbb{S}_{\left[0, \frac{\pi}{2}\right]}$  with the union:

$$\mathcal{C}_{\text{atan2}_{\left[0, \frac{\pi}{2}\right]}} = \mathcal{C}_{\text{atan2}_{\left[\frac{\pi}{4}, \frac{\pi}{2}\right]}} \cup \mathcal{C}_{\text{atan2}_{\text{seed}}},$$

and so on for the domain  $\mathbb{I}\mathbb{R} \times \mathbb{I}\mathbb{R} \times [-\pi, \pi]$  where *atan2* is defined. The Algorithm 5.1 details these steps that are based on contractor operation defined in the section 3.4.

Figure 5.4 shows the result of the SIVIA algorithm applied on the space  $[-10, -10]^2$ . Provided the angle  $[0.5, 1.0]$  as an input, the algorithm is capable of providing both the inner and outer approximation of the solution set.

Even so, the angle should not be limited to  $[-\pi, \pi]$  but rather to  $\mathbb{I}\mathbb{R}$  not to constraint any sensor on providing angles in the specific interval and having to deal with the discontinuity. Let's then define the contractor  $\mathcal{C}_{\text{angle}}$  for  $\mathbb{S} \subset \mathbb{I}\mathbb{R}^3$  as  $\mathcal{C}_{\text{angle}} = \left( \mathcal{C}_{\text{atan2}_{[-\pi, \pi]}}([x], [y], [\alpha] \bmod 2\pi) \right)$ , Algorithm 5.2.

Finally, the contractor for Cartesian and polar coordinates is built using  $\mathcal{C}_{\text{angle}}$  and other contractors, Algorithm 5.3. Figure 5.5 shows the result of SIVIA algorithm applied on a box to convert polar coordinates to Cartesian's. However, this contractor is not minimal but is sufficient as range and bearing will be used separately in this case. A minimal polar contractor can be found in [81].

**Algorithm 5.1**  $\mathcal{C}_{\text{atan2}}([x], [y], [\alpha])$ 


---

```

1: if  $([\alpha] \subset [0, \frac{\pi}{4}])$ :
2:   return  $\left( \begin{array}{l} \left( [x] \cap \mathbb{I}\mathbb{R}^+ \right) \cap \left( \left( [y] \cap \mathbb{I}\mathbb{R}^+ \right) \cdot \cot \left( [\alpha] \cap \left[ 0, \frac{\pi}{4} \right] \right) \right) \\ \left( [y] \cap \mathbb{I}\mathbb{R}^+ \right) \cap \left( \left( [x] \cap \mathbb{I}\mathbb{R}^+ \right) \cdot \tan \left( [\alpha] \cap \left[ 0, \frac{\pi}{4} \right] \right) \right) \\ \left( [\alpha] \cap \left[ 0, \frac{\pi}{4} \right] \right) \cap \arctan \left( \frac{([y] \cap \mathbb{I}\mathbb{R}^+)}{([x] \cap \mathbb{I}\mathbb{R}^+)} \right) \end{array} \right)$ 
3: else if  $([\alpha] \subset [\frac{\pi}{4}, \frac{\pi}{2}])$ :
4:    $\mathcal{C}_{\text{atan2}}([x], [y], \frac{\pi}{2} - [\alpha])$ 
5: else if  $([\alpha] \subset [0, \frac{\pi}{2}])$ :
6:    $\mathcal{C}_{\text{atan2}}([x], [y], ([\alpha] \cap [0, \frac{\pi}{4}])) \cup \mathcal{C}_{\text{atan2}}([x], [y], ([\alpha] \cap [\frac{\pi}{4}, \frac{\pi}{2}]))$ 
7: else if  $([\alpha] \subset [\frac{\pi}{2}, \pi])$ :
8:    $\mathcal{C}_{\text{atan2}}(-[x], [y], \pi - [\alpha])$ 
9: else if  $([\alpha] \subset [-\pi, -\frac{\pi}{2}])$ :
10:   $\mathcal{C}_{\text{atan2}}(-[x], -[y], \pi + [\alpha])$ 
11: else if  $([\alpha] \subset [-\frac{\pi}{2}, 0])$ :
12:   $\mathcal{C}_{\text{atan2}}([x], -[y], -[\alpha])$ 
13: else if  $([\alpha] \subset [0, \pi])$ :
14:   $\mathcal{C}_{\text{atan2}}([x], [y], ([\alpha] \cap [0, \frac{\pi}{2}])) \cup \mathcal{C}_{\text{atan2}}([x], [y], ([\alpha] \cap [\frac{\pi}{2}, \pi]))$ 
15: else if  $([\alpha] \subset [-\pi, 0])$ :
16:   $\mathcal{C}_{\text{atan2}}([x], [y], ([\alpha] \cap [-\pi, -\frac{\pi}{2}])) \cup \mathcal{C}_{\text{atan2}}([x], [y], ([\alpha] \cap [-\frac{\pi}{2}, 0]))$ 
17: else if  $([\alpha] \subset [-\pi, \pi])$ :
18:   $\mathcal{C}_{\text{atan2}}([x], [y], ([\alpha] \cap [-\pi, 0])) \cup \mathcal{C}_{\text{atan2}}([x], [y], ([\alpha] \cap [0, \pi]))$ 
19: else if  $([\alpha] \subset [-\pi, \pi])$ :
20:   $\mathcal{C}_{\text{atan2}}([x], [y], ([\alpha] \cap [-\pi, 0])) \cup \mathcal{C}_{\text{atan2}}([x], [y], ([\alpha] \cap [0, \pi]))$ 

```

---

**Algorithm 5.2**  $\mathcal{C}_{\text{angle}}([x], [y], [\alpha])$ 

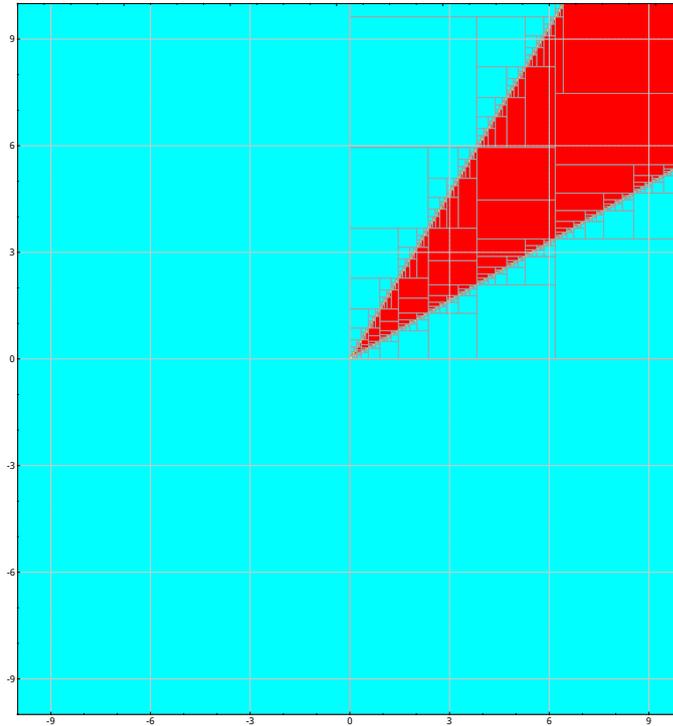

---

```

1: if  $([\alpha] \subset ([-\pi, \pi] \bmod 2\pi))$ :
2:    $\mathcal{C}_{\text{atan2}}([x], [y], [\alpha])$ 
3: else:
4:    $\mathcal{C}_{\text{atan2}}([x], [y], ([\alpha] \bmod 2\pi) \cap [0, \pi]) \cup \mathcal{C}_{\text{atan2}}([x], [y], ([\alpha] \bmod 2\pi) \cap [-\pi, 0])$ 

```

---



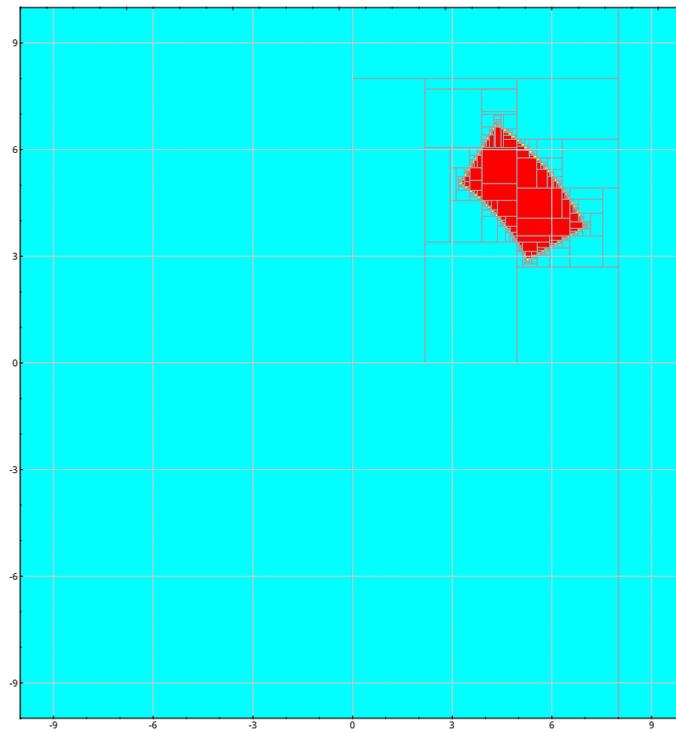
**Figure 5.4:** SIVIA Algorithm Applied Using The Contractor  $\mathcal{C}_{\text{atan2}}$  On A Box  $([-10, 10], [-10, 10], [0.5, 1])$ .

---

**Algorithm 5.3**  $\mathcal{C}_{\text{polar}}([x], [y], [d], [\alpha])$

---

- 1:  $\mathcal{C}_{\text{angle}}([x], [y], [\theta])$
  - 2:  $[d] = [d] \cap \left( \sqrt{[x]^2 + [y]^2} \right)$
  - 3:  $[x] = [x] \cap ([d] \cos [\alpha]) \cap \left( \sqrt{[d]^2 - [y]^2} \right)$
  - 4:  $[y] = [y] \cap ([d] \sin [\alpha]) \cap \left( \sqrt{[d]^2 - [x]^2} \right)$
-



**Figure 5.5:** SIVIA Algorithm Applied Using The Contractor  $\mathcal{C}_{\text{polar}}$  On A Box  $([-10, 10], [-10, 10], [6, 8], [0.5, 1])$ .

## 5.4 Cooperative Localization

Suppose now a swarm of robots on a mission. Each robot is equipped with on-board proprioceptive sensors and exteroceptive sensors depending on the case study below. Unlike single or small multi-vehicle missions, a large scale swarm mission must be as cost efficient as possible thus embedding the vehicles with only the necessary sensors. Below are the different scenarios that will be studied and then simulated. The cases range from the most expensive, vehicles equipped with range and bearing sensors and acoustic modems, to the least expensive setups, vehicles equipped with only range systems and without inter-vehicle communications.

As the mission are performed underwater, a comparison is done between a communication enabled scenario and a scenario where no localization data is exchanged between the vehicles. While acquiring range and bearing data, vehicles are able to differentiate between the neighbors. Some might consider the fact that exchanging identifiers is a kind of communication between the vehicles, which is true. However by communication is this chapter we mean the exchange of information about the state of the vehicles between each swarm element.

The comparison highlights the importance of the exchange of data in the outcome of such a collaborative positioning and also that it is still possible to acquire a relatively good positioning without communications when the vehicles collaborate by maneuvering to avoid positioning ambiguities.

### 5.4.1 Range & Bearing Measurements

In this scenario, each vehicle of the swarm is equipped with proprioceptive sensors that provide rough measurements from the vehicle state, and exteroceptive sensors that estimates both the range and the bearings from its neighbors. When exchange of information is available between the vehicles, the vehicle  $\mathbf{j}$  can send its commands  $u_j$  to the vehicle  $\mathbf{i}$  to help with the localization, and *vice versa*.

For a robot  $\mathbf{i} \in \{1, \dots, m\}$  in the swarm, that receives an acoustic ping  $\mathbf{p} \in \{1, \dots, \mathbf{p}_{\max}\}$  from a neighbor robot  $\mathbf{j} \in \{1, \dots, m\}$ , we consider the following discrete state equations at the instant  $k$ :

$$\begin{aligned} \mathbf{x}_{\mathbf{i}k} &= f(\mathbf{x}_{\mathbf{i}k-1}, u_{\mathbf{i}k-1}) \\ \mathbf{y}_{\mathbf{p}k} &= \begin{bmatrix} d_{\mathbf{p}k} & \alpha_{\mathbf{p}k} \end{bmatrix}^T = g(\mathbf{x}_{\mathbf{i}k}, \mathbf{x}_{\mathbf{j}k}) \\ \mathbf{x}_{\mathbf{j}k} &= f(\mathbf{x}_{\mathbf{j}k-1}, u_{\mathbf{j}k-1}) \end{aligned}$$

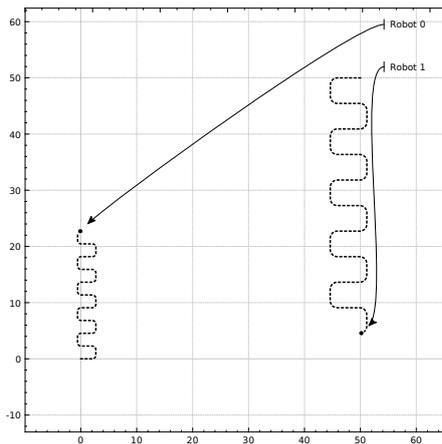
with  $f$  being the evolution function of the vehicle that is function of its previous state  $\mathbf{x}_{\mathbf{i}k-1}$  and its command  $u_{\mathbf{i}k-1}$ . The measurements  $\mathbf{y}_{\mathbf{p}k} = \begin{bmatrix} d_{\mathbf{p}k} & \alpha_{\mathbf{p}k} \end{bmatrix}^T$  are computed using the range and bearing observation function  $g$  according to the neighboring vehicle.

As shown in Figure 5.3, a transformation from the global frame, where the Equation 5.1 describe the vehicles movements, to the local frame of the vehicle localizing its neighbor. We suppose that the vehicles can measure their orientation via sensor, a compass for instance. Therefore, the transformation needed is only related to the translation from the global frame to the local frame  $\mathbf{x}_j^i = \begin{bmatrix} [x_j] - [x_i] & [y_j] - [y_i] & [\theta_j] \end{bmatrix}^T$ , then the evolution function of the relative vehicle position and observation function are

$$\mathbf{x}_{j_k}^i = f'(\mathbf{x}_{j_{k-1}}^i, u_{j_{k-1}}, u_{i_{k-1}}), \quad (5.7)$$

$$\mathbf{y}_{p_k}^i = g'(\mathbf{x}_{j_k}^i). \quad (5.8)$$

Suppose now a swarm of only two vehicles performing a sweep of an area. The path followed by each vehicle are shown in Figure 5.6. Considering any number of vehicles on the swarm does not affect the localization because vehicles at most only exchange their own commands  $u_i$  and no position information.



**Figure 5.6:** Path of the vehicles during collaborative mission.

In this scenario, we suppose that the vehicle *Robot 0* in the figure Figure 5.6 is performing a localization on the vehicle *Robot 1*. To do so, the vehicle uses the algorithms TubeCSIVIA explained in the previous chapter, section 4.4. The polar contractor is used in this case because the range and bearing measurements are equivalent to polar coordinates in the vehicles frame. Furthermore, we suppose that the measurements are subjects to noise:  $[\epsilon_d]$  for the range, and  $[\epsilon_\alpha]$  for the bearing.

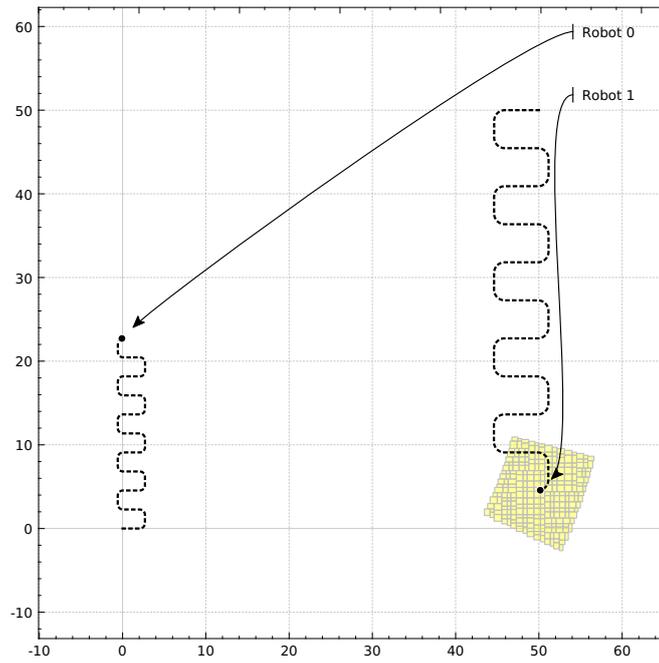
In Figure 5.7, the algorithm shows a slightly better localization when all the measurements are used to find the solution, 5.7b, compared to only using a unique measurement, *i.e.* the last one in this case, 5.7a.

### 5.4.2 Bearing Only

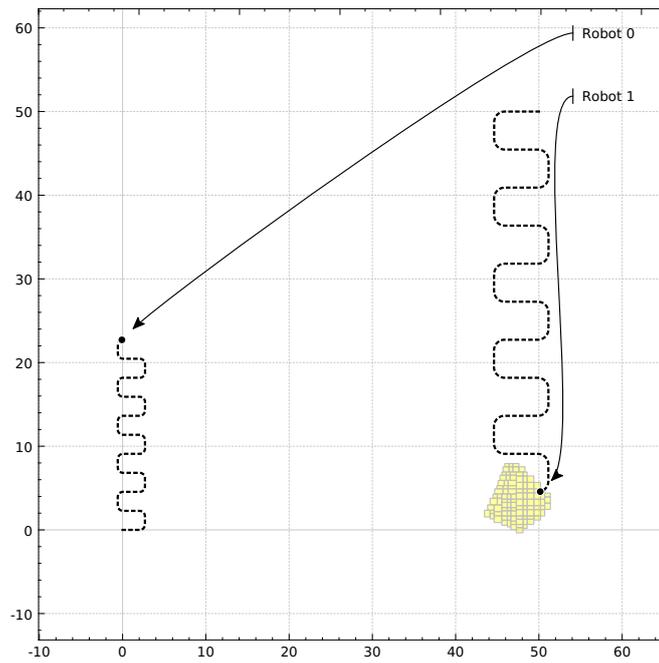
In this scenario, we suppose that the vehicles only have bearing information about their neighbors. Therefore, as shown in 5.8a, *Robot 0* is not capable of precisely locating its neighbor with such few information. Here, the usefulness of the data through the horizon is shown in 5.8b. Where the position of the neighboring vehicle was vague and unbounded in the first case, it is now bounded thanks to the contractions from all the measurement throughout the buffer.

### 5.4.3 Range Only

In this final scenario, vehicles only have access to range information from the neighboring vehicles. Using only a measurement to locate the neighboring vehicle translates to a disk, 5.9a. However, using multiple range measurements throughout the horizon leads to a more accurate localization for the targeted vehicle, 5.9b.

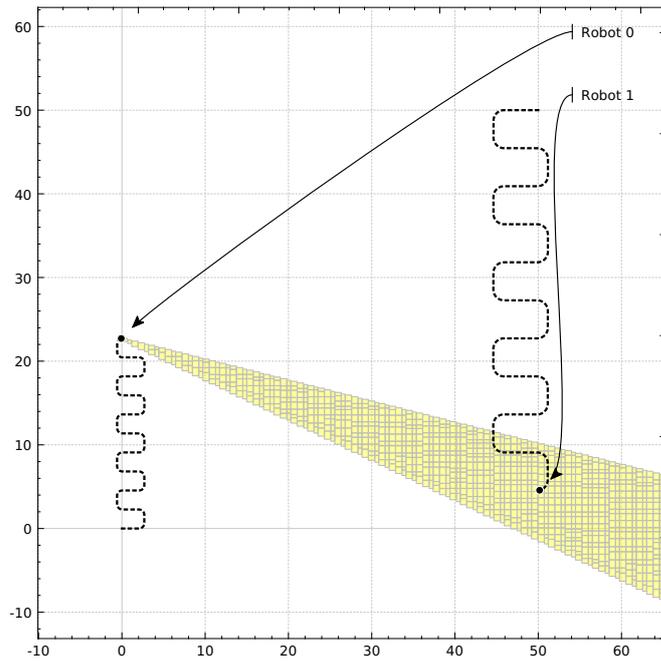


(a) Only Using Last Measurement.

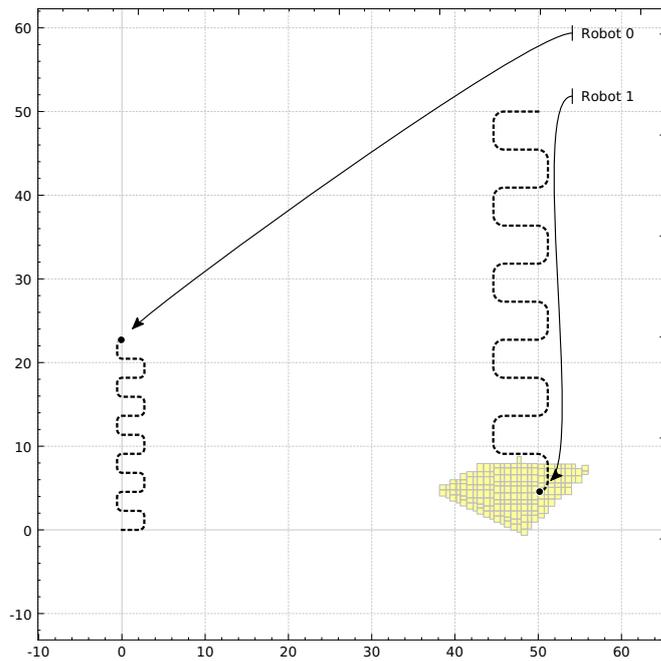


(b) Using Measurements Throughout The Path.

**Figure 5.7:** Range & Bearing Localization With Exchange Of Information With Noise Values  $[\epsilon_d] = [-5, 5]$  And  $[\epsilon_\alpha] = [-0.1, 0.1]$  rad.

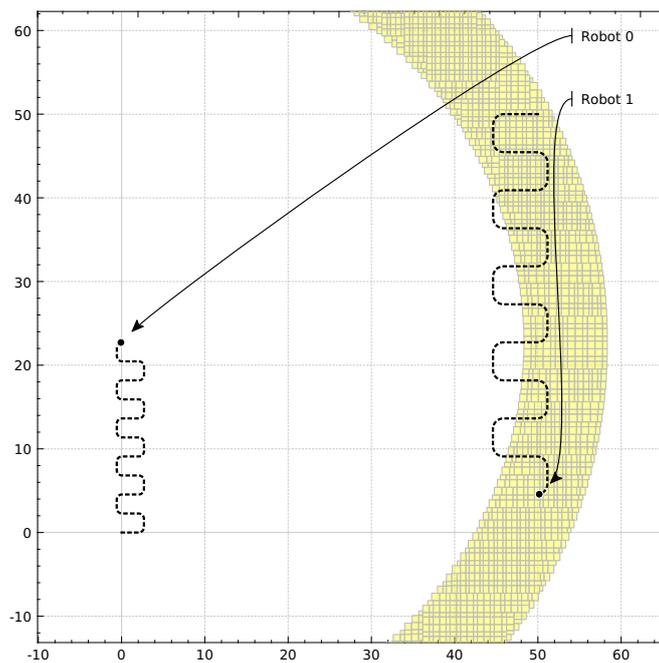


(a) Only Using Last Measurement.

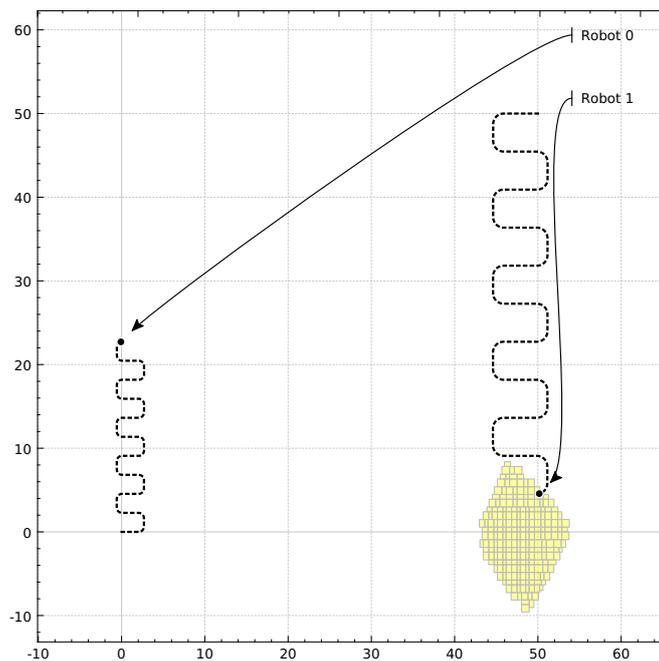


(b) Using Measurements Throughout The Path.

**Figure 5.8:** Bearing-Only Localization With Exchange Of Information With Noise Values  $[\epsilon_\alpha] = [-0.1, 0.1]$  rad.



(a) Only Using Last Measurement.



(b) Using Measurements Throughout The Path.

**Figure 5.9:** Range-Only Localization With Exchange Of Information With Noise  $[\epsilon_d] = [-5, 5]$ .

## 5.5 Conclusion

In this chapter, we presented a way for a swarm of vehicles to mutually localize itself relatively. This method is based on the same algorithm described in the previous chapter, using set-membership methods and propagating the acquired information throughout a window of time leads to a better set of solution of the equation system.

In these scenarios, we supposed that each vehicle can acquire range or bearing or both depending on the sensors on board. These range and bearing information are computed from pings that occur several times during the simulation/mission. Because these range and bearing information can be considered as polar coordinates of the neighboring vehicles, a polar contractor has been developed to solve the localization problem in the Cartesian space. When only one of the two information is provided, range or bearing, the algorithm considers the value of the other measurement as infinite, therefore always being able to reduce the search space. Because of the environment constraints, vehicles are not allowed to exchange data, therefore a comparison between the cases where vehicles are aware or not of the behaviors of the others is also provided for each scenario.

This relative cooperative localization technique has the advantage of being scalable for large number of vehicles. For instance, one can imagine a mission where swarm of vehicles is surveying an area. At predefined moments, each vehicle can emit a signed ping that will be received by the surrounding vehicles. Each vehicle then can compute range or bearing, depending the on-board sensors, of the other vehicles and can locate them in its frame. This way, with post-processing, the position of each vehicle can be retrieved through the mission by intersecting the recorded position of each vehicle in the different frame and projecting the whole on the global frame when possible.

A survey with 16 AUVs is planned on the red sea in December 2015 that will serve as field test for this algorithm. The vehicles will be tracked from the surface with an USBL to later compare the output of the algorithm with the real position of the vehicles.



## 6 Conclusion

As Autonomous Underwater Vehicles are getting broader attention, many industries consider them now more mature and useful in many cases. However, even if most of the known barriers are getting surpassed by advances in fields like energy, embedded computers, and resistance of materials to pressure, the use of these vehicles is still limited by the localization and navigation issue. Actually, one of the most important concerns by the users of this technology is to be able to position the recording data back on a map, whether this data is bathymetry of a sonar or seismic data in the particular case of Oil & Gas exploration industry.

This thesis starts by providing an overview of the challenges that lead to it. First, the recent *democratization* of the unmanned vehicles that expanded to different areas from air to land and marine robots. We provide an insight on the most common vehicles being deployed in the marine environment. Then, we introduce the context of Oil & Gas exploration and particularly marine seismic acquisition where the use of hundreds of AUVs is now being considered. The hypothesis, constraints and objectives are developed in the end of the first chapter.

In the second chapter, challenges and limitations faced by all underwater vehicles are described. These limitations range from the communication, due to the absence of electromagnetic waves and the use of the very noisy medium of acoustics, the energy availability, to finally the localization and navigation problem tackled with this thesis. We also provided a list of the most common and reliable systems that this thesis challenges.

The following chapter introduced the basics of Set-Membership methods that are the foundation of this thesis. The philosophy of using intervals instead of scalars and their arithmetic is detailed. Then, contractors and Set Inversion Via Interval Analysis, the main tools used through this thesis, are described.

The first contribution of this thesis is explained in the fourth chapter. There, we propose an innovative system for underwater localization that is based on a single beacon emitting a continuous sine wave for a certain period of time. Using the phase shift, introduced by the movement of the vehicle, and the proprioceptive measurements of the vehicle, the latter is capable of converging to a set of solutions that satisfy the evolution and observation equations. The algorithm uses interval analysis and contractors over a sliding horizon to converge toward the solution. We showed that the algorithm has poor performance when the vehicle is stationary or follows a straight line. However, as soon as the vehicles performs a maneuver that

breaks the symmetry of the solution sets, the algorithm is capable of eliminating all the sets that are incoherent with the observations. We showed also the advantage of using the sliding horizon with different buffer sizes. The latter technique allows the vehicle to navigate continuously without having to resurface when an anomaly happens. We detailed the case where the vehicle is kidnapped from a position to another. In that case, we showed how the algorithm fails to follow the vehicle in the beginning as the observations were coherent for an interval of time ignoring the anomaly. But as all the observations in the horizon are taken into account to produce the set of possible positions, the algorithm then discovers that an anomaly is present and reduces the number of observations therefore sliding the search horizon.

The second contribution is detailed in the fifth chapter. We proposed a system of inter-vehicle localization for a swarm of underwater vehicles. The localization approach in this chapter is different from the previous chapter as now it is fully relative to the other neighboring vehicles within range of the signal. We proposed to use the same algorithm as the previous chapter with a sliding horizon but with different contractors this time. In this chapter, the vehicle localizes a target vehicle within observation range. The vehicles use first range and bearing information to localize the target. Then we explore the two cases where only bearing or range information is available. In these scenarios we also suppose the cases where the vehicles do not exchange information. We introduce the polar contractor that is used in this case as range and bearing provide polar coordinates that need to be converted into Cartesian coordinate system.

Based on this thesis, future work can introduce better performance of these techniques. As of now, the performance of the algorithms highly depend on the handling of tubes and their contractors that have not been optimized. Furthermore, because of the lack of real data, both contributions still need to be tested on the field as many challenges might not have been considered. For instance, the phase-based localization still have to surpass the challenge of generating a continuous sine wave. Today's acoustic emitters are not rated for continuous emission over a long period and recent advances in underwater speakers are still too expensive to be deployed. As for the collaborative localization, the hypothesis of forever synchronous clocks needs to be challenged and resolved. Moreover, across the thesis, we supposed that the underwater environment is homogeneous and uniform with the absence of thermocline and only considered these characteristics as probable errors in measurements introducing noise. One may argue that such phenomena has further effect on the measurements to only be considered as noise.

# Bibliography

- [1] E. F. Danson, “Auv tasks in the offshore industry,” *Technology and Applications of Autonomous Underwater Vehicles*, Griffiths, G.(ed). Taylor and Francis, London, pp. 127–138, 2003.
- [2] T. Chance, A. Kleiner, and J. Northcutt, “The autonomous underwater vehicle (auv): a cost-effective alternative to deep-towed technology,” *Integrated Coastal Zone Management*, vol. 2, no. 7, pp. 65–69, 2000.
- [3] C. C. Eriksen, T. J. Osse, R. D. Light, T. Wen, T. W. Lehman, P. L. Sabin, J. W. Ballard, and A. M. Chiodi, “Seaglider: A long-range autonomous underwater vehicle for oceanographic research,” *Oceanic Engineering, IEEE Journal of*, vol. 26, no. 4, pp. 424–436, 2001.
- [4] M. Benjamin, J. Curcio, J. Leonard, and P. Newman, “Protocol-based colregs collision avoidance navigation between unmanned marine surface craft,” *Journal of Field Robotics*, vol. 23, no. 5, 2006.
- [5] M. R. Benjamin, J. A. Curcio *et al.*, “Colregs-based navigation of autonomous marine vehicles,” *Proceedings of Autonomous Underwater Vehicles*, pp. 32–39, 2004.
- [6] M. Seto, L. Paull, and S. Saeedi, “Introduction to autonomy for marine robots,” in *Marine Robot Autonomy*. Springer, 2013, pp. 1–46.
- [7] R. W. Button, J. Kamp, T. B. Curtin, and J. Dryden, “A survey of missions for unmanned undersea vehicles,” DTIC Document, Tech. Rep., 2009.
- [8] U. Navy, “The navy unmanned undersea vehicle (uuv) master plan,” *US Navy, November*, vol. 9, p. 90, 2004.
- [9] “Guide for unmanned undersea vehicles (UUV) autonomy and control.” [Online]. Available: <http://dx.doi.org/10.1520/f2541-06>
- [10] M. Benjamin, M. Grund, and P. Newman, “Multi-objective optimization of sensor quality with efficient marine vehicle task execution,” in *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on*. IEEE, 2006, pp. 3226–3232.
- [11] J. Gluyas and R. Swarbrick, *Petroleum geoscience*. John Wiley & Sons, 2013.
- [12] D. P. Hampson, B. H. Russell, B. Bankhead *et al.*, “Simultaneous inversion of pre-stack seismic data,” in *2005 SEG Annual Meeting*. Society of Exploration Geophysicists, 2005.

- 
- [13] O. S. Board *et al.*, *Ocean noise and marine mammals*. National Academies Press, 2003.
- [14] D. B. Kilfoyle and A. B. Baggeroer, “The state of the art in underwater acoustic telemetry,” *Oceanic Engineering, IEEE Journal of*, vol. 25, no. 1, pp. 4–27, 2000.
- [15] N. Størkersen and Ø. Hasvold, “Power sources for auvs,” in *Proceedings of the Science and Defence Conference, Brest, France*, 2004.
- [16] W. S. Burdic, “Underwater acoustic systems analysis,” *The Journal of the Acoustical Society of America*, vol. 89, no. 6, pp. 3020–3021, 1991.
- [17] IXBLUE, “Specifications of phins inertial navigation system.” IXBLUE, Tech. Rep., 2014. [Online]. Available: [https://www.ixblue.com/sites/default/files/downloads/ixblue-ps-phins\\_6000-02-2014-web.pdf](https://www.ixblue.com/sites/default/files/downloads/ixblue-ps-phins_6000-02-2014-web.pdf)
- [18] H.-P. Tan, R. Diamant, W. K. Seah, and M. Waldmeyer, “A survey of techniques and challenges in underwater localization,” *Ocean Engineering*, vol. 38, no. 14, pp. 1663–1676, 2011.
- [19] H. International, “Long baseline systems,” Hydro International, Tech. Rep., 2008. [Online]. Available: [http://www.hydro-international.com/files/productsurvey\\_v\\_pdfdocument\\_20.pdf](http://www.hydro-international.com/files/productsurvey_v_pdfdocument_20.pdf)
- [20] —, “Ultra short baseline systems,” Hydro International, Tech. Rep., 2008. [Online]. Available: [http://www.hydro-international.com/files/productsurvey\\_v\\_pdfdocument\\_21.pdf](http://www.hydro-international.com/files/productsurvey_v_pdfdocument_21.pdf)
- [21] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*. Cambridge, M.A.: MIT Press, 2005.
- [22] D. K. Montemerlo, S. Thrun and B. Wegbreit, “Fastslam 2.0: An improved particle filtering algorithm for simultaneous localization and mapping that provably converges,” *Proc. of the International Joint Conference on Artificial Intelligence (IJCAI)*, 2003.
- [23] J. Blanco, J. Gonzalez, and J. Fernández-Madriral, “A pure probabilistic approach to range-only SLAM,” in *Proceedings of the IEEE International Conference on Robotics and Automation*, 2008, pp. 1436–1441.
- [24] P. Newman and J. Leonard, “Pure range-only sub-sea slam,” in *ICRA03*, vol. 2, Washington DC, 2003, pp. 1921–1926.
- [25] J. S. F. Le Bars, A. Bertholom and L. Jaulin, “Interval slam for underwater robots; a new experiment,” in *NOLCOS 2010*, Italy, 2010.
- [26] L. Jaulin, “Range-only SLAM with occupancy maps; A set-membership approach,” *IEEE Transaction on Robotics*, vol. 27, no. 5, pp. 1004–1010, 2011.
- [27] —, *MSLAM, a solver for range-only SLAM with undistinguishable landmarks, available at [www.ensta-bretagne.fr/jaulin/mslam.html](http://www.ensta-bretagne.fr/jaulin/mslam.html)*. LabSTICC, ENSTA Bretagne, 2004.

- [28] C. Joly and P. Rives, *Bearing-only SLAM : comparison between probabilistic and deterministic methods*. Research Report RR-6602, INRIA. URL <http://hal.inria.fr/inria-00308722/en/>, 2008.
- [29] F. L. Bars, “Analyse par intervalles pour la localisation et la cartographie simultanées ; application à la robotique sous-marine,” PhD dissertation, Université de Bretagne Occidentale, Brest, France, 2011.
- [30] G. Chabert and L. Jaulin, “Propagation de contraintes sur les intervalles pour la planification d’expérience en slam,” *Soumis à Revue d’Intelligence Artificielle*, 2011.
- [31] L. Jaulin and B. Desrochers, “Separators: a new interval tool to bracket solution sets; application to path planning,” *Engineering Applications of Artificial Intelligence*, vol. 33, pp. 141–147, 2014.
- [32] L. Jaulin, “Path planning using intervals and graphs,” *Reliable Computing*, vol. 7, no. 1, pp. 1–15, 2001.
- [33] —, “Path Planning Using Intervals and Graphs,” *Reliable Computing*, vol. 7, no. 1, pp. 1–15, 2001.
- [34] M. Lhommeau, L. Hardouin, B. Cottenceau, and L. Jaulin, “Interval analysis and dioid: Application to robust controller design for timed event graphs,” *Automatica*, vol. 40, no. 11, pp. 1923–1930, 2004.
- [35] J. Aubin and H. Frankowska, *Set-Valued Analysis*. Birkhäuser, Boston, 1990.
- [36] F. Abdallah, A. Gning, and P. Bonnifait, “Box particle filtering for nonlinear state estimation using interval analysis,” *Automatica*, vol. 44, no. 3, pp. 807–815, 2008.
- [37] V. Drevelle, “Etude de methodes ensemblistes robustes pour une localisation multisensorielle integre. application a la navigation des vehicules en milieu urbain,” Ph.D. dissertation, etc, 2011.
- [38] C. Rohrig and M. Muller, “Indoor location tracking in non-line-of-sight environments using a ieee 802.15. 4a wireless network,” in *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*. IEEE, 2009, pp. 552–557.
- [39] A. Turetta, G. Casalino, E. Simetti, A. Sperinde, and S. Torelli, “Analysis of the accuracy of a lbl-based underwater localization procedure,” in *Oceans - St. John’s, 2014*, Sept 2014, pp. 1–7.
- [40] R. E. Moore, *Interval Analysis*. Englewood Cliffs, NJ: Prentice-Hall, 1966.
- [41] L. Jaulin and E. Walter, “Set inversion via interval analysis for nonlinear bounded-error estimation,” *Automatica*, vol. 29, no. 4, pp. 1053–1064, 1993.
- [42] D. Maksarov and J. P. Norton, “State bounding with ellipsoidal set description of the uncertainty,” *International Journal of Control*, vol. 65, no. 5, pp. 847–866, 1996.

- 
- [43] C. Combastel, “A state bounding observer for uncertain non-linear continuous-time systems based on zonotopes,” in *CDC-ECC '05*, 2005.
- [44] E. Goubault, S. Putot, and F. Védryne, “Modular static analysis with zonotopes,” in *Static Analysis*. Springer, 2012, pp. 24–40.
- [45] K. Ghorbal, E. Goubault, and S. Putot, “A logical product approach to zonotope intersection,” in *Computer Aided Verification*. Springer, 2010, pp. 212–226.
- [46] L. Jaulin, M. Kieffer, O. Didrit, and E. Walter, *Applied Interval Analysis*. Springer, 2001.
- [47] E. Goubault, “Static analyses of the precision of floating-point operations,” in *Static Analysis*. Springer, 2001, pp. 234–259.
- [48] E. Goubault, M. Martel, and S. Putot, “Asserting the precision of floating-point computations: a simple abstract interpreter,” in *Programming Languages and Systems*. Springer, 2002, pp. 209–212.
- [49] I. A.N.S.I., *A standard for binary floating-point arithmetic*. New York: ANSI/IEEE Std. 754-1985, 1985.
- [50] “Ieee standard for floating-point arithmetic,” *IEEE Std 754-2008*, pp. 1–70, Aug 2008.
- [51] M. Berz and K. Makino, “Verified Integration of ODEs and Flows using Differential Algebraic Methods on High-Order Taylor Models,” *Reliable Computing*, vol. 4, no. 3, pp. 361–369, 1998.
- [52] A. Kurzhanski and I. Valyi, *Ellipsoidal Calculus for Estimation and Control*. Boston, MA: Birkhäuser, 1997.
- [53] M. Milanese, J. Norton, H. Piet-Lahanier, and E. Walter, Eds., *Bounding Approaches to System Identification*. New York, NY: Plenum Press, 1996.
- [54] V. Drevelle and P. Bonnifait, “High integrity gnss location zone characterization using interval analysis,” in *ION GNSS*, 2009.
- [55] A. K. Mackworth, “Consistency in networks of relations,” *Artificial Intelligence*, vol. 8, no. 1, pp. 99–118, Feb. 1977.
- [56] J. G. Cleary, “Logical arithmetic,” *Future Computing Systems*, vol. 2, no. 2, pp. 125–149, 1987.
- [57] E. Davis, “Constraint propagation with interval labels,” *Artificial Intelligence*, vol. 32, no. 3, pp. 281–331, 1987.
- [58] E. Hyvonen, “Constraint reasoning based on interval arithmetics: the tolerance propagation approach,” *Artificial Intelligence*, vol. 58, no. 1–3, pp. 71–112, Dec. 1992.
- [59] O. Lhomme, “Consistency techniques for numeric CSPs,” in *Proceedings of the International Joint Conference on Artificial Intelligence*, Chambéry, France, 1993, pp. 232–238.

- [60] D. Sam-Haroud, “Constraint consistency techniques for continuous domains,” PhD dissertation 1423, Swiss Federal Institute of Technology in Lausanne, Switzerland, 1995.
- [61] D. Sam-Haroud and B. Faltings, “Consistency techniques for continuous constraints,” *Constraints*, vol. 1, no. 1-2, pp. 85–118, 1996.
- [62] P. van Hentenryck, L. Michel, and F. Benhamou, “Newton: Constraint programming over nonlinear constraints,” *Science of Computer Programming*, vol. 30, no. 1–2, pp. 83–118, Jan. 1998.
- [63] I. Araya, B. Neveu, and G. Trombettoni, “Exploiting Common Subexpressions in Numerical CSPs,” in *Proc. CP, Constraint Programming*, LNCS 5202, 2008, pp. 342–357.
- [64] M. Ceberio and L. Granvilliers, “Solving nonlinear systems by constraint inversion and interval arithmetic,” in *Artificial Intelligence and Symbolic Computation*, vol. 1930, LNCS 5202, 2001, pp. 127–141.
- [65] G. Chabert and L. Jaulin, “A Priori Error Analysis with Intervals,” *SIAM Journal on Scientific Computing*, vol. 31, no. 3, pp. 2214–2230, 2009.
- [66] F. Benhamou, F. Goualard, L. Granvilliers, and J.-F. Puget, “Revising Hull and Box Consistency,” in *ICLP*, 1999, pp. 230–244.
- [67] D. Waltz, “Generating semantic descriptions from drawings of scenes with shadows,” in *The Psychology of Computer Vision*, P. H. Winston, Ed. New York, NY: McGraw-Hill, 1975, pp. 19–91.
- [68] A. Gning and P. Bonnifait, “Constraints propagation techniques on intervals for a guaranteed localization using redundant data,” *Automatica*, vol. 42, no. 7, pp. 1167–1175, 2006.
- [69] R. Guyonneau, S. Lagrange, L. Hardouin, and P. Lucidarme, “Guaranteed interval analysis localization for mobile robots,” *Advanced Robotics*, vol. 28, no. 16, pp. 1067–1077, 2014.
- [70] M. Morgado, P. Batista, P. Oliveira, and C. Silvestre, “Position usbl/dvl sensor-based navigation filter in the presence of unknown ocean currents,” *Automatica*, vol. 47, no. 12, pp. 2604 – 2614, 2011. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S000510981100464X>
- [71] N. G. M. Sanjeev Arulampalam, Simon Maskell and T. Clapp, “A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking,” *IEEE Transactions on Signal Processing*, vol. 50, no. 2, pp. 174–188, 2002.
- [72] L. Jaulin, M. Kieffer, I. Braems, and E. Walter, “Guaranteed nonlinear estimation using constraint propagation on sets,” *International Journal of Control*, vol. 74, no. 18, pp. 1772–1782, 2001.
- [73] K. Kaouri, “Left-right ambiguity resolution of a towed array sonar,” Ph.D. dissertation, University of Oxford, 2000.

- 
- [74] J. J. Sallas, “Marine vibrator tuneable array,” Apr. 17 1990, uS Patent 4,918,668.
- [75] K. Jo, J. Lee, and J. Kim, “Cooperative multi-robot localization using differential position data,” in *Advanced intelligent mechatronics, 2007 IEEE/ASME international conference on*. IEEE, 2007, pp. 1–6.
- [76] K.-T. Song, C.-Y. Tsai, and C.-H. C. Huang, “Multi-robot cooperative sensing and localization,” in *Automation and Logistics, 2008. ICAL 2008. IEEE International Conference on*. IEEE, 2008, pp. 431–436.
- [77] A. Bahr, J. J. Leonard, and M. F. Fallon, “Cooperative localization for autonomous underwater vehicles,” *The International Journal of Robotics Research*, vol. 28, no. 6, pp. 714–728, 2009.
- [78] A. Bethencourt, “Interval analysis for swarm localization, application to underwater robotics.” Ph.D. dissertation, ENSTA Bretagne, 2014.
- [79] T. B. Curtin, J. G. Bellingham, J. Catipovic, and D. Webb, “Autonomous oceanographic sampling networks,” *Oceanography*, vol. 6, no. 3, pp. 86–94, 1993.
- [80] P. Herréro, “Quantified real constraint solving using modal intervals with applications to control,” PhD dissertation, University of Girona, Spain, 2006.
- [81] B. Desrochers and L. Jaulin., “A minimal contractor for the polar equation; application to robot localization,” *Submitted to Engineering Applications of Artificial Intelligence*.